Generating Leakage-Free Benchmarks for Robust RAG Evaluation

Jiayi Liu*[1], Jiaxing Zhang[2], Bowen Jin[3], and Jennifer Neville*[1,4]

[1]Department of Computer Science, Purdue University
[2]New Jersey Institute of Technology
[3]University of Illinois at Urbana-Champaign
[4]Microsoft Research
{liu2861, neville}@purdue.edu, jz48@njit.edu, bowenj4@illinois.edu

Abstract

Retrieval-augmented generation (RAG) is widely used to augment large language models (LLMs) with external knowledge. However, many benchmark datasets, designed to test RAG performance, comprise many questions that can already be answered from an LLM's parametric memory. This leads to unreliable evaluation. We refer to this phenomenon as knowledge leakage—cases where RAG tasks are solvable without retrieval. This issue worsens over time due to benchmark aging. As benchmarks are reused for training, their contents are increasingly absorbed into model parameters, making them less effective for evaluating retrieval.

We introduce **SeedRG**, a semi-synthetic benchmark generation pipeline that mitigates knowledge leakage and addresses the issue of benchmark aging. Starting from a seed benchmark dataset, SeedRG extracts a reasoning graph from question–context pairs to capture their underlying reasoning structure, and then generates new examples via type-constrained entity replacement. This process produces structurally similar but novel instances that are unlikely to exist in the model's parametric knowledge, while preserving the original reasoning patterns. To ensure quality, we incorporate two verification steps: (1) a reasoning-graph consistency check to maintain task difficulty, and (2) a knowledge-leakage filter to exclude instances answerable without retrieval.

We evaluate SeedRG on three seed benchmarks (HotpotQA, 2WikiMultihopQA, QASC) and three popular LLMs (GPT-5, Claude Sonnet 4.5, Gemini 2.5 Flash). SeedRG reduces knowledge leakage by at least 78% while preserving reasoning difficulty. By removing the confounding effect of parametric knowledge, SeedRG reveals meaningful variability across RAG systems that is otherwise obscured. Prior benchmarks show uniformly high performance across RAG methods (HippoRAG, GraphRAG, OGRAG, SemanticRAG), because performance is dominated by model knowledge. In contrast, SeedRG surfaces clear differences in retrieval and reasoning ability across the methods. Beyond benchmark construction, we provide a systematic analysis linking reasoning difficulty to graph structure, showing how structural variations induce predictable changes in model accuracy. Together, these results demonstrate that SeedRG enables more discriminative and robust evaluation of RAG systems.

1 Introduction

Retrieval-Augmented Generation (RAG) Lewis et al. (2020) is widely used to improve large language models (LLMs) by incorporating external knowledge at inference time. Because RAG avoids retraining by leveraging retrieval, it has become a standard approach for

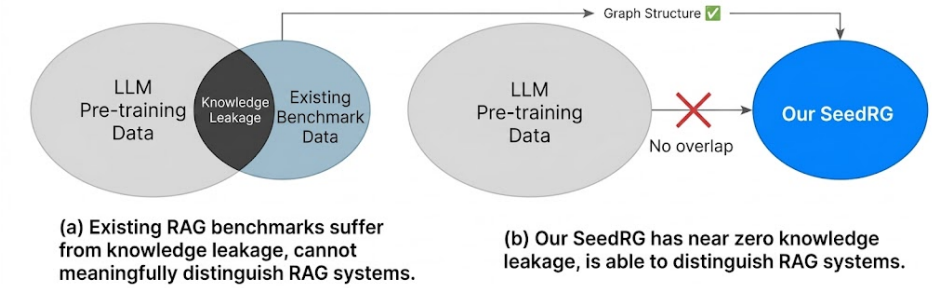


Figure 1: An example of the RAG evaluation gap. **(a)** Existing benchmarks overlap with LLM pretraining data, making retrieval redundant. **(b)** SeedRG preserves the reasoning structure of seed questions while replacing all entities with novel counterparts, ensuring no overlap with parametric knowledge.

knowledge-intensive tasks. Progress on RAG systems is typically measured using multi-hop question answering benchmarks, which are intended to require retrieval over multiple documents.

However, these benchmarks often fail to test retrieval at all. We find that on widely used multi-hop QA datasets , LLMs can answer even more than half of questions correctly *without retrieving any documents*. When the majority of questions are solvable from LLM parametric memory alone, evaluation results are dominated by the model's internal knowledge rather than the quality of retrieval. As a result, current benchmarks cannot meaningfully distinguish RAG systems—even weak retrievers inherit strong performance from the underlying model.

This issue undermines a large body of empirical work. RAG systems are now central to applications such as open-domain question answering Petroni et al. (2021), fact verification, and domain-specific reasoning, with a growing ecosystem of retrieval methods—including dense, graph-based, and ontology-guided approaches—competing on shared benchmarks. When benchmarks can be solved from parametric memory, performance is dominated by the underlying model rather than the retriever, collapsing the observable differences between systems. This masks variation in retrieval quality and limits our ability to measure how different retrieval strategies influence downstream reasoning.

We identify two mechanisms that drive this evaluation failure. **Knowledge leakage** Agarwal et al. (2024); Wu et al. (2025); Desai et al. (2026); Yoon et al. (2025) occurs when benchmark questions are answerable from an LLM's parametric memory, making retrieval unnecessary. **Benchmark aging** Zhou et al. (2023); Zhang et al. (2026) compounds this effect over time. As benchmarks are reused in training and data curation pipelines, their contents are absorbed into model parameters, progressively reducing their ability to test retrieval. Together, these mechanisms collapse the evaluation signal, obscuring differences between systems.

Addressing this problem requires benchmarks that are (1) outside the model's parametric knowledge and (2) regenerable to remain robust to benchmark aging. A natural approach is to generate new data using LLMs. However, naive generation fails to meet these requirements: generated questions often reuse well-known entities (leading to continued leakage), may introduce factual errors in supporting context, and provide no control over reasoning difficulty.

We introduce **SeedRG**, a semi-synthetic benchmark generation pipeline that addresses these challenges. Starting from a seed benchmark, SeedRG extracts a *reasoning graph* from each question–context pair to capture its underlying structure, and then generates new examples via *type-constrained entity replacement*. This produces structurally equivalent but novel instances that are unlikely to exist in the model's parametric knowledge, while preserving the original reasoning patterns.

To ensure quality, SeedRG incorporates two verification steps. A *reasoning graph consistency check* ensures that the transformed examples preserve the original reasoning structure and difficulty. A *knowledge leakage check* filters out instances that can be answered without retrieval. Together, these steps ensure that generated examples are both retrieval-dependent and difficulty-preserving. We further introduce two metrics—*leakage error* and *answerability accuracy*—to quantify the effectiveness of RAG benchmarks.

In summary, our contributions are as follows:

1. We provide systematic evidence that three widely used multi-hop QA benchmarks suffer from **knowledge leakage**, and formalize benchmark quality in terms of leakage error and answerability accuracy.
2. We propose **SeedRG**, a semi-synthetic pipeline that generates leakage-free benchmarks by combining reasoning graph extraction with type-constrained entity replacement, along with dual verification to preserve difficulty and enforce retrieval dependence.
3. We show that SeedRG produces benchmarks that reduce knowledge leakage, preserve reasoning difficulty, and reveal meaningful performance differences across RAG systems that are obscured in existing benchmarks. Compared to direct LLM generation, SeedRG yields higher-quality benchmarks with substantially lower leakage and fewer factual inconsistencies.

2 Background

2.1 Synthetic datasets

Synthetic Benchmarks for RAG Recent frameworks have standardized the evaluation of RAG systems through automated data synthesis. RAGEval Zhu et al. (2024) generates schema-driven datasets to assess *scenario-specific* factual accuracy. It defines dataset quality using three key metrics: *completeness* of the answer, absence of *hallucination*, and *irrelevance* of non-essential content. To scale this approach, BenchmarkQED Research (2025) employs the AutoQ tool to generate synthetic queries across a principled 2×2 taxonomy. It measures quality via *coverage* (diversity of query types) and *rigor* (stability of comparative rankings). However, both frameworks share a critical limitation: they fail to add restrictions that avoid generating queries already present in the LLM's knowledge. By not explicitly disentangling parametric memory from retrieval necessity, these benchmarks struggle to isolate the true utility of the retrieval component.

Synthetic Data for Instruction Tuning In the broader context of model alignment, synthetic data quality is often defined by downstream efficiency rather than retrieval isolation. Distilling Step-by-Step Hsieh et al. (2023) and Orca Mukherjee et al. (2023) demonstrate that "good" synthetic data allows smaller student models to achieve *teacher-parity* with significantly fewer training samples. Similarly, MetaMath Yu et al. (2023) and SynPO Dong et al. (2024) validate dataset quality through *reasoning transfer* to unseen math tasks and *iterative win-rate improvements* on public leaderboards. While these methods successfully enhance general reasoning and alignment, they do not address the specific knowledge-boundary constraints required to rigorously prevent memory-based hallucinations in RAG tasks.

We observe no significant improvement in accuracy across several benchmarks. In some cases, performance even degrades, despite the retrieval module returning correct supporting paragraphs. This suggests that the language model often already possesses sufficient parametric knowledge to answer the questions without relying on retrieved evidence, limiting the benefit of external retrieval.

2.2 Knowledge Leakage in LLM

Knowledge leakage has been widely discussed in previous research. Recent work has doubted the efficacy of LLM in recommendation Zhang et al. (2026); Zhou et al. (2023),

query expansion Yoon et al. (2025), privacy Agarwal et al. (2024); Wu et al. (2025); Desai et al. (2026), and many other tasks Baser et al. (2025).

To prevent the knowledge leakage, researchers applied different stratigies. Agarwal et al. (2024); Desai et al. (2026) use defense instructions and filtering guardrails within the prompt or system layer to block leakage. To prevent the knowledge leakage, researchers applied different stratigies. Agarwal et al. (2024); Desai et al. (2026) mitigate the problem via prompt to block leakage. Wu et al. (2025) and Baser et al. (2025) propose secure KV-cache management and knowledge graph monitoring as extra system design to track and solve leakage issues. Zhou et al. (2023); Zhang et al. (2026) added constraints in fine-tuning stage to prevent knowledge leakage.

3 Formalizing Valid RAG Benchmarks

Before proposing a solution, we formalize what it means for a RAG benchmark to be *valid*. Specifically, it should measure retrieval quality rather than parametric knowledge recall. We identify two necessary conditions for validity and show how existing benchmarks violate them.

A valid RAG benchmark must separate what the model already knows from what is provided through retrieval. When this separation fails, evaluation signal collapses and performance is dominated by parametric knowledge, making it difficult to attribute gains to retrieval. We formalize this failure as **knowledge leakage**, and argue that it is exacerbated over time due to **benchmark aging**.

3.1 Knowledge Leakage

Let M denote a language model and $\mathcal{Q}$ a benchmark dataset. For each question $q \in \mathcal{Q}$, we measure accuracy under two conditions: $\text{Acc}_{\text{no\_ctx}}$ (when the model answers are produced without any retrieved context) and Acc_{gold} (when the model answers are produced given only the ground-truth supporting documents).

Knowledge leakage occurs when $\text{Acc}_{\text{no\_ctx}}$ is high, indicating that questions can be answered directly from parametric memory. In this regime, retrieval has provided no additional information, and benchmark performance no longer reflects retrieval quality. As we show in Section 5.2, $\text{Acc}_{\text{no\_ctx}}$ reaches 52% on HotpotQA Yang et al. (2018), 62% on 2WikiMultihopQA Welbl et al. (2018), and 75% on QASC Khot et al. (2020).

We formalize two criteria for a valid RAG benchmark:

1. **Leakage Error.** The extent to which questions are answerable from parametric knowledge alone:
$$\text{Acc}_{\text{no\_ctx}}(\mathcal{Q})$$
A valid benchmark should have low leakage error.
2. **Answerability Accuracy.** The improvement in accuracy when the correct context is provided:
$$\text{Acc}_{\text{gold}}(\mathcal{Q}) - \text{Acc}_{\text{no\_ctx}}(\mathcal{Q})$$
A valid benchmark should exhibit high answerability accuracy.

Together, these criteria ensure that benchmark performance reflects retrieval-dependent reasoning rather than memorization.

3.2 Benchmark Aging

Even when a benchmark initially satisfies these criteria, it degrades over time Zhou et al. (2023); Zhang et al. (2026). As pretraining corpora expand, benchmark questions are increasingly incorporated into model training data. This progressively increases $\text{Acc}_{\text{no\_ctx}}$, increasing leakage error and weakening the benchmark's ability to test retrieval.

For static benchmarks, this process is irreversible—there is no mechanism to prevent their absorption into parametric knowledge. The result is a gradual inflation of reported performance that reflects improved memorization rather than improved retrieval. As a consequence, benchmarks lose their discriminative power to assess RAG systems. A valid RAG benchmark must therefore not only minimize knowledge leakage, but also remain robust to benchmark aging.

4 Methodology

Rather than introducing a static benchmark, we propose **SeedRG**, a framework for **continually generating** benchmarks that minimize **knowledge leakage** while preserving **answerability accuracy**, thereby mitigating **benchmark aging**.

SeedRG takes a multi-hop RAG benchmark as a seed and transforms each question–context pair into a new instance that is retrieval-dependent and difficulty-preserving. The key idea is to preserve the underlying reasoning structure while replacing entities with ones that fall outside the model's parametric knowledge.

We formalize three requirements for valid benchmark generation:

1. **Leakage and Answerability.** Generated questions must not be answerable from parametric knowledge alone, while remaining answerable given the correct context. This requires both a generation mechanism that produces novel content and a verification step that filters out leaking instances.
2. **Renewable Generation.** The framework must produce fresh benchmark instances on demand, so that as models absorb existing data, new instances can replace them. This ensures robustness to benchmark aging.
3. **Difficulty Preservation.** The generation process must preserve reasoning difficulty. Without structural constraints, generated questions may become trivially easy or arbitrarily hard, confounding retrieval performance with the model's reasoning ability.

The overall framework is shown in Figure 2. SeedRG consists of two main components: **type-constrained entity replacement** (Section 4.1), which generates **retrieval-dependent** samples, and **reasoning graph extraction** (Section 4.2), which ensures **difficulty preservation**.

4.1 Type-Constrained Entity Replacement

SeedRG replaces entities in both the question and context while ensuring the resulting instance is non-leaking and answerable.

Given a seed question q with context c and answer a, we first extract its reasoning graph to identify entity nodes. For each entity e_i, we prompt an LLM to generate a replacement of the same semantic type but unlikely to be present in parametric knowledge (e.g., a composer replaced by another composer, a city by another city). This produces a type-constrained substitute e_i'.

We define a mapping $\mathcal{M} : \{e_i\} \to \{e_i'\}$ and apply it jointly to the question, answer, and context: $q' = \mathcal{M}(q)$, $a' = \mathcal{M}(a)$, and $c' = \mathcal{M}(c)$.

For the context, we perform entity replacement in two ways. First, we directly substitute entity mentions in the original text, preserving surface form and information density. Second, we extract knowledge graph triplets $\mathcal{T} = \{(s_i, r_i, o_i)\}$ from c, apply $\mathcal{M}$ to all entities, and regenerate a natural-language passage from the transformed triplets. This second path enables controlled perturbations of graph structure (Section 5.3.2).

A generated sample is only valid if the question cannot be answered without context. For each candidate (q', a'), we query the LLM with q' alone (no context) multiple times ($N = 3$). If any response contains the correct answer a', the sample is rejected. The pipeline then

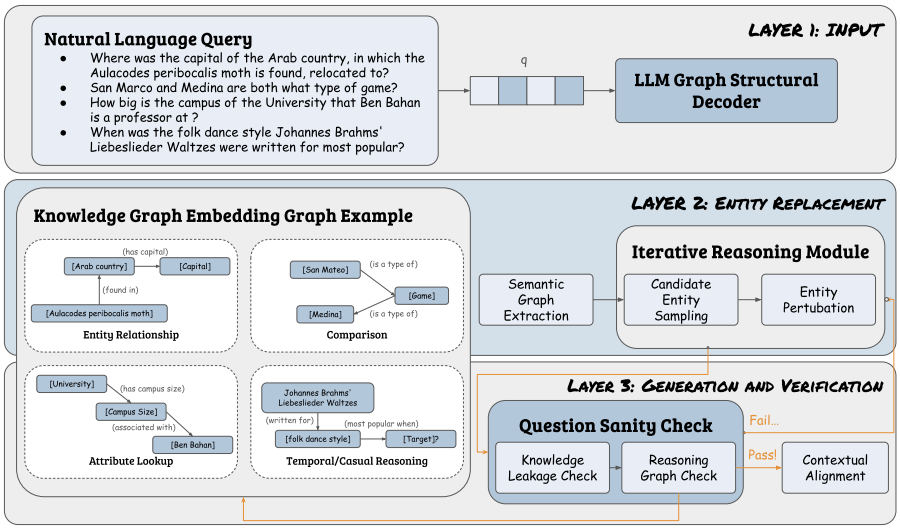


Figure 2: Overview of the SeedRG benchmark generation pipeline. Given a seed question–context pair, the pipeline extracts its reasoning graph, applies type-constrained entity replacement, and validates the result through reasoning graph and knowledge leakage checks. Samples that fail either check are rejected and regenerated.

resamples entity replacements with additional constraints to avoid previously tried entities. This process repeats until a non-leaking instance is obtained.

Because each seed question admits many valid replacements, this process supports renewable generation—new benchmark instances can be produced on demand, directly addressing benchmark aging.

4.2 Reasoning Graph Extraction

The difficulty of a multi-hop question is determined by its reasoning structure—the number of hops, the dependency chain, and the connectivity between entities. If the structure of the reasoning graph changes, it might result in different reasoning difficulties.

To preserve reasoning difficulty, we extract reasoning graphs before and after transformation. For a seed question q with context c, we construct a question graph $G_q = (V_q, E_q)$ that captures the reasoning chain, and a context graph $G_c = (V_c, E_c)$ derived from factual triplets $\mathcal{T} = \{(s_i, r_i, o_i)\}$. After transformation, we extract $G_{q'}$ and $G_{c'}$ and verify structural equivalence: $G_{q'} \cong G_q$ and $G_{c'} \cong G_c$. If the structure is not preserved, the sample is discarded and regenerated.

We also prove our hypothesis in Section 5.3.2, that modifying the structure directly changes task difficulty. Hence, the reasoning graph verification step is necessary to maintain the reasoning difficulty.

5 Experiments

5.1 Experimental Setup

All generated benchmarks are produced using GPT-4o-mini following the SeedRG pipeline described in Section 4. We evaluate benchmark quality using three frontier LLMs with strong reasoning capabilities: GPT-5, Claude Sonnet 4.5, and Gemini 2.5 Flash.[1]

[1]Our code is available at https://anonymous.4open.science/r/SeedRG-02D6.

Table 1: Benchmark comparison. **Leakage Error** ($\text{Acc}_{\text{no\_ctx}}$, lower is better): fraction of questions answerable without context. **Answerability Accuracy** ($\text{Acc}_{\text{gold}} - \text{Acc}_{\text{no\_ctx}}$, higher is better): potential gain from retrieval.

		GPT-5			Claude Sonnet 4.5			Gemini 2.5 Flash		
Metric	**Dataset**	SeedRG	DG	Orig	SeedRG	DG	Orig	SeedRG	DG	Orig
Leakage Error	HotpotQA	**.014**	.190	.500	**.135**	.310	.459	**.041**	.180	.311
	WikiHop	**.114**	.390	.629	**.129**	.420	.386	**.057**	.350	.400
	QASC	**.140**	.320	.750	**.170**	.530	.820	**.160**	.380	.770
Answerability Accuracy	HotpotQA	.418	**.500**	.338	**.595**	.470	.473	.473	.460	**.500**
	WikiHop	**.457**	.330	.200	**.657**	.460	.543	**.514**	.380	.414
	QASC	**.710**	.410	.180	**.690**	.290	.170	**.760**	.380	.210

Our experiments address two questions. First, we evaluate whether SeedRG produces higher-quality benchmarks compared to Direct Generation (DG) and the original benchmarks, in terms of knowledge leakage and answerability accuracy. Second, we use these benchmarks to evaluate RAG systems and examine whether SeedRG reveals differences in retrieval performance that are obscured in existing benchmarks.

We compare SeedRG against **Direct Generation (DG)**, where an LLM generates new question–context–answer triples from scratch given a seed example. The DG prompt explicitly instructs the model to (1) preserve reasoning type and difficulty, (2) use entirely different entities and facts, and (3) require the provided context for answering (i.e., not answerable from parametric knowledge alone).

To evaluate both benchmark quality and retrieval dependence, we measure performance under the following conditions/methods:

- **No-Context**: The LLM answers the question without any retrieved context. This measures knowledge leakage.
- **Gold**: The ground-truth supporting documents are provided as context, representing an upper bound on answerability.
- **HippoRAG** Gutiérrez et al. (2024) is a RAG framework that mimics human memory by constructing a knowledge graph as a long-term memory index.
- **OGRAG** Sharma et al. (2025) is an ontology-grounded RAG approach that organizes documents using expert-defined ontologies and hypergraph structures.
- **GraphRAG** Edge et al. (2024) constructs a global knowledge graph over the corpus and performs retrieval over graph-structured representations.
- **SemanticRAG** is a standard embedding-based retrieval approach that ranks documents using cosine similarity between query and document embeddings, representing a common production baseline.

5.2 Benchmark Comparison

We compare three benchmarks: SeedRG, Direct Generation (DG), and the original benchmark across two metrics defined in Section 3: leakage error, measured by $\text{Acc}_{\text{no\_ctx}}(\mathcal{Q})$ and answerability accuracy, measured by $\text{Acc}_{\text{gold}}(\mathcal{Q}) - \text{Acc}_{\text{no\_ctx}}(\mathcal{Q})$ (higher is better). Results are summarized in Table 1.

Leakage Error. On the original benchmarks, all three LLMs achieve high no-context accuracy (31–78%), confirming substantial knowledge leakage. In contrast, SeedRG reduces no-context accuracy dramatically—to 1.4% (GPT-5) on HotpotQA, 4.1% (Gemini), and 13.5% (Claude)—effectively eliminating leakage across models. Direct Generation also reduces leakage relative to the original benchmarks, but only partially (18–53%), indicating that naive generation is insufficient to remove parametric shortcuts.

Answerability Accuracy. The gap $\text{Acc}_{\text{gold}} - \text{Acc}_{\text{no\_ctx}}$ quantifies the potential contribution of retrieval, ie. how much accuracy could improve if retrieval were perfect. SeedRG consistently yields the largest gaps (42–76%), indicating that these benchmarks create substantial headroom for retrieval to matter. In contrast, the original benchmarks exhibit smaller gaps (18–54%) due to high no-context accuracy, while Direct Generation again lies in between. Taken together, these results show that SeedRG restores evaluation signal by reducing leakage and increasing the extent to which performance can depend on retrieval.

5.3 Generation Quality Analysis

5.3.1 Generation Quality for Question and Context

We evaluate generation quality along three dimensions: knowledge leakage, factual consistency, and reasoning difficulty. Results are shown in Figure 3.

Knowledge leakage and factual consistency. For each generated question, we query the LLM without context. If the model answers correctly, the question exhibits knowledge leakage. If the model produces a confident but incorrect answer that contradicts the generated context, the context is deemed factually inconsistent. Otherwise, the question is considered non-leaking. Figure 3(a) shows that Direct Generation (DG) produces low-quality benchmarks: only 0–11% of questions are non-leaking, while 19–39% leak and 53–70% exhibit factual inconsistencies. This occurs because DG generates contexts about entities that remain within the LLM's knowledge distribution, allowing parametric knowledge to override the generated evidence. In contrast, Figure 3(b) shows that SeedRG achieves 1–14% leakage and 0% factual inconsistencies. Additional results are provided in Appendix Table 2.

Graph structure preservation. Figure 3(c) compares graph statistics (number of nodes, edges, density, and average degree) against the original benchmark. SeedRG deviates by less than 5% across all metrics, while DG deviates by up to 27% in nodes and 38% in edges. These structural shifts indicate that DG alters the underlying reasoning structure, leading to uncontrolled changes in difficulty.

5.3.2 From Graph Structure to Reasoning Difficulty

To validate the connection between graph structure and reasoning difficulty, we regenerate SeedRG contexts from graph triplets under two settings: (1) preserving the original structure, and (2) applying cyclic permutations to rewire edges while keeping the same entities.

As shown in Figure 3(d), the structure-preserving condition closely matches SeedRG across all models and datasets, while the structure-perturbed condition exhibits consistent performance degradation. This demonstrates that reasoning difficulty is governed by graph structure, and validates the necessity of the reasoning graph check in the SeedRG pipeline.

5.4 RAG Algorithms Performance Comparison on SeedRG

Having established that SeedRG minimizes knowledge leakage, preserves answerability accuracy, and maintains reasoning difficulty, we now use it to evaluate RAG systems. Figure 4a reports accuracy across all retrieval conditions on SeedRG, using three QA engines. Full per-engine results are provided in Appendix Tables 3–5.

On the original benchmarks, RAG systems show limited differentiation—on HotpotQA, all systems cluster within 3% accuracy, making it impossible to distinguish retrieval quality. In contrast, SeedRG widens the spread to 10–18% on HotpotQA and WikiHop, revealing genuine differences in retrieval effectiveness. For example, HippoRAG consistently outperforms OGRAG and GraphRAG across all engines, a ranking that is not observable on the original benchmarks. QASC exhibits a narrower range (.81–.93) due to its multiple-choice format, but still shows consistent differences across systems.

We also analyze the stability of benchmarks in Figure 4b. We take WikiHop benchmarks for example, and regenerate SeedRG 4 times. We still do the generation with GPT-4o-mini

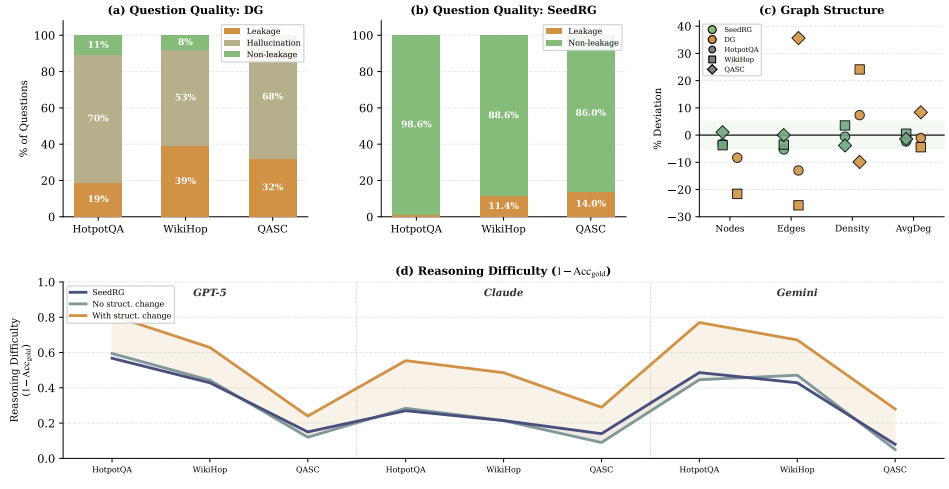
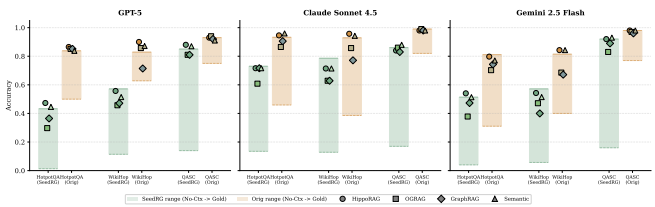
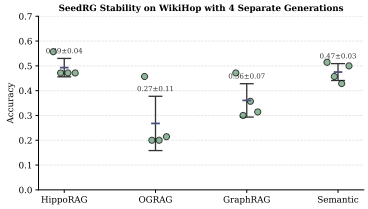


Figure 3: Generation quality comparison: SeedRG vs Direct Generation. SeedRG produces non-leaking, hallucination-free questions while preserving graph structure, which leads to same reasoning difficulty.



(a) RAG algorithm evaluation: SeedRG vs Original benchmarks. SeedRG provides meaningful differentiation, while Original benchmarks cluster due to knowledge leakage.



(b) Benchmark stability: 4 independent SeedRG generations on WikiHop. SeedRG shows a stable winner/loser.

Figure 4: RAG algorithm evaluation and stability on SeedRG.

and evaluate each with GPT-5. As shown in Figure 4b, the standard deviation across runs is stable for all RAG systems.

6 Conclusion and Future Work

We identify knowledge leakage as a fundamental failure mode in current multi-hop QA benchmarks. LLMs can answer 31–78% of questions without retrieval, collapsing evaluation signal and obscuring differences between RAG systems. This issue compounds over time through benchmark aging, as benchmarks are absorbed into model training.

We propose **SeedRG**, a semi-synthetic pipeline that generates leakage-free, difficulty-preserving benchmarks from existing datasets. By preserving reasoning structure while replacing entities outside the model's parametric knowledge, SeedRG enforces retrieval dependence and enables renewable benchmark generation. Empirically, SeedRG reduces leakage and restores discriminative power, revealing performance differences across RAG systems that are invisible on existing benchmarks.

Finally, we show that reasoning difficulty is governed by graph structure, providing a principled basis for controlling task difficulty. An important direction for future work is to move from preservation to control—enabling generation of benchmarks at targeted difficulty levels.

References

Divyansh Agarwal, Alexander Richard Fabbri, Ben Risher, Philippe Laban, Shafiq Joty, and Chien-Sheng Wu. Prompt leakage effect and mitigation strategies for multi-turn llm applications. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1255–1275, 2024.

Manit Baser, Dinil Mon Divakaran, and Mohan Gurusamy. Thinkeval: Practical evaluation of knowledge leakage in llm editing using thought-based knowledge graphs. *arXiv preprint arXiv:2506.01386*, 2025.

Pratyush Desai, Luoxi Tang, Yuqiao Meng, and Zhaohan Xi. Safegpt: Preventing data leakage and unethical outputs in enterprise llm use. *arXiv preprint arXiv:2601.06366*, 2026.

Qingxiu Dong, Li Dong, Xingxing Zhang, Zhifang Sui, and Furu Wei. Self-boosting large language models with synthetic preference data. *arXiv preprint arXiv:2410.06961*, 2024.

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.

Bernal J Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in neural information processing systems*, 37:59532–59569, 2024.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *arXiv preprint arXiv:2305.02301*, 2023.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8082–8090, 2020.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474, 2020.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*, 2023.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, et al. Kilt: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2523–2544, 2021.

Microsoft Research. Benchmarkqed: Automated benchmarking for retrieval-augmented generation. https://www.microsoft.com/en-us/research/blog/benchmarkqed-automated-benchmarking-of-rag-systems/, 2025.

Kartik Sharma, Peeyush Kumar, and Yunqing Li. Og-rag: ontology-grounded retrieval-augmented generation for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 32950–32969, 2025.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302, 2018.

Guanlong Wu, Zheng Zhang, Yao Zhang, Weili Wang, Jianyu Niu, Ye Wu, and Yinqian Zhang. I know what you asked: Prompt leakage via kv-cache sharing in multi-tenant llm serving. In *NDSS*, 2025.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William 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*. Association for Computational Linguistics, 2018.

Yejun Yoon, Jaeyoon Jung, Seunghyun Yoon, and Kunwoo Park. Hypothetical documents or knowledge leakage? rethinking llm-based query expansion. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19170–19187, 2025.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

Mingqiao Zhang, Qiyao Peng, Yumeng Wang, Chunyuan Liu, and Hongtao Liu. Benchmark leakage trap: Can we trust llm-based recommendation? *arXiv preprint arXiv:2602.13626*, 2026.

Kun Zhou, Yutao Zhu, Zhipeng Chen, Wentong Chen, Wayne Xin Zhao, Xu Chen, Yankai Lin, Ji-Rong Wen, and Jiawei Han. Don't make your llm an evaluation benchmark cheater. *arXiv preprint arXiv:2311.01964*, 2023.

Kunlun Zhu, Yifan Luo, Dingling Xu, Yukun Yan, Zhenghao Liu, Shi Yu, Ruobing Wang, Shuo Wang, Yishan Li, Nan Zhang, et al. Rageval: Scenario specific rag evaluation dataset generation framework. *arXiv preprint arXiv:2408.01262*, 2024.

A Additional Results

A.1 Generation Quality Examples

Table 2 shows concrete examples of the two failure modes discussed in Section 5.3. In Example 1 (knowledge leakage), given the same seed question about a band's nationality, SeedRG replaces the band with a novel entity ("Mellow Vibes Harmony") that the LLM cannot recognize, while DG generates a question about the well-known film "Pan's Labyrinth"—the LLM immediately answers "Mexican" without needing any context. In Example 2 (factual hallucination), DG fabricates a band called "The Echoes" with lead guitarist "Alex Chen," but the LLM's parametric knowledge associates "The Echoes" with a real band whose guitarist is "Vic Briggs"—the generated context directly contradicts what the LLM knows. SeedRG avoids this by reusing original factual structures with only entity names changed.

A.2 Full Results: Entity Replacement vs. Direct Generation

Tables 3–5 present the complete accuracy results across all RAG conditions, datasets, and generation methods. "SeedRG" denotes our pipeline; "DG" denotes Direct Generation; "Orig" denotes the original (unmodified) benchmark. SeedRG and Orig use N=74/70/100 questions for HotpotQA/WikiHop/QASC respectively; DG uses the same N with newly generated questions.

Table 2: Generation quality examples. Example 1 (WikiHop seed): DG produces a question about a well-known entity, causing knowledge leakage. Example 2 (HotpotQA seed): DG fabricates facts that contradict LLM knowledge. Red highlights critical differences.

	Question	**No-Context Answer**	**Ground Truth**
Example 1: Knowledge Leakage (WikiHop seed)			
WikiHop	Lead singer of Hurlingham Reggae Band's nationality?	Italian–Scottish	Italian–Scottish
SeedRG	Lead singer of Mellow Vibes Harmony's nationality?	Unknown.	Galician–Welsh
DG	Director of Pan's Labyrinth's nationality?	Mexican.	Mexican
Example 2: Factual Hallucination (HotpotQA seed)			
HotpotQA	Lowest vocal range in Cosmos?	Jānis Strazdi	Jānis Strazdiņš
SeedRG	Lowest vocal range in Aetherius?	Unclear.	Raimonds Bērziņš
DG	Lead guitarist in The Echoes?	Vic Briggs.	Alex Chen

Table 3: Full results: GPT-5. Bold indicates the lowest no-context accuracy (least knowledge leakage).

	HotpotQA			**WikiHop**			**QASC**		
RAG	SeedRG	DG	Orig	SeedRG	DG	Orig	SeedRG	DG	Orig
No-Ctx	**.014**	.190	.500	**.114**	.390	.629	**.140**	.320	.750
Gold	.432	.690	.838	.571	.720	.829	.850	.730	.930
HippoRAG	.473	.660	.865	.557	.690	.900	.880	.710	.930
OGRAG	.297	.260	.851	.457	.450	.857	.810	.390	.940
GraphRAG	.365	.240	.851	.471	.440	.714	.810	.460	.920
Semantic	.446	.670	.838	.514	.700	.871	.870	.720	.910

Table 4: Full results: Claude Sonnet 4.5.

	HotpotQA			**WikiHop**			**QASC**		
RAG	SeedRG	DG	Orig	SeedRG	DG	Orig	SeedRG	DG	Orig
No-Ctx	**.135**	.310	.459	**.129**	.420	.386	**.170**	.530	.820
Gold	.730	.780	.932	.786	.880	.929	.860	.820	.990
HippoRAG	.716	.790	.946	.714	.820	.957	.840	.860	.980
OGRAG	.608	.330	.865	.629	.430	.857	.860	.680	.990
GraphRAG	.716	.330	.905	.629	.470	.771	.830	.650	.980
Semantic	.716	.810	.959	.714	.830	.943	.880	.850	.980

Table 5: Full results: Gemini 2.5 Flash.

	HotpotQA			**WikiHop**			**QASC**		
RAG	SeedRG	DG	Orig	SeedRG	DG	Orig	SeedRG	DG	Orig
No-Ctx	**.041**	.180	.311	**.057**	.350	.400	**.160**	.380	.770
Gold	.514	.640	.811	.571	.730	.814	.920	.760	.980
HippoRAG	.541	.610	.797	.543	.730	.843	.920	.770	.980
OGRAG	.378	.070	.703	.471	.230	.686	.830	.470	.970
GraphRAG	.473	.190	.743	.400	.350	.671	.890	.450	.960
Semantic	.514	.620	.770	.514	.710	.843	.930	.790	.980