# Scalable and Reliable Evaluation of AI Knowledge Retrieval Systems: RIKER and the Coherent Simulated Universe

JV Roig
Kamiwaza AI
jv@kamiwaza.ai

December 2025

**Abstract**

Evaluating knowledge systems (LLMs, RAG, knowledge graphs, etc) faces fundamental challenges: static benchmarks are vulnerable to contamination, LLM-based judges exhibit systematic biases, and ground truth extraction requires expensive human annotation. We present RIKER (Retrieval Intelligence and Knowledge Extraction Rating), both a benchmark and a replicable methodology based on paradigm inversion - generating documents *from* known ground truth rather than extracting ground truth *from* documents. This approach enables deterministic scoring and scalable evaluation without human annotation or reference models, and contamination resistance through regenerable corpora. Our evaluation of 33 models using over 21 billion tokens reveals that context length claims frequently exceed usable capacity, with significant degradation beyond 32K tokens; cross-document aggregation proves substantially harder than single-document extraction; and grounding ability and hallucination resistance are distinct capabilities - models excelling at finding facts that exist may still fabricate facts that do not. Beyond the specific benchmark, we contribute a domain-agnostic methodology for constructing scalable and contamination-resistant evaluations wherever synthetic documents can be generated from structured ground truth.

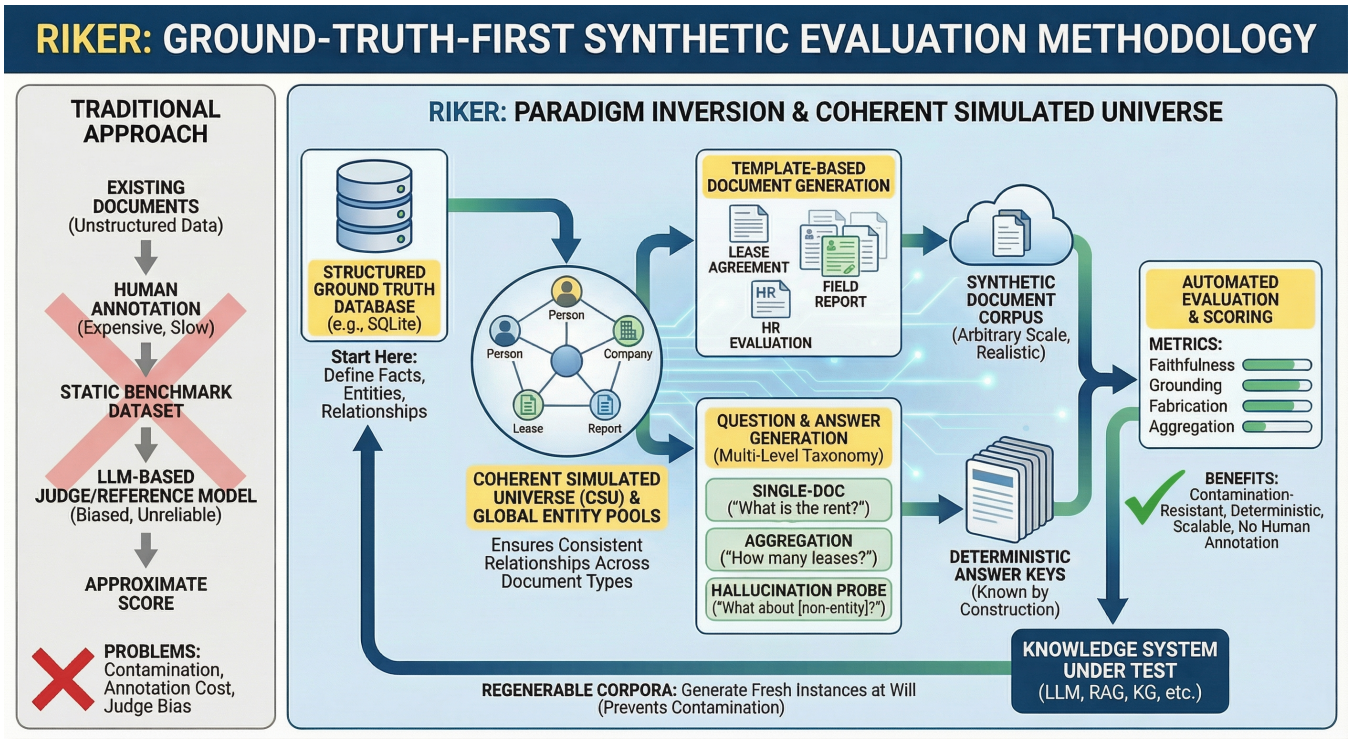


Figure 1: RIKER methodology overview. Traditional approaches (left) extract ground truth from existing documents via expensive human annotation, producing static benchmarks vulnerable to contamination and requiring biased LLM judges. RIKER (right) inverts this: structured ground truth is defined first, then documents and questions are generated from it, enabling deterministic scoring and regenerable corpora.

## 1 Introduction

One of the most common and also most critical uses of agentic AI in the enterprise is processing vast amounts of internal enterprise knowledge. This comes in many forms:

- Relevant enterprise documents are loaded into the context of the large language model (LLM) at the beginning of a chat session
- Snippets of relevant enterprise documents are loaded as needed, through a knowledge retrieval mechanism - such as through various Retrieval Augmented Generation (RAG) methodologies using vector databases, traditional enterprise search, ontologies, or any mix of these techniques
- An LLM agent, given appropriate tools and access, can also retrieve internal and external information as needed, in an Agentic RAG manner.

The list above is not meant to be exhaustive - merely illustrative of the many ways that LLMs are deployed to be able to use enterprise knowledge in order to be useful to the enterprise.

The huge enterpise gap here is: how do we QA (quality assurance) all of these in a scalable and reliable manner? This simple question extends to many related questions that enterprise teams can find themselves asking: *Which model hallucinates less? Which model is better retrieving facts if we dump documents into its context? Should I use a vector database or a graph database? How do I test and quantify how much better or worse a chunking/embedding/retrieval configuration is better over another? How much does an ontology improve our knowledge retrieval?*

As before, the list above is not meant to be exhaustive, just illustrative of the gap. This gap exists because, currently, answering questions like these is extremely difficult.

If this gap can be solved, it will not only be useful to enterprise, but also to the research community. A tool that can quantify how accurate or inaccurate certain retrieval methods are (from simple LLM context stuffing, to ornate ontologies and knowledge graphs) would be an extremely useful tool not only as a primary QA tool in the enterprise, but could help the research and development of improved retrieval systems.

In this work, we propose one such tool - Retrieval Intelligence and Knowledge Extraction Rating (RIKER). RIKER is an offshoot of our earlier work called PICARD (Probing Intelligent Capabilities via Artificial Randomized Data) [49] - a framework for contamination-resistant benchmarking of agentic AI capabilities of LLMs. RIKER builds upon PICARD to extend evaluation to knowledge extraction scenarios, as well as evaluation of knowledge retrieval systems in general beyond just LLMs.

We present RIKER as two contributions: first, a concrete benchmark for enterprise document understanding with empirical results across 31 models; second, and more broadly, a replicable *methodology* for constructing scalable and reliable evaluations of knowledge retrieval systems. The methodology - generating documents FROM known ground truth rather than extracting ground truth FROM documents - is domain-agnostic and can be applied wherever synthetic documents can be generated from structured facts. Figure 1 illustrates the approach.

The key contributions of this work are:

- **Ground truth by construction** through paradigm inversion - rather than extracting ground truth from documents, RIKER generates documents FROM known ground truth, eliminating human annotation
- **Scalable and reliable benchmarking** through procedural generation - regenerable corpora and tests create fresh, never-before-seen documents and questions. Documents and tests are independently scalable.
- The **Coherent Simulated Universe** approach - synthetic documents that maintain realistic entity relationships across document types
- A **multi-level question taxonomy** spanning single-document extraction, cross-document aggregation, and hallucination detection
- **Empirical evaluation of 31 models** across 32K, 128K, and 200K token contexts, totaling nearly 20 billion tokens of evaluation

Our evaluation reveals several key findings: top-tier models achieve over 80% accuracy at 32K context but degrade significantly at longer contexts; aggregation queries prove substantially harder than single-document extraction; and several models exhibit catastrophic failure modes including coherence loss and hallucination spikes exceeding 70%. Cross-corpus validation with independent document sets confirms that RIKER measures model capability rather than corpus-specific artifacts.

## 2 Related Work

Evaluating knowledge retrieval and extraction systems faces fundamental methodological challenges. Current approaches rely on static benchmarks vulnerable to contamination, LLM-based judges with documented biases, and approximated ground truth derived from expensive human annotation. This section reviews the current state of knowledge retrieval evaluations of LLMs and various knowledge retrieval systems like RAG and knowledge graphs.

### 2.1 Long-Context LLM Evaluation

The Needle-in-a-Haystack (NIAH) paradigm [31] places a random fact in a long context and tests retrieval. While influential, the original NIAH fundamentally tests *retrieval*, not *comprehension* - finding an arbitrary fact buried in unrelated text is pattern-matching, not document understanding. Ironically, the needle's incongruence with its context makes it *easier* to find; it stands out precisely because it doesn't belong.

Subsequent benchmarks address various limitations with meaningful improvements. **Sequential-NIAH** [72] tests whether models can extract facts in correct chronological order across 8K - 128K contexts, demonstrating that sequential understanding remains challenging even for frontier models. **RULER** [28] expands beyond retrieval to include multi-hop tracing, aggregation, and question answering tasks; despite models claiming 32K+ context support, many fail to maintain satisfactory performance at that length. **NeedleChain** [46] takes a stricter approach: every piece of context is essential for answering queries, so missing even one element results in failure - revealing that models achieving near-perfect NIAH scores struggle under these stricter conditions.

For more realistic evaluation, **LongBench v2** [5] provides human-annotated tasks across six categories (single/multi-document QA, code understanding, dialogue history, structured data) with contexts from 8K to 2M words, demonstrating that long-context comprehension remains challenging even for frontier models. **InfiniteBench** [74] pushes context length to 100K+ tokens using real content from novels, code repositories, and mathematical problems.

Additional benchmarks serve specialized purposes: U-NIAH [23] provides unified comparison between RAG and LLM approaches on NIAH-style tasks, while MMNeedle [63] extends the paradigm to multimodal (image) contexts.

The "Lost in the Middle" phenomenon [39] demonstrates that LLMs struggle with information placed in the middle of long contexts, with some mitigation approaches proposed [75].

### 2.2 Retrieval and Embedding Benchmarks

BEIR [60] and MTEB [47] are the standard benchmarks for retrieval evaluation. Newer domain-specific benchmarks include FreshStack [13] for technical documents and MIRAGE [66] for medical retrieval. However, all face contamination concerns. BEIR is no longer a true zero-shot benchmark, as researchers now routinely include BEIR datasets in their training pipelines [30]. MTEB's leaderboard now has 400+ models with marginal performance differences, suggesting either saturation or overfitting to the benchmark distribution.

FreshStack is less prone to contamination, but relies on LLM-as-a-judge for evaluation.

### 2.3 Multi-Hop Question Answering

Multi-hop QA benchmarks like HotpotQA [69] and 2WikiMultiHopQA [26] aim to test reasoning across multiple evidence pieces. More recent efforts include MoreHopQA [56] for deeper reasoning chains and multi-hop RAG approaches [59]. However, shortcut exploitation undermines validity: nearly 61% of HotpotQA's multi-hop questions can actually be answered using single-hop reasoning, with a simple BERT-based single-hop model achieving performance comparable to state-of-the-art multi-hop systems [44].

MuSiQue [61] was designed specifically to prevent shortcuts through "unanswerable" questions, but remains static and thus vulnerable to contamination.

### 2.4 RAG Evaluation

RAG evaluation has received significant survey attention [71, 21]. Popular frameworks like RAGAS [20] and ARES [53] rely on LLM-as-judge approaches, with domain-specific validation in areas like telecommunications [51]. However, empirical validation reveals significant limitations: correlation between RAGAS metrics and human evaluation yields a harmonic mean of only 0.55, far below what would be required for reliable automated evaluation [8].

The CALM framework [70] documents 12 distinct biases in LLM judges, including position bias [57], verbosity bias, self-enhancement bias, and authority bias. Additional critiques appear in [22], [12], and comparative studies of human vs. LLM judges [10]. RAGChecker [52] advances the field with fine-grained diagnostic metrics but still uses LLM-based entailment checking.

For agentic RAG, evaluation remains nascent. Recent surveys [36] and benchmarks like RAGCap-Bench [37], HopRAG [38], and TelAgentBench [33] address this gap, but RAGCap-Bench finds that current systems still struggle with challenging multi-hop questions and their intermediate reasoning capabilities remain underexplored. Community resources track ongoing developments [14].

### 2.5 GraphRAG and Knowledge Graph Evaluation

GraphRAG promises improved retrieval through knowledge graph augmentation [15], but evaluation challenges persist. Benchmarks like GraphRAG-Bench [24] and evaluation frameworks [73, 55] attempt to standardize assessment, while studies examine when graphs help RAG [65]. When evaluated with ground truth rather than LLM-as-judge, community-based GraphRAG, particularly with global search, generally underperforms compared to standard RAG [25]. This discrepancy arises because the original GraphRAG evaluation [18] used LLM-as-judge without ground truth - precisely the methodology shown to inflate results.

Knowledge graph construction faces fundamental evaluation challenges: recent surveys identify unresolved issues in establishing reliable benchmarks for KG quality assessment, particularly regarding intrinsic and extrinsic evaluation metrics [11]. The relationship between KGs and hallucination has been studied extensively [32]. GOSyBench [9] provides domain-specific KG extraction benchmarks, demonstrating that even frontier models struggle with accurate knowledge graph recovery. FinReflectKG [4] combines rule-based checks, statistical validation, and LLM-as-judge assessments to measure extraction quality - a multi-layered approach reflecting the difficulty of establishing absolute quality metrics for KG construction.

### 2.6 Hallucination and Factuality Benchmarks

Hallucination detection and factuality evaluation have received extensive attention [29]. TruthfulQA and SimpleQA [64] evaluate factual accuracy, but face distinct challenges. TruthfulQA shows evidence of contamination [16], while SimpleQA focuses on short-form responses where even frontier models struggle to achieve majority accuracy. HaluEval [34] and HalluLens [7] provide additional hallucination benchmarks, with FastFact [62] offering efficient fact verification.

For long-form factuality, FActScore [43] decomposes responses into atomic facts for verification. Studies on generalization vs. memorization [17] inform understanding of when models hallucinate. No benchmark adequately addresses long-form knowledge extraction from document corpora.

### 2.7 Benchmark Contamination

Data contamination undermines benchmark validity across domains, with comprehensive surveys documenting the extent of the problem [67, 41]. Meta-analyses question benchmark trustworthiness broadly [19]. Simple variations such as paraphrasing or translation easily bypass standard decontamination measures, allowing a 13B model to overfit leaked benchmarks and achieve GPT-4-level performance [68].

MMLU shows 52 - 57% exact-match guessing rates on contaminated subsets [16]. Current solutions - LatestEval [35] (temporal freshness), MMLU-CF [76] (counterfactual rephrasing) - are *reactive*: they detect and mitigate contamination rather than prevent it structurally.

### 2.8 Synthetic Data Generation and Simulation Validity

RIKER generates synthetic documents from structured ground truth, situating it within the synthetic data literature [1]. Surveys on LLM-driven synthetic data generation [40] and user simulation [6] provide theoretical grounding.

#### 2.8.1 The Bias Factor Problem

When LLMs generate benchmark data and perform the task, systematic biases emerge. Smaller LLMs exhibit biases towards their own generated data, whereas larger models do not - a phenomenon termed the "bias factor" [42]. This undermines validity for LLM-generated benchmarks. RIKER avoids this through **template-based generation** - no LLM is involved in document creation.

#### 2.8.2 The Reality Gap

The "reality gap" - the gap between synthetic training data and real-world deployment data - is not merely technical but epistemological: synthetic data requires real-world data about a domain in order to model it, the very data it purports to dispense with [58]. A validity-centered framework for AI evaluation [54] provides psychometric grounding for what claims benchmarks can legitimately support. RIKER makes bounded validity claims: it measures extraction accuracy against known ground truth. It does *not* claim that success on RIKER guarantees real-world performance. Rather, RIKER provides a *necessary* (not sufficient) condition - if a system cannot extract known facts from synthetic documents, it will fail on real documents.

### 2.8.3 Parameterized Evaluation

Two benchmarks validate template-based evaluation:

**GSM-Symbolic** [45] uses symbolic templates for math problems, finding that adding irrelevant clauses causes up to 65% performance drops. The authors conclude: "Current LLMs cannot perform genuine logical reasoning; they replicate reasoning steps from their training data."

**RV-Bench** [27] generates Random Variable Questions (RVQs) with randomized variable combinations, testing 30+ LLMs and finding "proficiency imbalance" between familiar and novel combinations.

Both GSM-Symbolic and RV-Bench demonstrate that parameterized generation exposes capability gaps that static benchmarks miss. RIKER extends this paradigm from mathematical reasoning to knowledge extraction.

## 2.9 The PICARD Framework

The PICARD framework [49] addresses evaluation gaps for agentic AI through:

- **Ground-truth-first generation**: "When the evaluation framework controls data generation, it inherently possesses complete ground truth"
- **Deterministic scoring**: Answer keys generated simultaneously with test data
- **Anti-memorization by design**: Combinatorial explosion makes memorization impossible
- **Multi-layered randomization**: Entity substitution, data generation, and environmental variation

PICARD demonstrates these principles for file manipulation, database operations, and multi-step workflows. However, PICARD does not address knowledge extraction from document corpora - the domain RIKER targets.

## 2.10 Summary: Research Gaps

Table 1 summarizes the gaps addressed by RIKER.

Table 1: Research Gaps and RIKER's Position

| Gap | Current Limitation | RIKER's Approach |
|---|---|---|
| LLM-as-Judge unreliability | 12 documented biases; 0.55 human correlation | Deterministic scoring |
| Benchmark contamination | Static benchmarks vulnerable to memorization | Regenerable corpora + combinatorial anti-memorization |
| Long-context retrieval $\neq$ comprehension | Needle retrieval $\neq$ document comprehension | Realistic document understanding |
| Static multi-hop QA | Single-hop shortcuts | Aggregation requires cross-document extraction |
| RAG ground truth approximated | Human annotation bottleneck | Ground-truth-first generation |
| No KG extraction ground truth | Reliable KG benchmark establishment remains unresolved | Generate FROM structured ground truth with known entity relationships |
| KG completeness unmeasurable | Graph edit distance insufficient | SQLite manifest defines expected output |
| Retrieval benchmark contamination | BEIR no longer zero-shot | Regenerable corpus |
| KG evaluation schema-dependent | Tied to specific ontology | Query answers, not structure |

| Gap | Current Limitation | RIKER's Approach |
|---|---|---|
| Data bottleneck | Annotation expensive, doesn't scale | Arbitrary-scale generation |
| GraphRAG evaluation inflated | LLM-judge without ground truth | Ground truth scoring |

The pattern reveals three fundamental problems: (1) static datasets get contaminated, (2) LLM judges are unreliable, and (3) ground truth is approximated rather than known. Industry reports confirm that data bottlenecks have increased significantly [3], while best practices for ground truth generation remain labor-intensive [2]. RIKER's paradigm inversion - generating documents FROM ground truth rather than extracting ground truth FROM documents - addresses all three.

## 3 The RIKER Approach

Unlike benchmarks that rely on static datasets or LLM-as-judge evaluation, RIKER generates synthetic corpora from embedded ground truth, enabling deterministic scoring at scale. The methodology is application-agnostic - it can evaluate context-stuffing, retrieval-augmented generation, or knowledge graph systems by instrumenting the system under test to answer RIKER-generated questions.

### 3.1 Synthetic Corpus Generation

RIKER employs a ground-truth-first architecture: the complete knowledge base—all entities, relationships, and facts—is populated in a relational database *before* any document is generated. Documents are then rendered as human-readable views of this underlying ground truth. This inversion of the typical "extract facts from documents"approach provides three key advantages: (1) every question has a verifiable answer by construction, enabling deterministic scoring, (2) corpora can be regenerated with different random seeds while maintaining structural equivalence, enabling robustness validation, and (3) the approach is easily scalable to practically-unlimited scale, requiring no human-intensive document annotation.

#### 3.1.1 Synthetic Data Generation

Built upon the PICARD framework [49], RIKER inherits and expands various synthetic data generation functions - names, amounts, dates, and other entities - which are used to generate a diverse set of ground truth elements, from which documents will later be created.

#### 3.1.2 Ground Truth Database

All generated ground truth is recorded in a SQLite database with full relational structure. For example, for a lease document corpus, this includes:

- Document metadata (parties, dates, amounts, clauses present, clauses absent)
- Entities (lessors, lessees, agents, addresses, etc.)
- Entity relationships (which lessors have which lessees)

This database serves as the authoritative answer key for all generated questions. Questions that require computed aggregations (counts, sums, temporal relationships) can be derived from this ground truth through SQL, which enables complex test question generation (e.g. "What is the total monthly rent of all leases $LESSOR has in $YEAR and $MONTH")

#### 3.1.3 Template-Based Document Generation

After the ground truth database is created, documents are generated using modular templates with controlled variation. Each template defines the document structure while allowing randomized selection of:

- Language style (formal, semi-formal, casual)
- Structural organization (section ordering, optional clauses)
- Boilerplate text variations

This produces documents that are structurally consistent yet superficially diverse, preventing models from exploiting surface-level patterns while maintaining ground truth integrity.

### 3.2 Coherent Simulated Universe

One of the significant shortcomings of synthetic generated data comes from naive generation - that is, when 100 synthetic documents are generated, but they are all *independent generations*. This gives the synthetic data set a characteristic that is very *un-enterprise-like* - the documents are not related to each other, or worse, they have chaotic and unrealistic connections.

For example, in a naive generation, we could generate 100 HR documents using a pool of human names. Being independently randomly generated, our documents (for example, HR evaluations or employee information) would naively randomize facts like employee name, manager, department, etc. The result is a dataset that models no realistic enterprise corpus, for example:

- Employees in similar randomized department (by chance) will have a different dept manager or supervisor named
- Employees in different departments may accidentally have a similar randomized person name
- An employee, named as a manager or supervisor in a particular department from a previous document, can have a different department or position

The above is not an exhaustive list. This incoherence resulting from naive random generation is a problem. It does not model a realistic enterprise scenario (therefore metrics against that dataset may have very weak correlation to real-world performance), and will prevent the creation of challenging comprehension and aggregation questions, such as *'which manager gave the most evaluations this quarter?'*, because necessary relationships will either not be diverse or coherent enough, or may not even exist at all.

RIKER solves this problem through its **Coherent Simulated Universe** approach. The very first step in document generation is ground truth creation (see 3.1.2), and this includes necessary relationships among the different entities. To make this coherence spread across different document types and across the entire universe of generated documents, RIKER has the concept of **Global Entity Pools**, which are pre-generated and filled as the first step of synthetic data generation. Relationships are then created by drawing from the global entity pools, which are saved in the ground truth database. These global entity pools are used across all document types to create a coherent dataset. For example, a global entity pool called 'sales_agent' contains human names that are used for three types of documents in the current RIKER implementation:

- An *optional sales agent* that is named and credited for closing a **Lease Contract**
- A *sales agent* that is named in a **Sales Agent Field Report**, as the agent who created and submitted the field report detailing sales activities and potential contract status
- An *employee* that is named in an **HR Employee Evaluation** document, as the sales employee being evaluated

In RIKER's Coherent Simulated Universe, all the three document types above (in bold) draw from the same Global Entity Pool - meaning the agents you will see who closed lease contracts will be the same agents named in relevant field reports, and named in relevant HR evaluation documents.

It will also never happen that a Field Report talking about a particular Lease Contract will have a different human name randomized for it as the sales agent - this incoherence is avoided through logic specifically baked-in to the document generation feature, as part of the Coherent Simulated Universe strategy.

This results in having arbitrary scale document generation where ground truth is immediately available with no human annotation effort (because ground truth is where the process actually begins), and the knowledge generated - the entire set of facts and documents - are coherent according to the generation design.

### 3.3 Multi-Level Question Taxonomy

RIKER generates questions across twelve difficulty levels organized into three categories, each testing distinct capabilities.

#### 3.3.1 Single-Document Questions (L01 - L04)

These questions require locating and extracting information from a single document:

- **L01 - Direct Extraction:** Surface-level facts stated explicitly ("What is the monthly rent?")
- **L02 - Indirect Extraction:** Facts requiring minimal inference ("What is the lease duration?" when start/end dates are given)
- **L03 - Conditional Extraction:** Facts from optional document sections ("What is the pet deposit?" — may be N/A)
- **L04 - Complex Extraction:** Facts requiring multiple conditions or cross-referencing within a document

#### 3.3.2 Aggregation Questions (L05 - L10)

These questions require synthesizing information across multiple documents:

- **L05 - Counting:** "How many leases does Lessor X have?"
- **L06 - Summation/Averaging:** "What is the total monthly rent across all leases?"
- **L07 - Comparison:** "Which lessor has more leases, X or Y?"
- **L08 - Enumeration:** "List all lessees for Lessor X"
- **L09 - Multi-hop:** "What is Lessor X's most recent lease end date?"
- **L10 - Temporal:** "How many leases were active in Q3 2024?"

Aggregation questions are particularly challenging because they require the model to: (1) identify all relevant documents, (2) extract the relevant facts from each, and (3) perform the required computation correctly.

#### 3.3.3 Hallucination Probe Questions (L11 - L12)

These questions are designed to detect fabrication:

- **L11 - Non-existent Entities:** Questions about entities that do not appear anywhere in the corpus. The entity names are drawn from unused portions of the entity pool, ensuring they are plausible but definitively absent. The only correct response is "Unknown" or equivalent.
- **L12 - Absent Information:** Questions about optional fields that are absent from specific documents. For example, asking about the pet deposit for a lease that has no pet clause. The only correct response is "N/A" or equivalent.

L11 questions are particularly valuable because any specific answer constitutes unambiguous fabrication—the model cannot have retrieved the information from the corpus because it does not exist.

### 3.4 Deterministic Scoring

RIKER employs answer-key-based scoring, eliminating the variability inherent in LLM-as-judge approaches.

#### 3.4.1 Scoring Mechanisms

Each question specifies its scoring type:

- **Exact match:** For categorical responses (names, yes/no)
- **Numeric extraction:** Parses numerical answers with tolerance for formatting variations
- **Set comparison:** For enumeration questions, compares answer sets regardless of ordering
- **Semantic equivalence:** For structured responses with known equivalent forms

All scoring logic operates against the ground truth database, ensuring reproducibility. The same model outputs will always receive the same scores.

In this particular

#### 3.4.2 Response Format Enforcement

Questions include explicit format instructions (e.g., "Reply with only the number" or "Indicate your final answer with: Final answer: [your answer]"). This structured output requirement reduces ambiguity in answer extraction and improves scoring reliability.

### 3.5 Fidelity Metrics Taxonomy

We define a three-level taxonomy for hallucination-related metrics

#### 3.5.1 Faithfulness (L01 - L04 + L11 - L12)

The broadest metric, measuring accuracy on all questions where the model had sufficient information to answer correctly. This encompasses both grounding failures (wrong answers from documents that exist) and fabrication (invented information). Faithfulness aligns with the colloquial enterprise definition of "hallucination" as any confidently wrong answer when correct information was available.

#### 3.5.2 Grounding (L01 - L04)

Accuracy on single-document questions only. Grounding failures indicate the model could not locate or correctly extract information from documents that definitively contain the answer. This isolates retrieval and comprehension errors from fabrication.

#### 3.5.3 Fabrication (L11 - L12)

Error rate on hallucination probe questions. Because L11 questions ask about non-existent entities, any specific answer is definitively fabricated—there is no ambiguity about the failure mode. This provides the cleanest signal for measuring a model's tendency to invent information.

#### 3.5.4 Aggregation (L05 - L10)

Reported separately as a capability metric rather than a hallucination metric. Aggregation errors conflate multiple failure modes (incomplete document retrieval, computation errors, working memory limitations) that are distinct from hallucination in the traditional sense.

### 3.6 Robustness Validation

A methodology is only useful if it produces stable, reproducible results. We validated RIKER's robustness through cross-corpus experiments.

#### 3.6.1 Cross-Corpus Stability

Four corpora were generated from identical configuration parameters but different random seeds, producing documents with different entity names, dates, and surface content while maintaining structural equivalence. Four models spanning different performance tiers were evaluated on all four corpora.

Results demonstrated strong stability: top-performing models showed less than 2% accuracy variance across corpora (CV $<$ 1%), with consistent ranking preservation. This validates that RIKER results reflect model capability rather than corpus-specific artifacts.

#### 3.6.2 Implications for Reproducibility

The cross-corpus stability finding has practical implications: researchers can generate their own RIKER corpora and expect comparable results to other studies using the same configuration parameters. This addresses a key limitation of static benchmarks: their fixed nature means they inevitably leak into training corpora, while RIKER's regenerability ensures fresh, uncontaminated test data.

## 4 Experimental Design

Table 2 summarizes the scale of our evaluation for this RIKER study. 33 models and over 21B tokens were processed.

Table 3 details the document breakdown across corpora; question counts scale proportionally, with approximately 50% single-document extraction, 40% cross-document aggregation, and 10% hallucination probes. All three document types appear in every corpus, demonstrating the Coherent Simulated Universe in practice: the same entities (people, properties, companies) appear across leases, field reports, and HR evaluations. The document distribution reflects realistic business ratios - leases are fewer (one per tenancy), field reports dominate (generated per agent-prospect interaction), and HR reports fall in between (periodic per employee).

Model coverage decreases with context size ($33 \rightarrow 24 \rightarrow 11$) as fewer models support longer contexts - this stratification is itself a finding. Eight runs per model enable statistical significance testing with variance and confidence interval reporting.

All experiments use a temperature setting of 0.4, balancing determinism with natural response variation. Future work will explore the effects of LLM temperature on performance in enterprise knowledge extraction settings.

Table 2: RIKER Experimental Scale

| Corpus | Questions | Runs | Models | Input Tokens | Output Tokens | Total Tokens |
|---|---|---|---|---|---|---|
| *Main Experiment:* | | | | | | |
| 32K | 110 | 8 | 33 | 0.79B | 7M | 0.80B |
| 128K | 301 | 8 | 24 | 5.67B | 47M | 5.72B |
| 200K | 525 | 8 | 11 | 9.26B | 85M | 9.34B |
| *Main Subtotal* | | | | 15.72B | 139M | 15.86B |
| *Cross-Corpus Validation (Section 5.9):* | | | | | | |
| 128K (B,C,D) | 301 | 8 | 4 | 3.02B | 7M | 3.03B |
| *Expanded Hallucination Analysis (Section 5.10):* | | | | | | |
| 32K (HA,HB,HC,HD) | 300 | 8 | 10 | 2.64B | 9M | 2.65B |
| **Grand Total** | | | | **21.38B** | **155M** | **21.54B** |

*Token counts reflect total compute consumed across all experimental attempts, including a few runs that failed to produce scorable output due to API errors, timeouts, or malformed responses.*

Table 3: Corpus Document Composition

| Corpus | Leases | Field Reports | HR Reports | Total Docs |
|---|---|---|---|---|
| *Main Experiment:* | | | | |
| 32K | 10 | 44 | 56 | 110 |
| 128K (Set A) | 37 | 216 | 116 | 369 |
| 200K | 60 | 381 | 196 | 637 |
| *Cross-Corpus Validation:* | | | | |
| 128K (Set B) | 37 | 255 | 128 | 420 |
| 128K (Set C) | 37 | 228 | 120 | 385 |
| 128K (Set D) | 37 | 211 | 120 | 368 |

# 5 Results

We evaluate 33 models at 32K context, 24 models at 128K context, and 11 models at 200K context. Results are aggregated across 8 runs per model to enable statistical significance testing.

## 5.1 Overall Model Performance

Tables 4, 5, and 6 present model performance across all three context sizes. Each table reports five metrics:

- **Overall**: Weighted accuracy across all question types.
- **Single-Doc**: Accuracy on questions answerable from a single document (e.g., "What is the monthly rent in lease X?"). Tests basic retrieval and extraction.
- **Aggregation**: Accuracy on questions requiring synthesis across multiple documents (e.g., "How many leases does lessor Y have?"). Tests cross-document reasoning.
- **Hall Detect**: Accuracy on hallucination probes - correctly answering "Unknown" when asked about non-existent entities. Higher is better.
- **Hall Rate**: Hallucination rate - percentage of probes where the model fabricated an answer instead of admitting uncertainty. *Lower is better.*

Table 4: Model Performance at 32K Context (sorted by overall accuracy)

| Model | Overall (%) | Single-Doc (%) | Aggregation (%) | Hall Detect (%) | Hall Rate (%) |
|---|---|---|---|---|---|
| GLM-4.5 | 94.7 | 93.6 | 94.6 | 100.0 | 0.0 |
| GLM-4.6 | 90.5 | 93.6 | 86.6 | 89.8 | 10.2 |
| Qwen3-Next-80B | 88.6 | 91.4 | 92.6 | 59.1 | 40.9 |
| Qwen3-235B-FP8 | 87.8 | 86.1 | 95.2 | 67.0 | 33.0 |
| Qwen3-235B | 87.4 | 85.2 | 95.5 | 65.9 | 34.1 |
| Mistral-Large-3 | 86.5 | 90.2 | 85.5 | 71.6 | 28.4 |
| Qwen3-Coder-480B | 86.3 | 84.5 | 86.6 | 93.2 | 6.8 |
| Llama-4-Maverick | 85.5 | 86.6 | 87.5 | 71.6 | 28.4 |
| DeepSeek-V3.1 | 84.9 | 89.5 | 80.4 | 79.5 | 20.5 |
| Qwen2.5-72B | 82.4 | 90.0 | 75.9 | 70.5 | 29.5 |
| Qwen3-30B | 81.1 | 82.3 | 86.4 | 54.5 | 45.5 |
| GLM-4.5-Air | 78.0 | 83.9 | 66.8 | 93.2 | 6.8 |
| Qwen3-Coder-30B | 77.5 | 79.3 | 86.6 | 31.8 | 68.2 |
| Qwen3-4B-Instruct | 77.5 | 86.6 | 69.3 | 64.8 | 35.2 |
| DeepSeek-V3 | 76.1 | 79.5 | 72.4 | 73.9 | 26.1 |
| Llama-3.1-405B | 75.6 | 79.1 | 70.5 | 78.4 | 21.6 |
| Llama-4-Scout | 71.7 | 80.9 | 66.5 | 46.6 | 53.4 |
| Qwen2.5-32B | 71.4 | 83.9 | 53.1 | 81.8 | 18.2 |
| Qwen3-32B | 69.8 | 75.2 | 63.1 | 69.3 | 30.7 |
| Qwen2.5-14B | 67.5 | 75.2 | 61.4 | 53.4 | 46.6 |
| Llama-3.1-70B | 67.4 | 78.0 | 56.0 | 60.2 | 39.8 |
| Qwen2.5-Coder-14B | 63.5 | 77.7 | 51.4 | 40.9 | 59.1 |
| Llama-3.3-70B | 60.9 | 73.4 | 48.9 | 46.6 | 53.4 |
| Qwen3-8B | 60.2 | 75.5 | 38.9 | 69.3 | 30.7 |
| Qwen3-14B | 60.1 | 57.7 | 60.2 | 71.6 | 28.4 |
| Qwen3-4B | 57.7 | 76.1 | 37.8 | 45.5 | 54.5 |
| Qwen2.5-Coder-7B | 53.4 | 62.0 | 40.3 | 62.5 | 37.5 |
| Granite-4-Small | 49.7 | 63.2 | 34.9 | 40.9 | 59.1 |
| Llama-3.1-8B | 48.5 | 64.5 | 31.3 | 37.5 | 62.5 |
| Granite-4-Tiny | 38.2 | 49.5 | 31.8 | 6.8 | 93.2 |
| Llama-3.2-3B | 30.3 | 36.8 | 14.2 | 62.5 | 37.5 |
| Granite-4-Micro | 26.9 | 38.9 | 14.2 | 18.2 | 81.8 |
| Llama-3.2-1B | 9.1 | 4.5 | 3.4 | 54.5 | 45.5 |

Table 5: Model Performance at 128K Context (sorted by overall accuracy)

| Model | Overall (%) | Single-Doc (%) | Aggregation (%) | Hall Detect (%) | Hall Rate (%) |
|---|---|---|---|---|---|
| Qwen3-Next-80B | 88.5 | 94.0 | 80.8 | 90.4 | 9.6 |
| Qwen3-235B | 84.6 | 92.3 | 75.5 | 80.0 | 20.0 |
| Qwen3-235B-FP8 | 84.6 | 92.1 | 76.1 | 78.3 | 21.7 |
| GLM-4.5 | 84.4 | 94.8 | 68.5 | 92.1 | 7.9 |
| DeepSeek-V3.1 | 84.1 | 94.0 | 72.0 | 79.6 | 20.4 |
| GLM-4.6 | 81.4 | 91.4 | 69.1 | 77.1 | 22.9 |
| Qwen3-Coder-480B | 80.7 | 87.6 | 70.6 | 84.2 | 15.8 |
| Mistral-Large-3 | 75.6 | 89.4 | 61.3 | 60.0 | 40.0 |
| Qwen3-30B | 71.7 | 82.7 | 61.9 | 52.9 | 47.1 |
| DeepSeek-V3 | 70.2 | 88.3 | 46.2 | 69.6 | 30.4 |
| Llama-4-Maverick | 68.2 | 83.4 | 51.9 | 52.9 | 47.1 |
| GLM-4.5-Air | 67.3 | 84.4 | 40.5 | 82.1 | 17.9 |
| Qwen3-Coder-30B | 59.9 | 71.7 | 49.8 | 38.3 | 61.7 |
| Qwen3-4B | 57.7 | 75.0 | 32.0 | 67.5 | 32.5 |
| Llama-3.1-405B | 52.7 | 69.9 | 26.6 | 64.6 | 35.4 |
| Llama-4-Scout | 51.5 | 66.5 | 30.6 | 54.6 | 45.4 |
| Llama-3.1-70B | 45.1 | 65.2 | 17.9 | 46.7 | 53.3 |
| Llama-3.3-70B | 40.8 | 55.2 | 18.6 | 52.1 | 47.9 |
| Granite-4-Small | 39.9 | 54.0 | 15.7 | 60.4 | 39.6 |
| Llama-3.1-8B | 35.3 | 51.1 | 16.0 | 28.8 | 71.3 |
| Llama-3.2-3B | 25.9 | 33.1 | 7.1 | 61.3 | 38.8 |
| Granite-4-Tiny | 22.6 | 32.1 | 14.9 | 3.8 | 96.3 |
| Granite-4-Micro | 22.5 | 25.7 | 9.8 | 55.0 | 45.0 |
| Llama-3.2-1B | 11.3 | 7.8 | 1.6 | 67.1 | 32.9 |

Table 6: Model Performance at 200K Context (sorted by overall accuracy)

| Model | Overall (%) | Single-Doc (%) | Aggregation (%) | Hall Detect (%) | Hall Rate (%) |
|---|---|---|---|---|---|
| Qwen3-Next-80B | 77.6 | 88.8 | 59.6 | 84.8 | 15.2 |
| Qwen3-235B | 72.3 | 83.5 | 55.5 | 75.5 | 24.5 |
| Qwen3-235B-FP8 | 71.6 | 82.1 | 56.0 | 74.5 | 25.5 |
| Qwen3-Coder-480B | 71.1 | 81.4 | 55.2 | 75.9 | 24.1 |
| Mistral-Large-3 | 68.3 | 80.5 | 52.4 | 63.6 | 36.4 |
| Qwen3-30B | 65.3 | 75.1 | 54.1 | 56.4 | 43.6 |
| Llama-4-Maverick | 62.3 | 75.5 | 46.9 | 51.1 | 48.9 |
| Qwen3-Coder-30B | 52.8 | 67.9 | 35.6 | 38.2 | 61.8 |
| Llama-4-Scout | 47.6 | 62.5 | 26.5 | 47.7 | 52.3 |
| Qwen3-4B | 42.1 | 54.3 | 17.8 | 67.0 | 33.0 |
| GLM-4.6 | 34.3 | 47.3 | 19.6 | 21.4 | 78.6 |

### 5.2 Performance by Question Category

Performance varies dramatically by question type. Single-document extraction (finding a specific fact in one document) is consistently the easiest task, with top models exceeding 90%. Aggregation queries (counting, comparing across documents) prove significantly harder, with even the best model achieving only 80.8%. Hallucination detection (correctly rejecting queries about non-existent entities) reveals the starkest differences between models.

Key observations:

- Single-document extraction is consistently highest - pattern-matching often suffices
- Aggregation queries are significantly harder - requires reasoning across multiple documents
- Hallucination rates vary from 7.9% (GLM-4.5) to 96.3% (Granite-4-Tiny)
- Some models (Granite-4-Tiny) hallucinate on nearly every probe question

### 5.3 Scaling Effects

Table 7 shows performance degradation as context length increases for models tested at multiple context sizes.

Table 7: Performance Degradation Across Context Sizes (models tested at 3 sizes)

| Model | 32K (%) | 128K (%) | 200K (%) | Δ (32K→200K) |
|---|---|---|---|---|
| Qwen3-Next-80B | 88.6 | 88.5 | 77.6 | −11.0 |
| Qwen3-235B | 87.4 | 84.6 | 72.3 | −15.1 |
| Qwen3-Coder-480B | 86.3 | 80.7 | 71.1 | −15.2 |
| Qwen3-30B-A3B-Instruct | 81.1 | 71.7 | 65.3 | −15.8 |
| Qwen3-235B-FP8 | 87.8 | 84.6 | 71.6 | −16.2 |
| Mistral-Large-3 | 86.5 | 75.6 | 68.3 | −18.2 |
| Llama-4-Maverick | 85.5 | 68.2 | 62.3 | −23.2 |
| Qwen3-Coder-30B | 77.5 | 59.9 | 52.8 | −24.7 |
| Llama-4-Scout | 71.7 | 51.5 | 47.6 | −24.1 |
| Qwen3-4B | 77.5 | 57.7 | 42.1 | −35.4 |
| GLM-4.6 | 90.5 | 81.4 | 34.3 | −56.2 |

Table 8 shows models tested at 32K and 128K only.

Table 8: Performance Degradation: 32K to 128K (models not tested at 200K)

| Model | 32K (%) | 128K (%) | Δ |
|---|---|---|---|
| GLM-4.5 | 94.7 | 84.4 | −10.3 |
| DeepSeek-V3.1 | 84.9 | 84.1 | −0.8 |
| GLM-4.5-Air | 78.0 | 67.3 | −10.7 |
| DeepSeek-V3 | 76.1 | 70.2 | −5.9 |
| Llama-3.1-405B | 75.6 | 52.7 | −22.9 |
| Llama-3.1-70B | 67.4 | 45.1 | −22.3 |
| Llama-3.3-70B | 60.9 | 40.8 | −20.1 |
| Granite-4-Small | 49.7 | 39.9 | −9.8 |
| Llama-3.1-8B | 48.5 | 35.3 | −13.2 |
| Granite-4-Tiny | 38.2 | 22.6 | −15.6 |
| Llama-3.2-3B | 30.3 | 25.9 | −4.4 |
| Granite-4-Micro | 26.9 | 22.5 | −4.4 |
| Llama-3.2-1B | 9.1 | 11.3 | +2.2 |

Notable findings:

- GLM-4.6 shows dramatic collapse at 200K (−56.2% from 32K)
- Llama models degrade 13 - 23% from 32K to 128K - consistently worse than other families
- Top-tier models (Qwen3, GLM-4.5 family) lose 10-16% accuracy from 32K to 200K, while others degrade more substantially
- Llama-3.2-1B anomalously *improves* at 128K (+2.2%), though from a very low baseline (9.1%), most likely because its performance is no different than random guessing even from the low-context 32K scenario.

### 5.4 Model Family Patterns

**Qwen3 Family:** Dominates the leaderboard. Qwen3-Next-80B achieves the best overall performance (88.5% at 128K). The 235B variants show strong performance but the smaller 30B and 4B models remain competitive for their size.

**DeepSeek:** V3.1 significantly outperforms V3 (84.1% vs 70.2%), suggesting meaningful architectural or training improvements between versions.

**GLM:** GLM-4.5 excels at hallucination detection (92.1% detection rate, only 7.9% hallucination rate) but GLM-4.6 shows catastrophic failure at 200K context.

**Llama:** The Llama 3.x family underperforms relative to model size. Llama-3.1-405B (52.7%) is outperformed by much smaller Qwen models. Llama 4 marks a significant improvement - Maverick (85.5% at 32K) outperforms Llama-3.1-405B by 10 points, but still gets outperformed by much smaller Qwen models at 128K and higher.

**Granite:** IBM's Granite models struggle significantly, with the tiny variant showing 96.3% hallucination rate - fabricating answers to nearly every hallucination probe.

**Mistral**: Mistral-Large-3 performs competitively (86.5% at 32K, 68.3% at 200K) but shows steeper degradation at longer contexts than Qwen models.

### 5.5 Hallucination Analysis

We decompose model hallucination measures into three nested metrics:

- **Faithfulness** (L01 - L04 + L11 - L12): Any error where the model had sufficient information to answer correctly. Combines single-document extraction and hallucination probe questions. Higher is better.
- **Grounding** (L01 - L04): Single-document questions only - measures whether the model can find and correctly read the relevant document. Higher is better.
- **Fabrication** (L11 - L12): Hallucination probe questions using non-existent entities. Any specific answer is *definitively* fabricated since the entities exist nowhere in the corpus. Lower is better.

Aggregation questions (L05 - L10) are excluded from hallucination metrics because errors conflate grounding failures with computation errors and incoherence loss - these are synthesis failures, not hallucination in any useful sense. Aggregation performance is analyzed separately in Section 5.6.

Tables 9, 10, and 11 present the hallucination metrics across all three context sizes.

Table 9: Hallucination Metrics at 32K Context

| Model | Faith. (%) | Ground. (%) | Fab. (%) |
|---|---|---|---|
| GLM-4.5 | 94.7 | 93.6 | 0.0 |
| GLM-4.6 | 93.0 | 93.6 | 10.2 |
| DeepSeek-V3.1 | 87.9 | 89.5 | 20.5 |
| Mistral-Large-3 | 87.1 | 90.2 | 28.4 |
| Qwen2.5-72B | 86.7 | 90.0 | 29.5 |
| Qwen3-Next-80B | 86.0 | 91.4 | 40.9 |
| Qwen3-Coder-480B | 86.0 | 84.5 | 6.8 |
| GLM-4.5-Air | 85.4 | 83.9 | 6.8 |
| Llama-4-Maverick | 84.1 | 86.6 | 28.4 |
| Qwen2.5-32B | 83.5 | 83.9 | 18.2 |
| Qwen3-4B-Instruct | 83.0 | 86.6 | 35.2 |
| Qwen3-235B-FP8 | 83.0 | 86.1 | 33.0 |
| Qwen3-235B | 82.0 | 85.2 | 34.1 |
| Llama-3.1-405B | 79.0 | 79.1 | 21.6 |
| DeepSeek-V3 | 78.6 | 79.5 | 26.1 |
| Qwen3-30B-A3B-Instruct | 77.7 | 82.3 | 45.5 |
| Llama-4-Scout | 75.2 | 80.9 | 53.4 |
| Llama-3.1-70B | 75.0 | 78.0 | 39.8 |
| Qwen3-8B | 74.4 | 75.5 | 30.7 |
| Qwen3-32B | 74.2 | 75.2 | 30.7 |
| Qwen2.5-Coder-14B | 71.6 | 77.7 | 59.1 |
| Qwen2.5-14B | 71.6 | 75.2 | 46.6 |
| Qwen3-Coder-30B | 71.4 | 79.3 | 68.2 |
| Qwen3-4B | 71.0 | 76.1 | 54.5 |
| Llama-3.3-70B | 68.9 | 73.4 | 53.4 |
| Qwen2.5-Coder-7B | 62.1 | 62.0 | 37.5 |
| Llama-3.1-8B | 60.0 | 64.5 | 62.5 |
| Qwen3-14B | 60.0 | 57.7 | 28.4 |
| Granite-4-Small | 59.5 | 63.2 | 59.1 |
| Granite-4-Tiny | 42.4 | 49.5 | 93.2 |
| Llama-3.2-3B | 41.1 | 36.8 | 37.5 |
| Granite-4-Micro | 35.4 | 38.9 | 81.8 |
| Llama-3.2-1B | 12.9 | 4.5 | 45.5 |

Table 10: Hallucination Metrics at 128K Context

| Model | Faith. (%) | Ground. (%) | Fab. (%) |
|---|---|---|---|
| GLM-4.5 | 94.4 | 94.8 | 7.9 |
| Qwen3-Next-80B | 93.4 | 94.0 | 9.6 |
| DeepSeek-V3.1 | 91.6 | 94.0 | 20.4 |
| Qwen3-235B | 90.3 | 92.3 | 20.0 |
| Qwen3-235B-FP8 | 89.9 | 92.1 | 21.7 |
| GLM-4.6 | 89.1 | 91.4 | 22.9 |
| Qwen3-Coder-480B | 87.0 | 87.6 | 15.8 |
| DeepSeek-V3 | 85.3 | 88.3 | 30.4 |
| Mistral-Large-3 | 84.6 | 89.4 | 40.0 |
| GLM-4.5-Air | 84.1 | 84.4 | 17.9 |
| Llama-4-Maverick | 78.4 | 83.4 | 47.1 |
| Qwen3-30B-A3B-Instruct | 77.9 | 82.7 | 47.1 |
| Qwen3-4B | 73.8 | 75.0 | 32.5 |
| Llama-3.1-405B | 69.1 | 69.9 | 35.4 |
| Qwen3-Coder-30B | 66.3 | 71.7 | 61.7 |
| Llama-4-Scout | 64.5 | 66.5 | 45.4 |
| Llama-3.1-70B | 62.2 | 65.2 | 53.3 |
| Granite-4-Small | 55.1 | 54.0 | 39.6 |
| Llama-3.3-70B | 54.7 | 55.2 | 47.9 |
| Llama-3.1-8B | 47.5 | 51.1 | 71.3 |
| Llama-3.2-3B | 37.7 | 33.1 | 38.8 |
| Granite-4-Micro | 30.5 | 25.7 | 45.0 |
| Granite-4-Tiny | 27.5 | 32.1 | 96.3 |
| Llama-3.2-1B | 17.4 | 7.8 | 32.9 |

Table 11: Hallucination Metrics at 200K Context

| Model | Faith. (%) | Ground. (%) | Fab. (%) |
|---|---|---|---|
| Qwen3-Next-80B | 88.1 | 88.8 | 15.2 |
| Qwen3-235B | 82.2 | 83.5 | 24.5 |
| Qwen3-235B-FP8 | 80.9 | 82.1 | 25.5 |
| Qwen3-Coder-480B | 80.5 | 81.4 | 24.1 |
| Mistral-Large-3 | 77.7 | 80.5 | 36.4 |
| Qwen3-30B-A3B-Instruct | 72.0 | 75.1 | 43.6 |
| Llama-4-Maverick | 71.4 | 75.5 | 48.9 |
| Qwen3-Coder-30B | 63.0 | 67.9 | 61.8 |
| Llama-4-Scout | 60.0 | 62.5 | 52.3 |
| Qwen3-4B | 56.4 | 54.3 | 33.0 |
| GLM-4.6 | 43.0 | 47.3 | 78.6 |

**Key findings:**

- **GLM-4.5 dominates at shorter contexts:** At 32K, GLM-4.5 achieves 94.7% faithfulness with *zero* fabrication - perfect hallucination resistance. This degrades to 7.9% fabrication at 128K, still best-in-class. GLM-4.5 is not tested at 200K due to its native max context limitation.
- **Fabrication rates increase with context:** Most models show higher fabrication at longer contexts. GLM-4.6 demonstrates this dramatically: 10.2% fabrication at 32K rises to 78.6% at 200K - a 7.7× increase that explains its overall collapse.
- **Qwen3-Next-80B dominates long context:** Fabrication starts very high, but vastly decreases in higher context sizes (40.9% → 9.6% → 15.2% for 32K/128K/200K). The 32K anomaly is most likely a test measurement limitation, as the test set for the 32K corpus has fewer questions, and thus even fewer hallucination-specific questions as composition. In the 200K context test, Qwen3-Next-80B dominates, and is second only to GLM-4.5 in 128K.
- **Grounding gaps reveal fabrication:** The difference between Faithfulness and Grounding quantifies fabrication's impact. Models with high grounding but high fabrication (e.g., DeepSeek-V3 at 128K: 88.3% grounding, 30.4% fabrication) can read documents correctly but invent information when asked about non-existent entities.

**Enterprise implications:** High fabrication rates indicate models that confidently invent information when asked about non-existent entities. In production, such fabrications are indistinguishable from correct answers without ground truth verification. The 96.3% fabrication rate of Granite-4-Tiny at 128K means it would fabricate answers to nearly every query about entities not in its context - a severe reliability risk. More concerning is the fabrication increase with context length: GLM-4.6's jump from 10.2% to 78.6% fabrication suggests that models reliable at shorter contexts can become extremely unreliable at longer ones. Exhaustive testing is warranted.

### 5.6 Aggregation Analysis

Aggregation queries require models to synthesize information across multiple documents - counting entities, comparing values, or computing statistics. Unlike single-document extraction (pattern matching) or hallucination probes (fabrication detection), aggregation tests cross-document reasoning.

Table 12 presents aggregation accuracy across all three context sizes, sorted by 32K performance. Table 13 shows coherence loss rates for aggregation queries (see Section 5.7 for coherence loss discussion).

**Key findings:**

- **Aggregation is consistently harder than single-document extraction:** Even the best model (Qwen3-235B) drops from 92.3% grounding to 75.5% aggregation at 128K - a 17-point gap. Cross-document reasoning is fundamentally more difficult.
- **Aggregation degrades faster with context:** Qwen3-Next-80B drops from 92.6% (32K) to 80.8% (128K) to 59.6% (200K) - a 33-point decline. Compare this to its faithfulness decline of only 5 points across the same range.
- **Llama family struggles disproportionately:** Llama-3.1-405B achieves 79.0% faithfulness at 32K but only 70.5% aggregation. At 128K, this gap widens: 69.1% faithfulness vs 26.6% aggregation - a 42-point disparity.
- **GLM-4.6 collapse is aggregation-specific:** GLM-4.6 drops from 86.7% aggregation at 32K to 19.6% at 200K (67-point drop), while its faithfulness drops from 93.0% to 43.0% (50-point drop). The aggregation failure is even more severe.
- **Some models maintain aggregation better:** Qwen3-Coder-480B shows remarkable stability: 86.7% → 70.6% → 55.2% across 32K/128K/200K - consistent 15-16 point drops per tier, compared to the erratic collapses seen in other models.

**Enterprise implications:** Aggregation queries are common in enterprise knowledge systems: "How many contracts expire this quarter?", "What is the total revenue across all subsidiaries?", "Which employees have pending reviews?" The steep performance degradation at longer contexts suggests that production systems should either (1) keep context windows conservative for aggregation tasks, (2) use retrieval strategies that minimize context size, or (3) explicitly decompose aggregation queries into multiple single-document lookups.

Table 12: Aggregation Accuracy (sorted by 32K)

| Model | 32K | 128K | 200K |
|---|---|---|---|
| Qwen3-235B | 95.5 | 75.5 | 55.5 |
| Qwen3-235B-FP8 | 95.2 | 76.1 | 56.0 |
| GLM-4.5 | 94.6 | 68.5 | - |
| Qwen3-Next-80B | 92.6 | 80.8 | 59.6 |
| Llama-4-Maverick | 87.5 | 51.9 | 46.9 |
| GLM-4.6 | 86.7 | 69.1 | 19.6 |
| Qwen3-Coder-480B | 86.7 | 70.6 | 55.2 |
| Qwen3-Coder-30B | 86.7 | 49.8 | 35.6 |
| Qwen3-30B-A3B-Instruct | 86.4 | 61.9 | 54.1 |
| Mistral-Large-3 | 85.5 | 61.3 | 52.4 |
| DeepSeek-V3.1 | 80.4 | 72.0 | - |
| Qwen2.5-72B | 75.9 | - | - |
| DeepSeek-V3 | 72.4 | 46.2 | - |
| Llama-3.1-405B | 70.5 | 26.6 | - |
| Qwen3-4B-Instruct | 69.3 | 57.7 | 17.8 |
| GLM-4.5-Air | 66.8 | 40.5 | - |
| Llama-4-Scout | 66.5 | 30.6 | 26.5 |
| Qwen3-32B | 63.1 | - | - |
| Qwen2.5-14B | 61.4 | - | - |
| Llama-3.1-70B | 56.0 | 17.9 | - |
| Qwen2.5-32B | 53.1 | - | - |
| Llama-3.3-70B | 48.9 | 18.6 | - |
| Qwen3-8B | 38.9 | - | - |
| Qwen3-4B | 37.8 | - | - |
| Llama-3.1-8B | 31.3 | 16.0 | - |
| Granite-4-Small | 34.9 | 15.7 | - |
| Granite-4-Tiny | 31.8 | 14.9 | - |
| Granite-4-Micro | 14.2 | 9.8 | - |
| Llama-3.2-3B | 14.2 | 7.1 | - |
| Llama-3.2-1B | 3.4 | 1.6 | - |

*Values in %. Dash (-) = not tested.*

Table 13: Coherence Loss

| Model | 32K | 128K | 200K |
|---|---|---|---|
| *Severe (>10%):* | | | |
| Qwen3-4B-Instruct | 0.3 | 11.3 | 37.0 |
| Qwen3-30B-A3B-Instruct | 0.0 | 2.8 | 15.9 |
| Llama-3.1-8B | 0.3 | 13.9 | - |
| Qwen3-Coder-30B | 0.0 | 1.5 | 13.4 |
| Qwen3-Next-80B | 0.0 | 0.3 | 13.3 |
| GLM-4.6 | 0.0 | 0.4 | 11.6 |
| *Moderate (5 - 10%):* | | | |
| Llama-3.2-1B | 9.6 | 6.0 | - |
| Llama-3.2-3B | 0.0 | 7.8 | - |
| Granite-4-Tiny | 3.4 | 7.7 | - |
| Llama-3.3-70B | 0.0 | 5.4 | - |
| Qwen3-4B | 5.1 | - | - |
| *Minor (1 - 5%):* | | | |
| Granite-4-Micro | 0.0 | 4.4 | - |
| Qwen3-235B-FP8 | 0.0 | 0.5 | 3.8 |
| Qwen3-235B | 0.0 | 0.9 | 3.5 |
| GLM-4.5-Air | 0.0 | 1.8 | - |
| Granite-4-Small | 0.3 | 1.6 | - |
| Qwen3-8B | 1.4 | - | - |
| *Rare (<1%):* | | | |
| Llama-3.1-70B | 0.0 | 0.9 | - |
| Llama-3.1-405B | 0.6 | 0.4 | - |
| Qwen3-14B | 0.6 | - | - |
| Qwen3-32B | 0.6 | - | - |
| DeepSeek-V3 | 0.0 | 0.1 | - |
| DeepSeek-V3.1 | 0.0 | 0.0 | - |
| GLM-4.5 | 0.0 | 0.0 | - |
| Llama-4-Scout | 0.0 | 0.0 | 0.5 |
| Qwen3-Coder-480B | 0.0 | 0.0 | 0.0 |
| Mistral-Large-3 | 0.0 | 0.0 | 0.0 |
| Llama-4-Maverick | 0.0 | 0.0 | 0.0 |
| Qwen2.5-72B | 0.0 | - | - |
| Qwen2.5-32B | 0.0 | - | - |
| Qwen2.5-14B | 0.0 | - | - |
| Qwen2.5-Coder-14B | 0.0 | - | - |
| Qwen2.5-Coder-7B | 0.0 | - | - |

*Values in %. Dash (-) = not tested.*

### 5.7 Coherence Loss and Infinite Generation

Under test conditions, our examined models have anywhere from 25K to >100K max output tokens available (depending on the model's capacity and the test set). When models hit their maximum token limit before completing a response, this typically indicates an *infinite generation loop* caused by coherence loss. Our benchmarking infrastructure truncates the response, records it, and moves on to the next item. We track truncation rates as a proxy for coherence loss under long-context pressure. Table 13 shows coherence loss organized by severity tier.

**Key observations from Table 13:**

- **Coherence loss correlates with model size:** Smaller models (Qwen3-4B, Llama-3.1-8B, Granite-4-Tiny) show higher coherence loss rates. The largest models (Qwen3-Coder-480B, Qwen3-235B, DeepSeek-V3.1) rarely devolve into incoherence.
- **Coherence loss increases with context length:** Qwen3-4B-Instruct goes from 0.3% at 32K to 37.0% at 200K - a 123× increase.
- **Aggregation is uniquely vulnerable:** Single-document and hallucination queries rarely trigger coherence loss (<0.5%

for most models). Aggregation queries require the model to enumerate and reason across multiple documents, which appears to trigger generation loops.

- **GLM-4.6's 200K collapse has multiple causes:** Beyond the 78.6% fabrication rate noted earlier, GLM-4.6 also shows 11.6% coherence loss frequency at 200K - the model fails in multiple ways simultaneously.

**Note: Hardware-dependent failures observed distinct from coherence loss:** Llama 4 models (Maverick and Scout) show near-zero truncation rates, but exhibited a different failure mode on MI300X hardware (producing only ASCII replacement characters in an infinite generation loop) and Gaudi 3 (real characters, but 100% coherence loss). Successful runs were only possible from our H200 hardware. This infrastructure sensitivity represents a separate reliability dimension not captured by coherence metrics. We note this here for completeness. When running properly, Llama 4 models appear less prone to infinite generation. When they don't, they appear completely broken under RIKER tests - a contradiction the coherence loss metrics in Table 13 cannot capture. None of the other 31 models tested had similar characteristics.

### 5.8 Examining the GLM 4.6 200K Collapse

By all measures - accuracy, hallucination metrics, coherence loss - GLM 4.6 showed a steep decline in output quality in the 200K tests. We investigated whether an adjustment in temperature would change this catastrophic performance, particularly the coherence loss.

Table 14: GLM-4.6 at 200K: Temperature 0.4 vs 1.0

| Metric | temp=0.4 | temp=1.0 |
|---|---|---|
| Coherence Loss | 184 | 4 |
| Coherence Loss Rate | 4.38% | 0.10% |
| Overall Accuracy | 37% | 27% |

Higher temperature dramatically reduces coherence loss (184 $\rightarrow$ 4 instances). Surprisingly, accuracy *decreased* from 37% to 27%.

The role of temperature in enterprise-relevant scenarios like long-context knowledge extraction (this study) or in overall agentic fitness (measured in our agentic merit index paper [50] with deeper analysis in our trace-level agentic AI failure study [48]) deserves a more intensive treatment and will be examined in much deeper detail in future work.

### 5.9 Cross-Corpus Benchmark Stability Analysis

A key claim of RIKER is that the Coherent Simulated Universe approach measures *model capability* rather than corpus-specific artifacts. To validate this, we generated three additional 128K corpora (Sets B, C, D) using identical generation parameters to the original (Set A) - same entity counts, document distributions, and question type ratios - but with independent random seeds producing entirely different documents, entities, and questions.

We ran four models spanning the performance spectrum across all three new corpora: DeepSeek-V3.1 and GLM-4.5 (top tier), Qwen3-Coder-480B (upper-middle tier), and Granite-4-Small (lower tier). Table 15 presents the results.

Table 15: Cross-Corpus Stability: Model Performance Across Independent 128K Corpora

| Model | Set A (%) | Set B (%) | Set C (%) | Set D (%) | Mean (%) | Spread (pts) |
|---|---|---|---|---|---|---|
| DeepSeek-V3.1 | 84.1 | 83.9 | 84.4 | 82.9 | 83.8 | 1.5 |
| GLM-4.5 | 84.4 | 83.0 | 83.6 | 84.4 | 83.8 | 1.5 |
| Qwen3-Coder-480B | 80.7 | 78.9 | 79.9 | 81.4 | 80.2 | 2.5 |
| Granite-4-Small | 39.9 | 36.3 | 33.8 | 34.9 | 36.2 | 6.1 |

*Note: Set A is the original corpus used throughout this paper. Sets B, C, D are independently generated validation corpora. Spread is max minus min across all tested sets.*

**Key findings:**

- **Top-tier models show remarkable stability:** DeepSeek-V3.1 and GLM-4.5 vary by only 1.4-1.5 percentage points across four completely independent corpora. This spread is comparable to run-to-run variance within a single corpus.

- **Mid-tier stability is also strong:** Qwen3-Coder-480B shows 2.5 points of spread - slightly higher but still within normal measurement noise for 8-run experiments.
- **Weaker models show higher corpus sensitivity:** Granite-4-Small exhibits 6.1 points of spread. However, its relative ranking is preserved - it remains clearly in the lower performance tier across all corpora. This suggests that weaker models may be more sensitive to specific document phrasings or entity configurations.
- **Ranking stability matters most:** The practical question for benchmarks is whether relative model rankings are preserved. Across all four corpora, the ordering DeepSeek-V3.1 $\approx$ GLM-4.5 ¿ Qwen3-Coder-480B ¿¿ Granite-4-Small remains consistent.

These results show a simple validation of RIKER's core design. Because ground truth is generated *before* documents, and documents are generated *from* that ground truth, any corpus instantiation with the same parameters measures the same underlying capability. The benchmark is not testing whether models memorized specific phrasings or entity names - it is testing whether they can extract structured information from realistic enterprise documents. This property enables contamination-resistant evaluation: even if a specific corpus leaks, regenerating with the same parameters produces an equally valid benchmark.

### 5.10 Expanded Hallucination Analysis

Our main experiment includes hallucination questions (L11-L12) as part of the broader question taxonomy. To enable deeper analysis of hallucination behavior, we conducted an expanded study with 300 questions per evaluation: 150 grounding questions (L1-L4) testing extraction of facts that *do* exist in the corpus, and 150 fabrication questions (L11-L12) testing whether models correctly identify facts that do *not* exist.

We evaluated 10 models across four independent question sets (HA, HB, HC, HD) on the same 128K corpus, with 8 runs per configuration. This design serves two purposes: (1) deeper hallucination analysis with balanced grounding/fabrication questions, and (2) within-corpus stability validation using different question sets on identical documents.

Table 16: Expanded Hallucination Analysis: Grounding vs. Fabrication Performance

| Model | Grounding (L1-L4) | Fabrication (L11-L12) | Overall Accuracy |
|---|---|---|---|
| GLM-4.5 | 96.7% | 2.1% | 97.3% |
| GLM-4.6 | 93.8% | 9.7% | 92.1% |
| DeepSeek-V3.1 | 95.1% | 17.3% | 88.9% |
| Qwen3-235B | 92.7% | 15.5% | 88.6% |
| Qwen3-235B-FP8 | 92.7% | 16.5% | 88.1% |
| Qwen3-Coder-480B | 90.1% | 18.4% | 85.9% |
| Qwen3-Next-80B | 91.4% | 19.9% | 85.7% |
| Llama-3.1-70B | 86.3% | 38.8% | 73.7% |
| Llama-3.3-70B | 81.4% | 45.0% | 68.2% |
| Granite-4-Small | 67.5% | 54.5% | 56.5% |

*Grounding measures accuracy on facts present in corpus. Fabrication shows hallucination rate on non-existent facts (lower is better). Results averaged across 4 question sets $\times$ 8 runs = 32 evaluations per model.*

**Key findings:**

- **GLM-4.5 exhibits exceptional hallucination resistance:** With only 2.1% fabrication rate, GLM-4.5 almost never claims to find information that does not exist. This is remarkable given that the questions are designed to be plausible (asking about entities that *could* exist but don't).
- **Grounding ability does not predict fabrication resistance:** DeepSeek-V3.1 achieves 95.1% grounding accuracy (second only to GLM-4.5) but has 17.3% fabrication rate - 8$\times$ higher than GLM-4.5. Models can be excellent at finding facts that exist while still being prone to "finding" facts that don't.
- **Llama models show elevated hallucination rates:** Both Llama-3.1-70B (38.8%) and Llama-3.3-70B (45.0%) exhibit substantially higher fabrication rates than similarly-sized models. Notably, the newer Llama-3.3 performs *worse* on hallucination resistance than Llama-3.1.
- **Quantization has minimal impact:** Qwen3-235B and its FP8 quantized variant show nearly identical performance (15.5% vs 16.5% fabrication), suggesting that quantization does not significantly affect hallucination behavior for this

model. However, not all models and quantization methods are equal. Future work will examine the effects of quantization on enterprise reliability more broadly.

**Within-corpus stability:** Table 17 shows fabrication rates across the four independent question sets. Most models exhibit stable behavior: GLM-4.5 shows only 1.4 percentage points spread, while DeepSeek-V3.1 and Qwen3-Coder-480B show 2.0 points each. Notably, Granite-4-Small has the smallest spread (0.7 pts) despite having the worst absolute performance - it consistently hallucinates at the same high rate regardless of which questions are asked.

Table 17: Within-Corpus Stability: Fabrication Rate Across Question Sets

| Model | HA | HB | HC | HD | Spread |
|---|---|---|---|---|---|
| GLM-4.5 | 2.0% | 1.2% | 2.7% | 2.5% | 1.4 pts |
| GLM-4.6 | 9.9% | 7.9% | 10.7% | 10.2% | 2.8 pts |
| DeepSeek-V3.1 | 17.7% | 18.3% | 17.0% | 16.3% | 2.0 pts |
| Qwen3-235B | 11.2% | 18.5% | 16.2% | 16.0% | 7.3 pts |
| Qwen3-235B-FP8 | 14.0% | 19.8% | 16.5% | 15.7% | 5.8 pts |
| Qwen3-Coder-480B | 18.2% | 19.6% | 18.2% | 17.6% | 2.0 pts |
| Qwen3-Next-80B | 18.0% | 19.2% | 20.8% | 21.7% | 3.7 pts |
| Llama-3.1-70B | 38.0% | 37.5% | 38.5% | 41.3% | 3.8 pts |
| Llama-3.3-70B | 42.8% | 45.1% | 43.5% | 48.5% | 5.7 pts |
| Granite-4-Small | 54.3% | 54.9% | 54.2% | 54.7% | 0.7 pts |

*Fabrication rate (%, lower is better) for each of four independent question sets on the same 128K corpus. Spread = max - min across sets. Each cell averages 8 runs.*

Taken together with the findings in Section 5.9, this confirms the feasibility and practicality of RIKER's regenerable corpora and regenerable test set features for contamination resistance. Both approaches show stable measurements, suggesting the measurement of capabilities instead of particular question-specific artifacts.

## 6 Discussion

Our evaluation reveals several findings with implications for both researchers and practitioners deploying LLMs in enterprise contexts. Before discussing specific results, we distinguish between RIKER as an implementation and as a methodology - the latter being the more generalizable contribution.

### 6.1 RIKER as Methodology

RIKER is one implementation of a more general approach: *ground-truth-first synthetic evaluation*. The core methodology consists of three principles:

1. **Ground truth precedes documents.** Define the facts, entities, and relationships first. Generate documents that embed these facts second. This inverts the traditional approach of extracting ground truth from existing documents.

2. **Questions derive from ground truth.** Generate questions whose answers are known by construction, not by human annotation. This enables deterministic scoring without reference models or human judges.

3. **Regenerability enables contamination resistance.** Because documents and questions are procedurally generated, fresh instances can be created at will. Benchmark validity does not depend on any particular corpus remaining unseen.

The effectiveness of regenerability depends on implementation quality: document distinctiveness requires sufficient randomization of templates, entity pools, and variable combinations. A RIKER-based document generator that lacks sufficient document diversity and distinctiveness due to limited randomization provides little to no protection against contamination.

Beyond contamination resistance, ground-truth-first generation enables unprecedented scale. Traditional benchmarks require human annotation for each test item, making large-scale evaluation expensive. Many recent benchmarks substitute LLM-as-a-judge for human annotation, but this introduces judge model biases, adds inference cost and increases overall evaluation time. RIKER's automatic ground truth and deterministic scoring eliminate both bottlenecks - our 21+ billion token evaluation would be prohibitively expensive with human annotation and impractically slow with LLM judges. Deterministic scoring also ensures perfect reproducibility: the same response always receives the same score.

Our specific implementation to demonstrate the methodology uses commercial leases, sales agent field reports, and HR records - but these are incidental. The methodology applies wherever synthetic documents can be generated from structured

ground truth: medical records from patient databases, legal documents from case parameters, scientific papers from experimental results, financial reports from transaction logs.

The stability validations (Sections 5.9 and 5.10) confirm that this methodology produces reliable measurements. Cross-corpus stability shows that different document instantiations yield consistent rankings. Within-corpus stability shows that different question sets on identical documents yield consistent results. These properties should transfer to any properly implemented ground-truth-first benchmark, not just our specific implementation in RIKER.

### 6.2 Cross-Corpus and Within-Corpus Stability

RIKER's core design hypothesis - that generating documents FROM ground truth rather than extracting ground truth FROM documents enables reliable evaluation - is supported by the stability analyses. Cross-corpus validation (Section 5.9) shows that different document instantiations with identical parameters produce consistent measurements: models that perform well on Set A perform well on Sets B, C, and D. Within-corpus validation (Section 5.10) shows that different question sets on the same corpus also yield stable results. Together, these findings confirm that RIKER measures model capability rather than corpus-specific or question-specific artifacts.

This addresses the fundamental contamination problem identified in our gap analysis. Unlike static benchmarks where performance may reflect memorization of specific examples, RIKER's regenerable corpora create fresh, never-before-seen documents while maintaining the same underlying evaluation parameters. The benchmark can be regenerated indefinitely without losing validity.

### 6.3 Context Length Claims Exceed Usable Capacity

Models frequently advertise 128K or 200K token context windows, but our results suggest these claims overstate usable capacity for enterprise tasks. Top-tier models achieve over 80% accuracy at 32K tokens but show meaningful degradation at longer contexts. Some models exhibit catastrophic failures: GLM-4.6 collapses from 70.9% accuracy at 128K to 26.6% at 200K, despite nominally supporting both context lengths.

For practitioners, this implies that marketed context length should not be conflated with effective context length. Enterprise deployments requiring long-context processing should validate performance at their actual operating context sizes rather than relying on vendor specifications.

### 6.4 Aggregation Reveals a Capability Gap

Single-document extraction and cross-document aggregation appear to require fundamentally different capabilities. Models that excel at extracting facts from individual documents often struggle when required to aggregate information across multiple documents. This is not simply a matter of "more work" - aggregation questions require the model to identify relevant information scattered across the corpus, perform accurate extraction from each source, and combine results correctly.

RIKER's multi-level taxonomy explicitly separates these capabilities, revealing that aggregation accuracy consistently lags extraction accuracy across all models tested. This finding validates the taxonomy design and suggests that benchmarks testing only single-document retrieval may overestimate model capability for realistic enterprise workloads that inherently involve multi-document reasoning.

### 6.5 Grounding and Hallucination Resistance Are Distinct

Perhaps our most striking finding is that strong grounding performance does not predict hallucination resistance. DeepSeek-V3.1 achieves 95.1% grounding accuracy - second only to GLM-4.5 - yet exhibits a 17.3% fabrication rate, eight times higher than GLM-4.5's 2.1%. Models can be excellent at finding information that exists while simultaneously prone to "finding" information that does not exist.

This distinction has significant implications for enterprise deployment. A model that reliably extracts correct answers when information is present may still confidently fabricate answers when queried about non-existent entities. Applications requiring high factual reliability must evaluate both capabilities independently.

### 6.6 Limitations

RIKER's current implementation has several limitations that constrain generalizability:

**Domain scope.** Our evaluation uses enterprise documents (commercial leases, facility field reports, HR records). While chosen for realism, performance on these document types may not transfer to other domains such as scientific literature, legal contracts, or medical records.

**Language.** All documents and questions are in English. Multilingual performance remains untested.

**Architecture.** In this study, we evaluate pure context-stuffing - the entire corpus is provided in the prompt. Future work will test retrieval-augmented generation and agentic retrieval patterns.

**Synthetic realism.** While our Coherent Simulated Universe approach maintains entity consistency across documents, synthetic documents may lack certain characteristics of real-world data: OCR errors, inconsistent formatting, contradictory information across sources, or domain-specific jargon. Models may perform differently on messier real-world corpora. These characteristics can be modeled into RIKER's (or any RIKER-like system's) synthetic data generation logic, but in this current work and results, such characteristics are not included.

**Model coverage.** Our evaluation covers 33 models available at the time of testing. The rapidly evolving LLM landscape means new models may exhibit different patterns.

### 6.7 Future Work

Several extensions would strengthen RIKER's utility:

**Additional document types.** Expanding beyond the current three document types to include contracts, technical specifications, financial statements, and other enterprise documents would broaden applicability.

**Additional test types.** For hallucination detection in particular, more types of hallucination-specific test types can be added that offer a finer-grained view of hallucination failure modes. Instead of single-answer questions (entity name or amount value), questions that expect lists of specific items (e.g., document sources paired with specific retrieved details each) can provide a more comprehensive hallucination-centered metric.

**RAG and agentic evaluation.** Instrumenting retrieval systems to answer RIKER questions would enable direct comparison between context-stuffing and various RAG approaches on identical ground truth.

**Multilingual corpora.** Generating documents in multiple languages would enable cross-lingual evaluation.

**Adversarial robustness.** Introducing deliberate inconsistencies or contradictions in the corpus could test model behavior when ground truth is ambiguous.

**Continuous benchmarking.** RIKER's regenerable design enables ongoing evaluation as new models emerge, potentially as an automated leaderboard with fresh corpora for each evaluation cycle.

## 7 Conclusion

We presented RIKER, both as a benchmark for enterprise document understanding and as a demonstration of a more general methodology: *ground-truth-first synthetic evaluation*. The core insight - generating documents FROM known ground truth rather than extracting ground truth FROM documents - enables deterministic scoring, contamination resistance through regenerable corpora, and systematic evaluation of capabilities that matter for real-world deployment.

Our evaluation of 33 models using over 21 billion tokens reveals several key findings. Context length claims frequently exceed usable capacity, with significant performance degradation beyond 32K tokens. Cross-document aggregation proves substantially harder than single-document extraction. Grounding ability and hallucination resistance are distinct capabilities that must be evaluated separately.

The stability analyses confirm that the methodology works: both cross-corpus validation (different documents, same parameters) and within-corpus validation (same documents, different questions) produce consistent measurements. This suggests that any properly implemented ground-truth-first benchmark should exhibit similar stability properties, regardless of domain.

Beyond the specific benchmark, we contribute a replicable methodology for constructing contamination-resistant evaluations in any domain where synthetic documents can be generated from structured ground truth.

## Data Availability

The experiment data, including all of the generated ground truth, document corpora, test sets, and the various model raw results, will be made available at `https://docs.kamiwaza.ai/research/datasets`.

## Acknowledgments

This research was made possible through the generous provision of GPU compute by Signal65, who provided four servers with 8x AMD MI300X GPUs each for experimental evaluation. We thank Ryan Shrout, Brian Martin, Mitch Lewis, and Russ Fellows for their support. (`https://signal65.com/`)

## AI Usage Disclosure

The researchers used the following generative AI services to assist with the manuscript:

- Claude Code: Claude Opus 4.5
- Gemini 3 Pro Image (Nano Banana Pro)

Most of the language in this paper was drafted by generative AI using RIKER project documents and code, plus certain reference material (especially previous work in PICARD) and the raw and summarized RIKER results in CSV form, and then heavily revised, edited and polished by human researchers. 100% of this document was read and reviewed several times by human researchers.

In addition, all tables were created through generative AI directly using raw source data, and then reviewed by human researchers. Nano Banana Pro was used to generate the RIKER diagram by feeding it the final draft of this paper.

In all cases, final editorial control, technical validation, and intellectual responsibility rest solely with the human authors. The authors take full responsibility for the accuracy and integrity of all content in this manuscript.

## References


[1] Wasi Uddin Ahmad. Awesome LLM synthetic data: A reading list on LLM based synthetic data generation, 2024.

[2] Amazon Web Services. Ground truth generation and review best practices for evaluating generative AI question-answering with FMEval, 2024.

[3] Appen. State of AI in 2024 report, 2024.

[4] Abhinav Arun, Fabrizio Dimino, Tejas Prakash Agarwal, Bhaskarjit Sarmah, and Stefano Pasquali. FinReflectKG: Agentic construction and evaluation of financial knowledge graphs. *arXiv preprint arXiv:2508.17906*, 2025.

[5] Yushi Bai, Shangqing Tu, Jiajie Zhang, Hao Peng, Xiaozhi Wang, Xin Lv, Shulin Cao, Jiazheng Xu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longbench v2: Towards deeper understanding and reasoning on realistic long-context multitasks, 2025.

[6] Krisztian Balog and ChengXiang Zhai. User simulation in the era of generative AI: User modeling, synthetic data generation, and system evaluation, 2025.

[7] Yejin Bang, Ziwei Ji, Alan Schelten, Anthony Hartshorn, Tara Fowler, Cheng Zhang, Nicola Cancedda, and Pascale Fung. Hallulens: Llm hallucination benchmark, 2025.

[8] Beatrust Tech Blog. RAG evaluation: Assessing the usefulness of RAGAS, 2024.

[9] Andres M. Bran, Zlatko Jončev, and Philippe Schwaller. Knowledge graph extraction from total synthesis documents. In *Proceedings of the 1st Workshop on Language + Molecules (L+M 2024)*, pages 74–84, Bangkok, Thailand, 2024. Association for Computational Linguistics.

[10] Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. Humans or llms as the judge? a study on judgement biases, 2024.

[11] Seungmin Choi and Yuchul Jung. Knowledge graph construction: Extraction, learning, and evaluation. *Applied Sciences*, 15(7):3727, 2025.

[12] Collective Intelligence Project. LLM judges are unreliable, 2024.

[13] Databricks. FreshStack: Building realistic benchmarks for evaluating retrieval on technical documents, 2025.

[14] DavidZWZ. Awesome RAG reasoning: Resources for RAG reasoning in LLMs and agents, 2025.

[15] DEEP-PolyU. Awesome-GraphRAG: A curated list of resources on graph-based retrieval-augmented generation, 2024.

[16] Chunyuan Deng, Yilun Zhao, Xiangru Tang, Mark Gerstein, and Arman Cohan. Investigating data contamination in modern benchmarks for large language models, 2024.

[17] Yihong Dong, Xue Jiang, Huanyu Liu, Zhi Jin, Bin Gu, Mengfei Yang, and Ge Li. Generalization or memorization: Data contamination and trustworthy evaluation for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*. Association for Computational Linguistics, 2024.

[18] Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. From local to global: A graph RAG approach to query-focused summarization. *arXiv preprint arXiv:2404.16130*, 2024.

[19] Maria Eriksson, Erasmo Purificato, Arman Noroozian, Joao Vinagre, Guillaume Chaslot, Emilia Gomez, and David Fernandez-Llorca. Can we trust AI benchmarks? an interdisciplinary review of current issues in AI evaluation. *arXiv preprint*, 2025.

[20] Shahul Es, Jithin James, Luis Espinosa-Anke, and Steven Schockaert. RAGAS: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, 2024.

[21] Aoran Gan, Hao Yu, Kai Zhang, Qi Liu, Wenyu Yan, Zhenya Huang, Shiwei Tong, and Guoping Hu. Retrieval augmented generation evaluation in the era of large language models: A comprehensive survey, 2025.

[22] Jiaxin Gao, Chen Chen, Yanwen Jia, Xueluan Gong, Kwok-Yan Lam, and Qian Wang. Evaluating and mitigating llm-as-a-judge bias in communication systems, 2025.

[23] Yunfan Gao, Yun Xiong, Wenlong Wu, Zijing Huang, Bohan Li, and Haofen Wang. U-niah: Unified rag and llm evaluation for long context needle-in-a-haystack, 2025.

[24] GraphRAG-Bench Team. GraphRAG-Bench: A comprehensive benchmark for evaluating GraphRAG models, 2025.

[25] Haoyu Han, Li Ma, Harry Shomer, Yu Wang, Yongjia Lei, Kai Guo, Zhigang Hua, Bo Long, Hui Liu, Charu C. Aggarwal, and Jiliang Tang. Rag vs. graphrag: A systematic evaluation and key insights, 2025.

[26] Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, 2020.

[27] Zijin Hong, Hao Wu, Su Dong, Junnan Dong, Yilin Xiao, Yujing Zhang, Zhu Wang, Feiran Huang, Linyi Li, Hongxia Yang, and Xiao Huang. Benchmarking LLMs' mathematical reasoning with unseen random variables questions. *arXiv preprint arXiv:2501.11790*, 2025.

[28] Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. Ruler: What's the real context size of your long-context language models?, 2024.

[29] Hugging Face. The hallucinations leaderboard, an open effort to measure hallucinations in large language models, 2024.

[30] Hamel Husain. Modern IR evals for RAG, 2024.

[31] Gregory Kamradt. LLMTest_NeedleInAHaystack: Doing simple retrieval from LLM models at various context lengths to measure accuracy, 2023.

[32] Ernests Lavrinovics, Russa Biswas, Johannes Bjerva, and Katja Hose. Knowledge graphs, large language models, and hallucinations: An NLP perspective. *Web Semantics: Science, Services and Agents on the World Wide Web*, 85:100844, 2024.

[33] Sunwoo Lee, Daseong Jang, Dhammiko Arya, Gyoung-eun Han, Injee Song, SaeRom Kim, Sangjin Kim, Seojin Lee, Seokyoung Hong, Sereimony Sek, Seung-Mo Cho, Sohee Park, Sungbin Yoon, Wonbeom Jang, and Eric Davis. TelAgentBench: A multi-faceted benchmark for evaluating LLM-based agents in telecommunications. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1173–1211, Suzhou, China, 2025.

[34] Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2023.

[35] Yucheng Li, Frank Guerin, and Chenghua Lin. LatestEval: Addressing data contamination in language model evaluation through dynamic and time-sensitive test construction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024.

[36] Jintao Liang, Gang Su, Huifeng Lin, You Wu, Rui Zhao, and Ziyue Li. Reasoning rag via system 1 or system 2: A survey on reasoning agentic retrieval-augmented generation for industry challenges, 2025.

[37] Jingru Lin, Chen Zhang, Stephen Y. Liu, and Haizhou Li. Ragcap-bench: Benchmarking capabilities of llms in agentic retrieval augmented generation systems, 2025.

[38] Hao Liu, Zhengren Wang, Xi Chen, Zhiyu Li, Feiyu Xiong, Qinhan Yu, and Wentao Zhang. Hoprag: Multi-hop reasoning for logic-aware retrieval-augmented generation, 2025.

[39] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024.

[40] Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. On llms-driven synthetic data generation, curation, and evaluation: A survey, 2024.

[41] lyy1994. Awesome data contamination: The paper list on data contamination for large language models evaluation, 2024.

[42] Gaurav Maheshwari, Dmitry Ivanov, and Kevin El Haddad. Efficacy of synthetic data as a benchmark. *arXiv preprint arXiv:2409.11968*, 2024.

[43] Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2023.

[44] Sewon Min, Eric Wallace, Sameer Singh, Matt Gardner, Hannaneh Hajishirzi, and Luke Zettlemoyer. Compositional questions do not necessitate multi-hop reasoning, 2019.

[45] Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. GSM-Symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*, 2024.

[46] Hyeonseok Moon and Heuiseok Lim. Needlechain: Measuring intact long-context reasoning capability of large language models, 2025.

[47] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. Mteb: Massive text embedding benchmark, 2023.

[48] JV Roig. How do llms fail in agentic scenarios? a qualitative analysis of success and failure scenarios of various llms in agentic simulations, 2025.

[49] JV Roig. Testing what models can do, not what they've seen: Picard: Probing intelligent capabilities via artificial randomized data. Technical report, Kamiwaza AI, 2025.

[50] JV Roig. Towards a standard, enterprise-relevant agentic ai benchmark: Lessons from 5.5 billion tokens' worth of agentic ai evaluations, 2025.

[51] Sujoy Roychowdhury, Sumit Soman, H. G. Ranjani, Neeraj Gunda, Vansh Chhabra, and Sai Krishna Bala. Evaluation of RAG metrics for question answering in the telecom domain. *arXiv preprint arXiv:2407.12873*, 2024.

[52] Dongyu Ru, Lin Qiu, Xiangkun Hu, Tianhang Zhang, Peng Shi, Shuaichen Chang, Cheng Jiayang, Cunxiang Wang, Shichao Sun, Huanyu Li, Zizhao Zhang, Binjie Wang, Jiarong Jiang, Tong He, Zhiguo Wang, Pengfei Liu, Yue Zhang, and Zheng Zhang. Ragchecker: A fine-grained framework for diagnosing retrieval-augmented generation, 2024.

[53] Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 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 (NAACL)*. Association for Computational Linguistics, 2024.

[54] Olawale Salaudeen, Anka Reuel, Ahmed Ahmed, Suhana Bedi, Zachary Robertson, Sudharsan Sundar, Ben Domingue, Angelina Wang, and Sanmi Koyejo. Measurement to meaning: A validity-centered framework for AI evaluation. *arXiv preprint arXiv:2505.10573*, 2025.

[55] Hannah Sansford, Nicholas Richardson, Hermina Petric Maretic, and Juba Nait Saada. Grapheval: A knowledge-graph based llm hallucination evaluation framework, 2024.

[56] Julian Schnitzler, Xanh Ho, Jiahao Huang, Florian Boudin, Saku Sugawara, and Akiko Aizawa. MoreHopQA: More than multi-hop reasoning, 2024.

[57] Lin Shi, Chiyu Ma, Wenhua Liang, Xingjian Diao, Weicheng Ma, and Soroush Vosoughi. Judging the judges: A systematic study of position bias in llm-as-a-judge, 2025.

[58] James Steinhoff and Sam Hind. Simulation and the reality gap: Moments in a prehistory of synthetic data. *Big Data & Society*, 12(1):1–14, 2025.

[59] Yixuan Tang and Yi Yang. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries, 2024.

[60] Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, 2021.

[61] Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. MuSiQue: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 2022.

[62] Yingjia Wan, Haochen Tan, Xiao Zhu, Xinyu Zhou, Zhiwei Li, Qingsong Lv, Changxuan Sun, Jiaqi Zeng, Yi Xu, Jianqiao Lu, Yinhong Liu, and Zhijiang Guo. FaStfact: Faster, stronger long-form factuality evaluations in LLMs, 2025.

[63] Hengyi Wang, Haizhou Shi, Shiwei Tan, Weiyi Qin, Wenyuan Wang, Tunyu Zhang, Akshay Nambi, Tanuja Ganu, and Hao Wang. Multimodal needle in a haystack: Benchmarking long-context capability of multimodal large language models, 2025.

[64] Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models, 2024.

[65] Zhishang Xiang, Chuanjie Wu, Qinggang Zhang, Shengyuan Chen, Zijin Hong, Xiao Huang, and Jinsong Su. When to use graphs in rag: A comprehensive analysis for graph retrieval-augmented generation, 2025.

[66] Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. Benchmarking retrieval-augmented generation for medicine, 2024.

[67] Cheng Xu, Shuhao Guan, Derek Greene, and M-Tahar Kechadi. Benchmark data contamination of large language models: A survey, 2024.

[68] Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E. Gonzalez, and Ion Stoica. Rethinking benchmark and contamination for language models with rephrased samples. *arXiv preprint arXiv:2311.04850*, 2023.

[69] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[70] Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, Nitesh V Chawla, and Xiangliang Zhang. Justice or prejudice? quantifying biases in llm-as-a-judge, 2024.

[71] Hao Yu, Aoran Gan, Kai Zhang, Shiwei Tong, Qi Liu, and Zhaofeng Liu. Evaluation of retrieval-augmented generation: A survey, 2024.

[72] Yifei Yu, Qian-Wen Zhang, Lingfeng Qiao, Di Yin, Fang Li, Jie Wang, Zengxi Chen, Suncong Zheng, Xiaolong Liang, and Xing Sun. Sequential-niah: A needle-in-a-haystack benchmark for extracting sequential needles from long contexts, 2025.

[73] Qiming Zeng, Xiao Yan, Hao Luo, Yuhao Lin, Yuxiang Wang, Fangcheng Fu, Bo Du, Quanqing Xu, and Jiawei Jiang. How significant are the real performance gains? an unbiased evaluation framework for graphrag, 2025.

[74] Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Khai Hao, Xu Han, Zhen Leng Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. $\infty$bench: Extending long context evaluation beyond 100k tokens, 2024.

[75] Zhenyu Zhang, Runjin Chen, Shiwei Liu, Zhewei Yao, Olatunji Ruwase, Beidi Chen, Xiaoxia Wu, and Zhangyang Wang. Found in the middle: How language models use long contexts better via plug-and-play positional encoding, 2024.

[76] Qihao Zhao, Yangyu Huang, Tengchao Lv, Lei Cui, Qinzheng Sun, Shaoguang Mao, Xin Zhang, Ying Xin, Qiufeng Yin, Scarlett Li, and Furu Wei. Mmlu-cf: A contamination-free multi-task language understanding benchmark, 2024.