Evaluating Multi-Hop Reasoning in Large Language Models: A Chemistry-Centric Benchmark

Mohammad Khodadad[1,2]*, Ali Shiraee Kasmaee[1,2]*, Mahdi Astaraki[1,2], Nicholas Sherck[3], Hamidreza Mahyar[1], Soheila Samiee[2]

[1]McMaster University, Canada, [2]BASF Canada Inc., Canada, [3]BASF Corporation, USA

Correspondence: khodam3@mcmaster.ca

Abstract

We introduce ChemComp, a benchmark for evaluating compositional multi-hop reasoning in large language models (LLMs) focused on the chemistry domain. To support industrial applications, we develop an automated pipeline that can construct benchmarks from proprietary data sources. The pipeline integrates generative reasoning models, chemical named-entity recognition, and external knowledge bases to build knowledge graphs. Crucially, ChemComp targets domain-specific multi-hop compositional reasoning capabilities rather than factual recall, emphasizing reasoning over memorization in chemical tasks. In this study, we apply this pipeline to recent chemistry literature. Multi-hop questions are generated and refined with domain-expert feedback, resulting in a high-quality dataset and robust evaluation protocols. Using ChemComp, we conduct systematic experiments comparing LLM performance with and without retrieval augmentation, including the idealized scenario of gold-context provision. Our findings reveal that even state-of-the-art models struggle with compositional reasoning: retrieval significantly boosts accuracy, yet reasoning errors persist even under perfect retrieval conditions. These results underscore the limitations of current LLMs and the critical role of retrieval-augmented methods in scientific reasoning. Furthermore, our pipeline is generalizable through domain-specific fine-tuning of the pipeline, enabling the creation of challenging benchmarks across domains and proprietary datasets, advancing the study of multi-hop reasoning in LLMs.

1 Introduction

Large Language Models (LLMs) have achieved impressive performance on a wide range of tasks, yet their ability to perform complex, multi-step reasoning remains an ongoing challenge. Techniques such as chain-of-thought (CoT) prompting (Wei et al., 2022; Wang et al., 2022; Yao et al., 2023; Besta et al., 2024; Xiang et al., 2025) and structural innovations (Lewis et al., 2020; d'Avila Garcez and Lamb, 2020; Santoro et al., 2017) have enabled notable improvements in reasoning, particularly in mathematics and coding. OpenAI's o-series (OpenAI, 2024a,b) was among the first to introduce inference-time scaling of CoT reasoning depth, and subsequent open-source models such as DeepSeek R1 (Zelikman et al., 2022; Guo et al., 2025) and Qwen QwQ (Patil, 2025) have adopted similar strategies. Notably, these models also leverage reinforcement learning during training to further refine their CoT reasoning, consistently improving performance with increased test-time compute.

For the integration of these models in enterprise AI, a thorough evaluation of the models, reflecting real-world use cases, is required. However, evaluating reasoning capabilities remains challenging, especially in domain-specific settings. The community relies on multi-hop reasoning benchmarks spanning mathematics (Cobbe et al., 2021; Hendrycks et al., 2021), programming (Chen et al., 2021; Austin et al., 2021; Jimenez et al., 2023), and general QA tasks such as HotpotQA (Yang et al., 2018) and StrategyQA (Geva et al., 2021). While reasoning-focused scaling methods have improved performance on these benchmarks, they are largely general-purpose and may not capture the nuanced reasoning required in specialized domains. A key open question is whether reasoning strategies learned from general data transfer effectively to these domains.

In scientific domains such as chemistry, multi-hop reasoning is essential for integrating interconnected, domain-specific knowledge. Although several multi-hop question answering benchmarks exist, evaluations tailored to chemical

*These authors contributed equally.

Findings of the Association for Computational Linguistics: EACL 2026, pages 6117–6143

reasoning remain limited (Wellawatte et al.; Huang et al., 2024; Rein et al., 2024). Recent efforts, including datasets targeting subfields such as reticular chemistry Zheng et al. (2025), further highlight the need for more comprehensive and challenging chemistry-focused benchmarks.

While complete prevention of data leakage is ultimately unattainable, especially for widely deployed LLMs, its effects can be substantially mitigated through careful benchmark design. Leveraging more recent or curated sources, along with automatically generated tasks that require multi-hop compositional reasoning, reduces the probability that benchmark instances overlap with pretraining data. Rather than claiming strict isolation from pretraining corpora, our objective is to create evaluation settings in which correct answers are unlikely to be retrieved through memorization alone, thereby providing a more faithful measure of chemical reasoning capability.

While general-domain knowledge graph question answering (KGQA) has seen significant progress, chemistry-specific KGQA remains relatively underexplored. Existing chemistry benchmarks predominantly emphasize factual recall or single-hop inference, leaving a gap for domain-grounded KGQA that explicitly evaluates structured, multi-step reasoning. To address this gap, future benchmarks should rely on carefully curated or automatically generated data that is unlikely to be memorized by general-purpose LLMs.

Furthermore, real-world industrial and scientific use cases often require long-range, multi-step reasoning accompanied by multi-source data integration (Gao et al., 2025); therefore, a reliable benchmark should extend beyond evaluating the domain knowledge of the models, mimicking the real-world scenario of answering complex questions that necessitate long-range reasoning and multi-source integration using proprietary data.

To address this gap, we propose an automated multi-hop reasoning QA generation pipeline that leverages generative AI models (OpenAI's `o3-mini` and `gpt-4o`), domain-specific data, and feedback from domain experts. Our pipeline systematically extracts and verifies chemical entities through named entity recognition (NER) from recent domain-specific literature, linking them to external databases to construct a domain-specific knowledge graph. It also systematically generates challenging multi-hop question-answer pairs. Three rounds of expert feedback and iterative pipeline refinement were implemented to enhance the quality of these complex, multi-hop question generations. Verification by chemists on a random sample of 120 automatically generated questions revealed an increase in the acceptance rate of questions from 5% to 90%. Our solution, ChemComp, is designed to measure domain-specific multi-hop compositional reasoning capabilities rather than factual recall, so every component of the dataset aims to test reasoning capabilities in chemistry beyond memorized facts.

In the next step, we evaluate 13 language models, covering reasoning fine-tuned and non-reasoning models, as well as open-source and proprietary systems, in these tasks. We assess both their question understanding and reliance on internal knowledge, and measure how effectively they use gold context to answer these challenging multi-hop questions. Our contributions are as follows:

1. We provide extensive experimental evidence that multi-hop compositional reasoning in scientific domains remains a significant limitation for current state-of-the-art LLMs.
2. We demonstrate that even retrieval augmentation with perfect context does not guarantee flawless reasoning, highlighting the intrinsic difficulty of compositional reasoning.
3. We introduce an automated, scalable pipeline that constructs knowledge graphs and generates domain-specific multi-hop QA datasets using NER and generative models.
4. We release a challenging Chemistry benchmark, curated and validated by domain experts, to facilitate future research on scientific reasoning.

2 Background and related works

Multi-hop Question Multi-hop question answering (QA) has evolved as a key method to evaluate the multi-step reasoning abilities of large language models (Mavi et al., 2024). Answering multi-hop questions requires integrating multiple pieces of evidence. Traditionally, datasets such as HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), and MedHop (Welbl et al., 2018) were manually or semi-automatically curated through crowd-sourcing and knowledge-base

relations. While these approaches yield high-quality, human-validated questions, they are resource-intensive. More recently, LLMs are leveraged to automatically generate multi-hop datasets. In many cases, single-hop QA pairs are first generated and later merged using entity linking techniques, a common approach that connects individual entities across questions. For example, the MuSiQue framework (Trivedi et al., 2022) fuses two QA pairs by linking a named entity from the first answer to the subsequent question, thereby forming a chain of reasoning. Other methods, such as MultiHop-RAG (Tang and Yang, 2024), extend this paradigm by incorporating retrieval-augmented generation (RAG) to paraphrase factual sentences and group them based on shared topics, reflecting the diverse strategies that are emerging in multi-hop QA generation. Recent work has advanced multi-hop question generation by improving controllability and incorporating constraints. The Dual-Perspective Keyword Guidance (DPKG) model uses question- and document-oriented keywords to guide transformer-based generation, enhancing coherence and reasoning depth (Li et al., 2025). (Zhao and Li, 2025) introduce RUBY, which applies semantic-constraint dimensionality reduction and dynamic projection to produce more accurate, human-like multi-hop questions.

Recent work increasingly grounds multi-hop reasoning in structured knowledge graphs to improve compositionality and interpretability. Ontology-guided and evidence-path-based KGQA models explicitly model reasoning chains over graph substructures, enabling controllable inference beyond surface-level entity linking (Liu et al., 2025; Long et al.). Some studies combine LLM-assisted reasoning with graph exploration, using reward-driven search or prompt-guided generation to identify interpretable multi-hop paths for (Long et al., 2025; Zhou et al., 2025). LLM-augmented KGQA frameworks and benchmark generators further integrate retrieval, reasoning path refinement, and dataset audit steps to improve evaluation quality, demonstrating strong performance on multi-hop benchmarks such as GrailQA and WebQSP (Zhang et al., 2025; Gu et al., 2021; Yih et al., 2016).

Chemistry Domain. The chemical sciences pose distinct challenges for multi-hop QA because they require expert domain knowledge. Only about 900 of HotpotQA's questions focus on chemistry, and those items rarely go beyond two hops or delve deeply into chemistry-specific concepts, which constrains both their difficulty and topical relevance. More specialized datasets, such as ChemLitQA (Wellawatte et al.), provide around 1,000 single-hop and 700 multi-hop question-answer pairs generated using LLMs based on ChemRxiv papers. However, ChemLitQA-multi's multi-hop questions typically rely on a single linked entity across all hops, which limits compositional complexity; the original authors therefore highlight the need for future work to develop more challenging multi-hop QA benchmarks in chemistry.

The chemistry subset of the OlympicArena benchmark (Huang et al., 2024) and the dataset from Rein et al. (2024) contain challenging problems that primarily evaluate models' chemical knowledge rather than explicit multi-hop reasoning. Amiri and Bocklitz (2025) constructs a QA dataset from recent ChemRxiv papers and takes measures to minimize pretraining data leakage; however, all their questions are single-hop. In another effort, Mirza et al. (2025) introduces ChemBench, an automated benchmark of over 2,700 expert-validated chemistry QA pairs to systematically assess LLMs' chemical reasoning against human experts. In this benchmark, chemical knowledge was more targeted than reasoning, and there was no clarity on the number of hops required to reach the answer in both question generation and answer evaluation. Humanity's Last Exam (HLE) (Phan et al., 2025) is another recent benchmark targeting the broad frontier of human knowledge across multiple modalities. However, it consists of only 2,500 questions, with approximately 7% covering chemistry, and primarily evaluates closed-ended academic questions rather than domain-specific *compositional* reasoning.

Since a direct comparison with HLE, Olympic Arena, and ChemBench would obscure the domain-specific reasoning challenges ChemComp is designed to capture, in this study we compare against HotpotQA in related experiments, as it contains a higher proportion of chemistry-related questions and shares our emphasis on multi-hop reasoning.

Knowledge Graph Generation. Automated KG construction from text progressed along

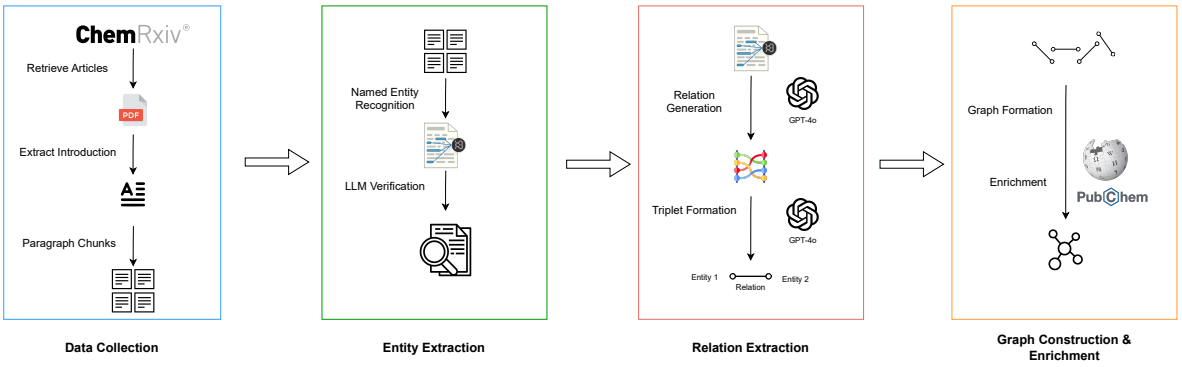

Figure 1: An Overview of the knowledge graph generation pipeline.

three complimentary directions. **(1) General, schema-light pipelines.** Early end-to-end systems generate nodes then edges with sequence models or classifiers (Melnyk et al., 2022); EDC extends this with a three-phase Extract-Define-Canonicalize process that induces relations, writes natural language definitions, and merges equivalents (Zhang and Soh, 2024). **(2) Dynamics and domain tailoring.** KG-MRC couples neural reading comprehension with recurrent graph updates to track evolving entity states in procedural text (Das et al., 2018), while domain specific frameworks (e.g., CEAR) leverage tailored ontologies to boost accuracy on scientific corpora; coarse to fine adaptation further specializes broad biomedical KGs to niches like oncology with minimal manual labels (Langer et al., 2024; Liao et al., 2023). **(3) LLM-augmented extraction.** Recent work integrates linguistic signals with LLMs. *CoDe-KG* adds sentence-complexity modeling and co-reference to lift recall on rare relations and remain domain agnostic (Anuyah et al., 2025) and scales to large scientific graphs. Mirza et al. (2025) build a large scale knowledge graph for framework materials from 100k+ papers using Qwen2-72B and pair it with RAG for 91.67% QA accuracy, highlighting the promise of LLM–KG hybrids for precise, interpretable reasoning.

3 Methodology

Our methodology comprises three main components: knowledge graph generation, multi-hop question-answer generation, and evaluation of state-of-the-art large language models on the question-answering task. The first two components are described in this section, and the last one is explained in the next section.

3.1 Knowledge Graph Generation

We began by constructing a comprehensive knowledge graph from chemical literature. First, we used the ChemRxiv API to collect all ChemRxiv articles with licenses that permitted redistribution (CC-BY 4.0). Next, we cleaned the articles using regular expressions to extract their introductions. Focusing on objective and factual information, we extracted the first few paragraphs of each introduction (up to 500 words). Finally, we segmented the extracted text into chunks of up to 128 words, ensuring that no paragraph was split across chunks.

Next, we applied named entity recognition (NER) models to these text chunks to identify chemical entities. In particular, we utilized an NER model (Ruas and Couto, 2022) that leverages a PubMedBERT architecture (Gu et al., 2020) fine-tuned on various chemical datasets. To ensure that the extracted entities were specific, verified, and chemically relevant, we utilized OpenAI's **`gpt-4o`** to review and refine the outputs. The same model was also employed to extract relations between these verified entities, forming triplets that capture the interactions and associations present in the text. Additionally, large language models were utilized to extract descriptive features from textual data associated with each entity. To enrich the nodes further during knowledge graph construction, supplementary information from Wikipedia and the PubChem dataset (Kim et al., 2021) was integrated into the nodes' attributes without creating any new edges. Consequently, the finalized knowledge graph comprises nodes representing chemical entities, enhanced by metadata and descriptive annotations from these sources, as well as edges representing the relationships extracted from the

textual data. Figure 1 illustrates the procedure followed to generate the knowledge graph. Refer to the Appendix S1.2 for a detailed explanation of Knowledge Graph Generation.

3.2 Multi-hop Question-Answer Generation

To generate multi-hop questions, we first sampled paths of varying lengths from the constructed knowledge graph using a randomized breadth-first search (BFS) path sampling algorithm. During path sampling, we ensured that the sources for the edges were distinct, encouraging solutions to integrate information from multiple sources and different parts of the context to answer the questions. Therefore, each path with a length of K involves K+1 entities, coming from K distinct source texts extracted from the original ChemRxiv database.

Adopting a bottom-up approach, we began by sampling paths and generating individual 1-hop questions from each hop. Specifically, For every $(\text{entity}_1, \text{relation}, \text{entity}_2)$ triplet, we crafted a prompt that asks which entity stands in the specified relation to entity_2, with entity_1 as the correct answer. When a question lacked sufficient specificity, we instructed an LLM to enrich it with additional metadata or context from the other sources, thereby enhancing clarity and precision. These individual questions were then combined into a single multi-hop question using OpenAI's `o3-mini` model. Importantly, the final aggregated question was constructed to begin with the last sub-question and chain the entities up to the entity_1 of the first relation, ensuring that the final answer corresponds to the answer of the first question.

During the verification phase, each one-hop question was first reviewed for clarity, relevance to chemistry, and alignment with the corresponding text that provided the answer. The multi-hop question was then assessed through an additional evaluation step, ensuring that its logical flow effectively led to the final answer. An LLM-based verification process was employed to confirm factual accuracy, answerability based on available context and metadata, and the logical coherence of the sub-questions. Feedback from domain experts was continuously incorporated into the prompts to enhance the accuracy of verification. To enhance the quality of the dataset, an additional round of verification was conducted on the final questions, resulting in the removal of a subset of questions. Figure 2 illustrates the pipeline of Multi-hop QA generation. Refer to the Appendix S1.3 for a detailed explanation of QA Generation.

To minimize the impact of writing style and summarization on accuracy evaluation, all questions are designed to have short answers. Answering these questions requires breaking down the main question into smaller sub-questions, finding the answer to each, and combining them to arrive at the final answer. Even with full context available, a correct answer cannot be obtained if the model is incapable of inferring and integrating different pieces of knowledge.

While our pipeline is largely automated, domain expert input was integral to the development process. We conducted two rounds of pilot generations followed by detailed reviews from chemists, using their feedback to improve the pipeline and significantly enhance the quality of the generated questions. Furthermore, as explained in Section 5.1, a random subset of questions from the final dataset was evaluated by a domain expert to verify the quality of the generated dataset. See the Supplementary Materials S1.4 for a statistical analysis of the generated graph and questions, as well as a comparative overview with other chemistry QA datasets.

Although complete prevention of data leakage is impossible, we took strong measures to minimize overlap with previous benchmarks. The corpus includes ChemRxiv material published after 2017 and prioritizes works whose licences permit commercial redistribution. ChemComp is designed to emphasize compositional reasoning over factual recall: each question chains together foundational chemistry knowledge with more recent findings, requiring models to synthesize information across multiple hops (e.g., combining a hop about basic properties with a hop citing contemporary literature). This structure reduces the chance that correct answers can be memorized from any single source, even if some overlap remains.

4 Experiments and Results

In our experiments, we evaluated the domain-specific multi-hop question-answering capabilities of a diverse range of state-of-the-art large language models, including both reasoning-focused and general-purpose models. For clarity, throughout this work, we refer to models specifically optimized to scale test-time compute as **reasoning models**. These included

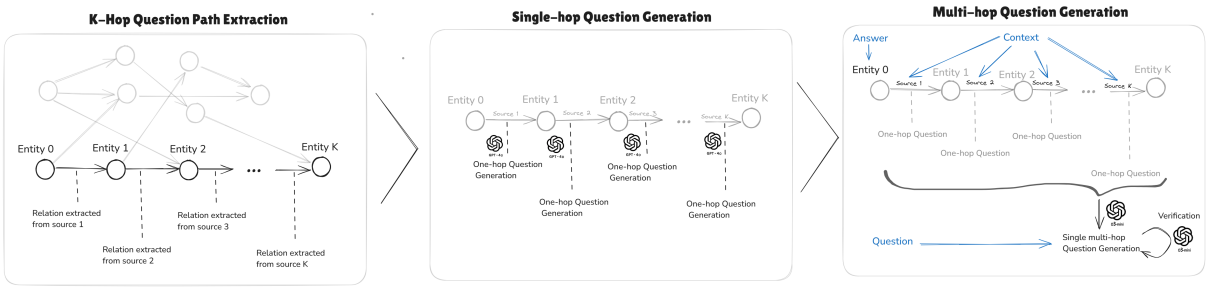

Figure 2: Overview of the QA generation Pipeline.

open-source and proprietary variants, tested with or without provided context.

To access the selected models for evaluation in this experiment, we used different API providers: (i) all tested OpenAI models (`gpt-4o`, `gpt-4o-mini`, `o1-mini`, `o3-mini` and `gpt-5`) are accessed via the OpenAI platform (OpenAI, 2025a,b); (ii) Amazon Bedrock Platform has been used to access `Anthropic Sonnet 3.7` (with and without extended thinking), `Anthropic Sonnet 3.5 V2` (Anthropic, 2025), `Mistral Large` (Jiang et al., 2023), `DeepSeek R1` (Guo et al., 2025) and `Llama 3.3 70B Instruct` (Touvron et al., 2023); and (iii) `Google Gemma 3 27B` (Team et al., 2025), `Qwen QwQ 32B` (Bai et al., 2023), and `DeepSeek R1 Distill Qwen 32B` are accessed via the OpenRouter Platform and operate at bf16 precision. All of these models can perform function-calling tool use, so they were instructed to produce valid JSON outputs to ensure consistency and enable automated validation. All models are evaluated in two settings: with and without provided context. The first scenario reflects performance when the models are paired with an ideal uni-directional RAG system, while the second scenario relies on the model's internal parametric memory to answer the questions. After parsing the JSON, the correctness of the answers was verified using a combination of exact-match evaluation and binary judgements by GPT-5-mini guided with careful instructions (including multiple human-verification phases); these judgments were used to compute the *Correctness Rate* (%). The GPT-5-mini judge checks whether the generated response matches the provided expected answer, so it does not independently assess truth beyond exact alignment. Our dataset comprises 1188 questions spanning 1 to 4 hops (On average, 299 questions per hop), generated using the approach described in Section 3.2. The full Q&A dataset, along with the evaluation code and results of human evaluations, are accessible here. All artifacts are made available under a CC BY-NC 4.0 (NonCommercial) license.

4.1 Models Performance

Figure 3 and Table S3 Summarize the performance of 13 large language models, evaluated in terms of correctness rate, cost, and latency under two setups: *gold context-provided* and *context-not-provided*. The gold-context setup simulates a perfect retrieval system by supplying each model with the gold context associated with all hops alongside the question, while the context-not-provided setup measures model performance when the models must rely solely on their internal (parametric) knowledge. The no-context results therefore primarily reflect parametric knowledge; however, they also demonstrate that compositional reasoning in ChemComp places demands beyond mere factual recall, as even state-of-the-art models with strong parametric memory fail to surpass a 50% correctness rate. Supplementary Table S3 further reports detailed, per-model performance across both evaluation scenarios.

In our evaluation, `Llama 3.3 70B Instruct` and `GPT-4o` models achieved the lowest cost and notably low latency, but also the lowest correctness rate, making them cost-efficient yet less accurate. In contrast, Claude Sonnet 3.7 (with extended thinking) achieved the highest correctness rate when provided with gold-context, albeit at the expense of significantly higher cost and latency. Meanwhile, gpt-5, QWEN 32B and Deepseek R1 Distil QWEN 32B offered a favorable balance between cost and correctness rate with gold context; among them, gpt-5 had the lowest latency. `Claude Sonnet 3.7` also showed the highest correctness rate among the none-reasoning models (models without reasoning tokens), both with and without context. This observation supports the

Figure 3: Performance of selected models based on correctness rate, cost, and latency. The cost axis is log-scaled to emphasize differences. The y-axis represents the percentage of questions answered correctly by each model, while the size of the dots indicates the average latency for each model when answering questions. The top panel (above 0.5 correctness rate) displays results for setups where context is provided to the models, while the bottom panel presents results for setups without context.

hypothesis that the regular `Claude Sonnet 3.7` shares a similar architecture and training regime with its *extended thinking* variant, differing mainly in an operational "thinking budget" that is set to zero in the regular configuration.

In *context-not-provided* setup, open-source reasoning models (`R1 Distill Qwen`, `QWQ-32B`, and `R1`) tend to have lower performance ranking compared to other models. In contrast, for OpenAI models, we observed an opposite trend, which may indicate potentially richer pre-training data. Among evaluated models, gpt-5 delivered the highest correctness rate, which may reflect its more recent release and potential incorporation of newer data. For Claude 3.7, using extended thinking only slightly improved the performance compared to the no-thinking setup when the context is not provided to the model; but it increased the output tokens, latency and cost significantly.

4.2 Comparison with HotpotQA and ChemlitQA

To demonstrate that large language models, even those designed for reasoning, often struggle with domain-specific multi-hop questions, we sampled a chemistry-related subset of HotpotQA (Yang et al., 2018), a well-known general text benchmark primarily sourced from Wikipedia. We sampled chemistry questions by starting from Wikipedia's Chemistry category, recursively exploring its subcategories (up to three levels), and then filtering HotpotQA based on exact title matches. To maintain consistency with our evaluation scheme, we excluded distractors and included only supporting documents as context. This chemistry-specific subset of HotpotQA data is made available here.

Figure 4 illustrates the average performance of all models on each dataset under two conditions:

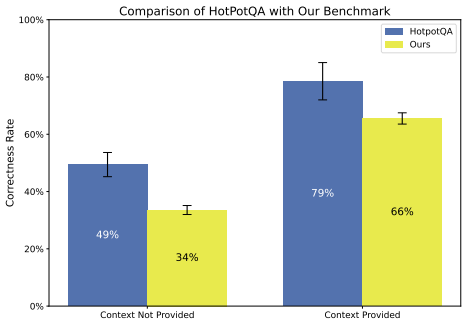

Figure 4: Comparison of LLMs' performance on the chemical subset of HotPotQA with ChemComp

with gold context provided and without context in the prompt. In both setups, models found our benchmark more challenging than the HotpotQA chemistry subset, with the gap widening when no context was provided. This likely reflects HotpotQA's reliance on Wikipedia, text included in the pretraining corpora of the evaluated models, whereas our benchmark draws from more recent ChemRxiv papers, reducing overlap with model pretraining data and increasing difficulty.

5 Analysis and Ablation

This section presents detailed ablation studies and analyses of the benchmark. We begin with the results from a manual evaluation by our panel of domain experts on a subset of the final dataset. Next, we examine the effects of context availability and test-time reasoning on model performance and efficiency. Finally, we evaluate how the number of reasoning hops, used as a proxy for question difficulty, impacts model accuracy and the number of tokens required to generate answers.

5.1 Expert Feedback

A panel of domain experts with graduate degree in Chemistry was invited to review a randomly selected subset of questions from the database. This review focused on the accuracy and quality of both the questions and their corresponding answers, as well as the necessity of multi-hop reasoning for answering them. 120 multi-hop questions, each paired with a fully worked, hop-by-hop answer were evaluated by domain experts.

These 120 questions fall into three expert-rated categories: *Perfect* (83 questions, 69%), *Good* (25 questions, 21%), and *Poor* (12 questions, 10%), generally approving 90% of the evaluated questions. For each category, the mean number of reasoning hops per question is reported in Table S5. The results indicate that more complex questions, measured by greater number of hops, are more prone to low quality. Most questions labeled as low quality were rejected based on expert feedback citing the presence of multiple valid answers. All annotation guidelines, per-question ratings, and raw model outputs are available here.

5.2 Context and Reasoning

In this analysis, we compare the performance of reasoning and non-reasoning models, i.e., models with and without test-time reasoning capabilities, respectively, in two scenarios: with context provided and without context provided. As illustrated in Figure 5-A, providing context significantly improves model performance, nearly doubling the correctness of both reasoning and non-reasoning models. Additionally, reasoning models outperform non-reasoning models in correctly answering questions, and their reasoning capabilities further benefit them in the *context-provided* setup.

To investigate a model's reasoning capability independently of input context, we compare reasoning and non-reasoning models—i.e., models with and without test-time reasoning capabilities—across two scenarios: with context provided and without context, in terms of correctness, token usage, and latency.As expected and shown in Figure 5B, non-reasoning models are faster than reasoning models. Although these non-reasoning models benefit significantly from context to improve their performance, their latency did not significantly change when context was provided to the model. This could be explained by the fact that the number of output tokens in these models is not sensitive to the length of the input prompt and the availability of context (see Figure S4). For reasoning models, we observed that the availability of context lowers latency

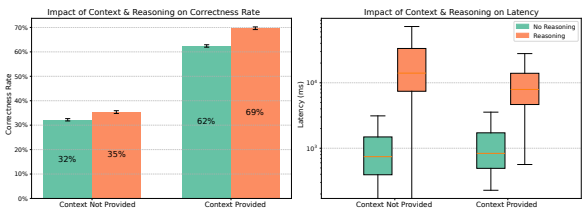

Figure 5: Impact of reasoning and context on models' Correctness Rate (left panel) and latency (right panel). Error bars represent the standard error of the mean for the models in each category.

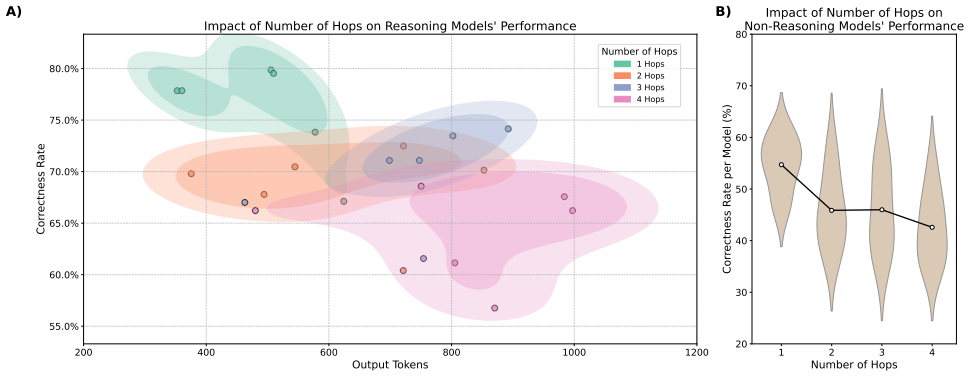

Figure 6: Analysis of the impact of the number of hops on models' performance in *Context Provided* setup. panel *A* illustrates the relationship between output token usage (x-axis) and correctness rate (y-axis) for varying numbers of hops. Each point represents a model, while the colored areas indicate distribution estimates for each hop count. Panel B, depicts the distribution of correctness rates for non-reasoning models across questions with different numbers of hops, with the dots representing the median of each distribution.

and output token count, likely because available context streamlines the thought process. For further analysis of context and reasoning in these models, refer to Appendix S1.10.

Evaluating bare LLMs without retrieval or fine-tuning establishes a clear baseline, showing how far off-the-shelf reasoning remains before incorporating more advanced methods such as enhanced prompting, supervised fine-tuning, reinforcement learning, or RAG pipelines.

5.3 Impact of the Number of Hops

In this section, we investigate how the number of reasoning hops influences the correctness rate and the output token count. Figure 6 presents the results for the setup with *Context Provided*. For reasoning models, Figure 6-A illustrates the distribution of answer correctness in relation to the number of generated output tokens. The first observation is that as the number of hops increases, the output token count, including the number of thinking tokens, also increases. Correctness rate generally decreases with more hops, although two- and three-hop questions show comparable accuracy. In non-reasoning models, output token count is insensitive to question context or complexity; so we evaluated these models solely by answer correctness. Figure 6-B depicts the distribution of answer correctness rate across these models for different numbers of hops. These models exhibit a trend similar to that seen in reasoning models.

6 Conclusion

Our study introduces ChemComp: a domain-specific multi-hop QA benchmark and an automated pipeline for generating complex scientific reasoning tasks, and uses them to evaluate state-of-the-art large language models in chemistry. Compared to established baselines, models show a marked drop in performance on our dataset, indicating that ChemComp poses more complex, domain-specific challenges. When context is unavailable, all models correctly answer fewer than half of the questions; and reasoning-fine-tuned models show only modest improvements compared to non-reasoning models. Supplying gold context substantially improves accuracy—nearly doubling performance for both reasoning and non-reasoning models, but no model reaches near-perfect performance even with gold context. We also present a fully automated pipeline that combines advanced chemical named-entity recognition, knowledge-graph construction, and constrained multi-hop question generation. The pipeline is adaptable to proprietary data in chemistry and other domains by swapping domain sources and ontologies, providing a practical, generalizable framework for creating challenging reasoning benchmarks. By releasing the knowledge graph, linked hop paths, and dataset, we offer a reusable resource for future works and evaluations also on synergized reasoning and retrieval augmented systems (Gao et al., 2025). Overall, ChemComp highlights current LLM limitations on compositional scientific reasoning and establishes a robust foundation for efforts to improve reasoning capabilities in specialized domains.

Limitations

While our work makes significant strides in evaluating multi-hop reasoning across large language models, we acknowledge six key limitations that also highlight promising directions for future research.

Model Coverage We evaluated 13 representative models available at the time of study. GPT-5, introduced later, was tested separately from other proprietary models. As the landscape of generative models continues to evolve, newer architectures, fine-tuning strategies, decoding methods, or temperature settings may yield different outcomes. This underscores the importance of ongoing benchmarking to capture emerging capabilities.

Automated Data Construction Our knowledge graph (KG) construction and question generation pipeline relies heavily on generative models, including multi-stage verification. While this approach enables scale and diversity, it may introduce noise. To mitigate this, we incorporated iterative domain-expert refinement, targeted expert reviews, and additional validation stages. These efforts substantially improved data quality, though some automation-induced bias may persist.

Evaluation with External Knowledge Augmentation Our gold-context setting provides an idealised upper bound that helps disentangle reasoning failures from retrieval errors. However, this condition does not reflect the behavior of realistic retrieval systems. Intermediate configurations—such as varying retrievers, retrieval budgets, and RAG architectures—may yield different performance trade-offs. Systematic exploration of these settings could clarify how retrieval quality and reasoning interact, and whether integrated approaches can narrow or even overcome the observed parametric limitations.

Advanced Reasoning Strategies Our evaluation focuses on bare LLMs, which isolates core parametric reasoning capabilities but does not capture the full range of techniques used in practice. Bridging the observed compositional reasoning gap may require more advanced interventions, including targeted prompt engineering, domain-specific fine-tuning, reinforcement learning from preferences, or retrieval-augmented generation. Investigating such strategies represents a natural direction for future work, and would help position ChemComp as a benchmark that supports progress toward increasingly sophisticated reasoning systems.

Computational Resources Constructing the KG, generating questions, and evaluating multiple LLMs required substantial computational resources, which may pose challenges for some research groups. To support broader accessibility, we plan to release the KG, dataset, generation pipeline, and evaluation scripts. This will allow others to reproduce our results with reduced compute—by evaluating a subset of models or using smaller local models—and to extend benchmarking with new approaches and models.

Risk Associated with Expanding Beyond the Current Setup Our current setup demonstrates strong potential for generating multi-hop questions across proprietary datasets, as well as in other domains and languages. However, extending the pipeline to new contexts requires careful attention to updating all components to reflect the specific characteristics of the new use case. To ensure reliability and avoid producing incomplete or inaccurate results, it is essential to incorporate domain expert feedback throughout the refinement process. While the pipeline is designed to be adaptable, thoughtful calibration and validation remain critical to maintaining quality and avoiding unintended risks in downstream decision-making.

Acknowledgments

The author(s) gratefully acknowledge the financial support for the research, authorship, and/or publication of this article provided by *MITACS* under funding number IT32409. We also thank *Adam Wojciech Bartwiki* for his leadership and project management expertise; *Tobias Roth* for his essential contributions to infrastructure; *Stephen Dokas* for his invaluable recommendations in chemistry; and *Amirreza Behzad Moghadam* for his invaluable inputs as a chemistry subject matter expert. We used AI-assisted tools for proofreading and improving the clarity of the manuscript.

References

Mahmoud Amiri and Thomas Bocklitz. 2025. Chemrxivquest: A curated chemistry question-answer database extracted from chemrxiv preprints. *arXiv preprint arXiv:2505.05232*.

Anthropic. 2025. Claude sonnet 3.7, cloude sonnet 3.5 v2. https://www.anthropic.com. Accessed March 2025.

Sydney Anuyah, Mehedi Mahmud Kaushik, Krishna Dwarampudi, Rakesh Shiradkar, Arjan Durresi, and Sunandan Chakraborty. 2025. Automated knowledge graph construction using large language models and sentence complexity modelling. *arXiv preprint arXiv:2509.17289*.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and 1 others. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Rajarshi Das, Tsendsuren Munkhdalai, Xingdi Yuan, Adam Trischler, and Andrew McCallum. 2018. Building dynamic knowledge graphs from text using machine reading comprehension. *arXiv preprint arXiv:1810.05682*.

Artur d'Avila Garcez and Luis C Lamb. 2020. Neurosymbolic ai: the 3rd wave. *arXiv e-prints*, pages arXiv–2012.

Yunfan Gao, Yun Xiong, Yijie Zhong, Yuxi Bi, Ming Xue, and Haofen Wang. 2025. Synergizing rag and reasoning: A systematic review. *arXiv preprint arXiv:2504.15909*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Yu Gu, Sue Kase, Michelle Vanni, Brian Sadler, Percy Liang, Xifeng Yan, and Yu Su. 2021. Beyond iid: three levels of generalization for question answering on knowledge bases. In *Proceedings of the web conference 2021*, pages 3477–3488.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2020. Domain-specific language model pretraining for biomedical natural language processing.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. *arXiv preprint arXiv:2011.01060*.

Zhen Huang, Zengzhi Wang, Shijie Xia, and Pengfei Liu. 2024. Olympicarena medal ranks: Who is the most intelligent ai so far? *arXiv preprint arXiv:2406.16772*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Theophile Le Scao, Thibaut Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2023. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*.

Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, and 1 others. 2021. Pubchem in 2021: new data content and improved web interfaces. *Nucleic acids research*, 49(D1):D1388–D1395.

Stefan Langer, Fabian Neuhaus, and Andreas Nürnberger. 2024. Cear: Automatic construction of a knowledge graph of chemical entities and roles from scientific literature. *arXiv preprint arXiv:2407.21708*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis,

Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Maodong Li, Longyin Zhang, and Fang Kong. 2025. Multi-hop question generation via dual-perspective keyword guidance. *arXiv preprint arXiv:2505.15299*.

Wenxiong Liao, Zhengliang Liu, Yiyang Zhang, Xiaoke Huang, Fei Qi, Siqi Ding, Hui Ren, Zihao Wu, Haixing Dai, Sheng Li, and 1 others. 2023. Coarse-to-fine knowledge graph domain adaptation based on distantly-supervised iterative training. In *2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1294–1299. IEEE.

Runxuan Liu, Luobei Luobei, Jiaqi Li, Baoxin Wang, Ming Liu, Dayong Wu, Shijin Wang, and Bing Qin. 2025. Ontology-guided reverse thinking makes large language models stronger on knowledge graph question answering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15269–15284.

Xiao Long, Liansheng Zhuang, Aodi Li, Minghong Yao, and Shafei Wang. Eperm: An evidence path enhanced reasoning model for knowledge graph question and answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 12282–12290.

Xiao Long, Liansheng Zhuang, Chen Shen, Shaotian Yan, Yifei Li, and Shafei Wang. 2025. Enhancing large language models with reward-guided tree search for knowledge graph question and answering. *arXiv preprint arXiv:2505.12476*.

Vaibhav Mavi, Anubhav Jangra, Adam Jatowt, and 1 others. 2024. Multi-hop question answering. *Foundations and Trends® in Information Retrieval*, 17(5):457–586.

Igor Melnyk, Pierre Dognin, and Payel Das. 2022. Knowledge graph generation from text. *arXiv preprint arXiv:2211.10511*.

Adrian Mirza, Nawaf Alampara, Sreekanth Kunchapu, Martiño Ríos-García, Benedict Emoekabu, Aswanth Krishnan, Tanya Gupta, Mara Schilling-Wilhelmi, Macjonathan Okereke, Anagha Aneesh, and 1 others. 2025. A framework for evaluating the chemical knowledge and reasoning abilities of large language models against the expertise of chemists. *Nature Chemistry*, pages 1–8.

OpenAI. 2024a. Openai o1 system card. Accessed: 2025-03-20.

OpenAI. 2024b. Openai o3 mini system card. Accessed: 2025-03-20.

OpenAI. 2025a. Gpt-5. https://openai.com. Accessed October 2025.

OpenAI. 2025b. Openai language models: Gpt-4o, gpt-4o-mini, o1-mini, o3-mini. https://openai.com. Accessed March 2025.

Avinash Patil. 2025. Advancing reasoning in large language models: Promising methods and approaches. *arXiv preprint arXiv:2502.03671*.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, and 1 others. 2025. Humanity's last exam. *arXiv preprint arXiv:2501.14249*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2024. Gpqa: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Pedro Ruas and Francisco M Couto. 2022. Nilinker: attention-based approach to nil entity linking. *Journal of Biomedical Informatics*, 132:104137.

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. 2017. A simple neural network module for relational reasoning. *Advances in neural information processing systems*, 30.

Yixuan Tang and Yi Yang. 2024. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries. *arXiv preprint arXiv:2401.15391*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

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

Geemi Wellawatte, Huixuan Guo, Magdalena Lederbauer, Anna Borisova, Matthew Hart, Marta Brucka, and Philippe Schwaller. Chemlit-qa: A human evaluated dataset for chemistry rag tasks. In *AI for Accelerated Materials Design-NeurIPS 2024*.

Violet Xiang, Charlie Snell, Kanishk Gandhi, Alon Albalak, Anikait Singh, Chase Blagden, Duy Phung, Rafael Rafailov, Nathan Lile, Dakota Mahan, and 1 others. 2025. Towards system 2 reasoning in llms: Learning how to think with meta chain-of-though. *arXiv preprint arXiv:2501.04682*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Wen-tau Yih, Matthew Richardson, Christopher Meek, Ming-Wei Chang, and Jina Suh. 2016. The value of semantic parse labeling for knowledge base question answering. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 201–206.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488.

Bowen Zhang and Harold Soh. 2024. Extract, define, canonicalize: An llm-based framework for knowledge graph construction. *arXiv preprint arXiv:2404.03868*.

Liangliang Zhang, Zhuorui Jiang, Hongliang Chi, Haoyang Chen, Mohammed Elkoumy, Fali Wang, Qiong Wu, Zhengyi Zhou, Shirui Pan, Suhang Wang, and 1 others. 2025. Diagnosing and addressing pitfalls in kg-rag datasets: Toward more reliable benchmarking. *arXiv preprint arXiv:2505.23495*.

Wenzhuo Zhao and Shuangyin Li. 2025. Ruby: An effective framework for multi-constraint multi-hop question generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 18164–18188.

Zhiling Zheng, Nakul Rampal, Theo Jaffrelot Inizan, Christian Borgs, Jennifer T Chayes, and Omar M Yaghi. 2025. Large language models for reticular chemistry. *Nature Reviews Materials*, pages 1–13.

Xiujun Zhou, Pingjian Zhang, and Deyou Tang. 2025. Pgda-kgqa: A prompt-guided generative framework with multiple data augmentation strategies for knowledge graph question answering. *arXiv preprint arXiv:2506.09414*.

S1 Appendix

S1.1 Released Dataset

The extracted knowledge graph, question–answer pairs, sub-questions, and entity links used in ChemComp, together with the evaluation code and human annotation results, are publicly available at this repository. All released resources are distributed under the Creative Commons Attribution–NonCommercial 4.0 International (CC BY-NC 4.0) license. The benchmark covers the chemistry domain and is provided in English.

S1.2 Detailed Knowledge Graph Generation

In this section, we explain each step in our graph generation process, illustrating how unstructured chemical text is transformed into a structured representation suitable for downstream tasks.

S1.2.1 Text Preprocessing

Our preprocessing pipeline is designed to isolate the most informative portions of ChemRxiv articles and prepare them for entity and relation extraction. First, we retrieved all ChemRxiv articles whose licenses permit redistribution via the ChemRxiv API. Using regular expressions, we extracted the introduction sections of each paper, as these typically focus on concise, objective statements about chemical phenomena. From each introduction, we extracted the first 500 words; this limit strikes a balance between covering key background information and minimizing the inclusion of less relevant or overly general text. To avoid fragmenting paragraphs, we ensured that no paragraph was split across chunks. Finally, we segmented each introduction into contiguous chunks of up to 128 words, preserving natural sentence boundaries. This segmentation reduces the input length for subsequent processing steps, ensuring that each chunk remains within the token limits of our downstream models while retaining semantic coherence.

S1.2.2 Node Extraction

Once the text is segmented, we identify chemical entities and their interrelations. We apply a named entity recognition (NER) model based on the PubMedBERT architecture, fine-tuned on multiple chemical datasets, to detect candidate entities within each 128-word chunk. To further improve precision and eliminate spurious mentions, we refine the NER output using OpenAI's `gpt-4o`, prompting it with a few shots to verify whether each detected span indeed corresponds to a chemically meaningful entity and to standardize entity labels (for instance, converting "MeOH" to "methanol"). Below, we showed the prompt for Entity verification.

> You are a chemistry expert specializing in entity recognition. Your task is to **validate and filter** the extracted entities, ensuring they are **chemically meaningful** based on the provided text. Remove any irrelevant terms, including general descriptors, numerical values, reaction conditions, and vague terms.
>
> **Entities Extracted by NER:**
> `{entities}`
>
> **Text for Context:**
> `{text}`
>
> **Criteria for Valid Entities:**
>
> ✓ Chemical compounds (e.g., *HCl*, *Sodium hydroxide*, *Ethanol*, *Benzene*)
>
> ✓ Chemical elements (e.g., *Carbon*, *Oxygen*, *Cesium*)
>
> ✓ Specific catalysts, solvents, reagents (e.g., Cs_2CO_3, *Toluene*, *Palladium*)
>
> **Remove the Following Types of Entities:**
>
> ✗ Generic terms (e.g., *Reaction*, *Solvent*, *Acid*, *Base*, *Solution*)
>
> ✗ Experimental conditions (e.g., *pH*, *Temperature*, *2 M*, *Strong acid*)
>
> ✗ Measurement/technique terms (e.g., *X-ray diffraction*, *NMR*)
>
> ✗ General descriptors (e.g., *High concentration*, *Low efficiency*)
>
> **Output Format:**
> Return only a **Python list** of valid chemical entities—no explanations, markdown, or extra formatting.

S1.2.3 Edge Extraction

After obtaining a vetted set of entities, we extract pairwise relations using the same `gpt-4o` instance. For each pair of entities co-occurring within a chunk, we prompt the model with a few shots to

classify or generate the nature of their relationship, producing triplets of the form (entity_A, relation, entity_B). The result of this step is a set of entity nodes (chemical names) and directed edges (relation labels) extracted directly from the text, forming a raw triplet collection that reflects the functional associations present in the chemical literature. Below, we show the prompt for Edge Extraction.

You are an expert in chemical text analysis. Your task is to extract **only chemically meaningful relationships** between a given set of entities from the provided text.

Guidelines for Relation Extraction:

1. **Entity Matching:** Consider only the entities in the given set. If an entity appears in the text but has no meaningful chemical relationship with another entity in the set, ignore it.
2. **Chemically Significant Relations Only:** Extract relations describing **interactions, transformations, or properties** (e.g., "reacts with," "catalyzes," "dissolves in," "produces").
3. **Factual Relations:** Only extract factual relations; avoid observations or opinions.
4. **Tuple Format:** Output facts as **(entity1, relation, entity2)**.
5. **Avoid Generic Relations:** Exclude weak relations like "is," "are," "exists," "relates to."

Valid Relation Types (Examples):

✓ reacts with

✓ catalyzes

✓ binds to

✓ dissolves in

✓ oxidizes

✓ inhibits

✓ precipitates with

✓ acts as a solvent for

✓ is synthesized from

Entities Provided:
`{entities}`

Text:
`{text}`

Extract at most `{max_facts}` factual statements.

Output Format:
Provide a **Python list of tuples**, only the extracted relationships (no code fences/backticks).

Example Output:

[("HCl","dissolves in","Water"), ("HCl","reacts with","Sodium hydroxide")]

S1.2.4 Knowledge Enrichment

To enrich each node with descriptive metadata and resolve naming ambiguities, we retrieved supplemental information from two external resources: Wikipedia and PubChem. From Wikipedia, we extracted the introductory summary of each entity's page, providing a concise description of its common usage, historical context, or primary function. From PubChem, we obtained the official compound name (Record Title) along with additional names and identifiers sourced from the "Names and Identifiers" section. We retrieved a textual description from the "Record Description" heading, summarizing key properties or applications. Safety annotations, including hazard statements or pictograms, were collected from the "Chemical Safety" subsection. Structural information was captured in the form of the canonical SMILES string under "Computed Descriptors," and the molecular formula was fetched from the "Molecular Formula" field. Finally, we extracted computed physicochemical properties, such as molecular weight, topological polar surface area, and log P values, from the "Computed Properties" list within "Chemical and Physical Properties." All of these metadata fields were stored as additional text and added as external information for each node.

S1.2.5 Graph Generation

At this stage, we constructed the graph by forming triplets of the form (node, edge, node). Each node was linked to the original text segment from

which it was extracted and, if available, to its corresponding external metadata. Likewise, each edge was associated with the specific text chunk that produced it. In this way, both nodes and edges maintain direct references to their source text, ensuring traceability throughout the knowledge graph.

S1.3 Detailed Question Generation

S1.3.1 Path Sampling from the Knowledge Graph

To generate multi-hop questions, we first sampled paths of varying lengths from the constructed knowledge graph using a randomized breadth-first search (BFS) path sampling algorithm. During path sampling, we enforced that each edge in a sampled path originates from a distinct source text, thereby encouraging the integration of information from multiple, separate context passages. Concretely, a path of length K comprises $K + 1$ entities and traces through K different source texts extracted from the original ChemRxiv database. By requiring unique sources for each edge, we ensure that correct solutions must draw on evidence scattered across several documents rather than a single paragraph.

S1.3.2 One-Hop Question Formulation

Adopting a bottom-up approach, we generated individual one-hop questions for each edge along a sampled path. For every triplet $(\text{entity}_1, \text{relation}, \text{entity}_2)$, we framed a question whose answer is entity_1 by asking, "Which entity holds the relation relation to entity_2?" If the initial phrasing lacked sufficient specificity or clarity, we invoked a language model (OpenAI's `o3-mini`) to augment the prompt with metadata drawn from the original text segment, such as contextual phrases or qualifying details. This enrichment step ensures that each one-hop question remains clear, precise, and answerable from the associated source text alone. Below is the prompt for generating the one-hop questions.

> You are given a text along with an entity and its relation to another entity.
>
> **Entity 1:** `{entity1}`
> **Relation:** `{relation}`
> **Entity 2:** `{entity2}`
> **Text:** `{text}`
> **Information about Entity 1:** `{entity1_meta` if `entity1_meta` else `None}`
>
> **Task.** Generate a **factual** question whose answer is **Entity 1**. The question must:
>
> - ask for the entity that has the specified **Relation** to **Entity 2**;
> - **not** mention the answer (Entity 1) in the question text;
> - be answerable **solely** from the provided **Text** and (optionally) the **Information about Entity 1**;
> - avoid meta references such as "Abstract," "Table #1," "in the text," or "in the article."
>
> If **Entity 1** and the **Relation** are not specific enough (i.e., multiple answers are possible), incorporate precise descriptors from the **Text** and/or the **Information about Entity 1** so that **Entity 1 is uniquely determined**.
>
> **Output.** Return exactly a dictionary (no code fences/backticks/markdown) with keys `"q"` and `"a"`:
>
> - `"q"`: the question you generated;
> - `"a"`: `{entity1}`.

S1.3.3 Multi-Hop Question Aggregation

Once all one-hop questions for a path were vetted, we combined them into a single multi-hop question by prompting OpenAI's `o3-mini` model, with a few shots. The final aggregated question is constructed by starting with the sub-question corresponding to the last edge (i.e., the entity nearest to the "tail" of the path) and then chaining backward through each preceding entity until reaching entity_1 of the first relation. In other words, if the sampled path is $(\text{entity}_1 \xrightarrow{\text{relation}_1} \text{entity}_2), (\text{entity}_2 \xrightarrow{\text{relation}_2} \text{entity}_3), \ldots, (\text{entity}_K \xrightarrow{\text{relation}_K} \text{entity}_{K+1})$ the aggregated question asks first about entity_{K+1} (the final tail), then uses its answer as context for the penultimate question, and so on, so that the final answer corresponds to entity_1. This reverse-chaining structure ensures that the multi-hop prompt leads directly to the original target node while preserving logical flow. Below, we show the prompt for generating the multi-hop question.

You are given multiple factual questions and their answers that are logically connected. Your task is to chain them into a single, coherent **multi-hop question** that requires multiple reasoning steps. The **only** correct answer must be the answer to the **first** question. Do not include any of the answers in the generated question text.
Start from the last generated question and build upward so the final question aggregates all steps, yet its answer remains the first question's answer.

Example

- Q1: What is oxidized to form Carbon Dioxide? A1: Methane
- Q2: What is used in Photosynthesis? A2: Carbon Dioxide
- Q3: What produces Oxygen? A3: Photosynthesis

Multi-hop question:
Q: What is oxidized to produce a substance that is used in a process that results in Oxygen?
A: Methane

Generated Q&A (input):
`{formatted_qas}`

Output Return exactly a Python dictionary (no code fences/backticks/markdown) with keys:

- `"q"` — the final multi-hop question,
- `"a"` — the final answer (must equal the first question's answer).

S1.3.4 Verification and Filtering

During verification, we first reviewed each one-hop question for clarity, chemical relevance, and direct answerability based on its corresponding source text. Below is the prompt for evaluating one-hop questions.

You are a chemistry expert. Your task is to determine whether the given question is **(i)** a factual chemistry question, **(ii)** unambiguous (has only one correct answer), and **(iii)** answerable from the provided context. A factual question is grounded in actual chemical properties, reactions, or experimentally verified principles and is strictly related to chemistry. An answerable question must be solvable using the given context and must not be open-ended or have multiple correct answers. **Make sure the question has only one correct answer.** There should not be any other entity besides the given answer that could also be correct.

Question:
`{question}`

Answer:
`{answer}`

Context:
`{context}`

Please analyze the context and verify if the question is factual, unambiguous, and answerable. If the question is factual, has only one correct answer, is strictly related to chemistry, and can be answered based on the context, return `yes`. Otherwise, return `no`.

Examples of Factual Chemistry Questions:

✓ What dissolves in water and evaporates at $0\,^{\circ}\mathrm{C}$?

✓ What catalyst is used in the reaction between A and B?

Examples of Non-Factual or Ambiguous Questions:

✗ What is the song of Nirvana that is a chemical entity?

✗ What chemical entity *and* structural unit form the layered hydroxide structures with intercalated water ions used in battery materials and OER catalysis? (Both $M(OH)_6$ and α-$Ni(OH)_2$ are valid answers.)

✗ Any question with multiple possible correct answers or not strictly related to chemistry.

Next, the multi-hop question underwent an additional evaluation step to confirm that the logical chain, from the final sub-question back to the first, correctly guides a reader (or model) to entity_1. We employed an LLM-based verification process to assess factual accuracy, answerability given available context and metadata, and overall coherence among sub-questions. Feedback from domain experts was continuously incorporated into the prompt templates to refine the accuracy of verification. Any question, either one-hop or multi-hop, that was answered incorrectly by all evaluated models was excluded from the benchmark to minimize ambiguity and ensure high-quality, unambiguous reasoning tasks. Below

is the prompt for evaluating the whole path.

You are a chemistry expert. Decide whether the question is **(i)** a factual chemistry question and **(ii)** answerable from the provided path. A factual question is grounded in real chemical properties, reactions, or experimentally verified principles (strictly within chemistry). An answerable question must be solvable using the given path information (not open-ended or opinion-based).
Path information:
`{path_text}`
Question:
`{question}`
Answer:
`{answer}`
Analyze the path and decide if the question is factual and answerable from it. If *both* hold, return `yes`; otherwise return `no`.
Examples of factual chemistry questions:
✓ What dissolves in water?
✓ What catalyst is used in the reaction between A and B?
✓ Which compound undergoes oxidation in this reaction?
✓ What product is formed when sodium reacts with chlorine?
Examples of non-factual / opinion-based questions:
✗ Why do some scientists think this reaction is inefficient?
✗ What is the best solvent for this reaction?
✗ Is this reaction useful in industry?
✗ Do you think this compound is a good catalyst?
Output: Provide only `yes` or `no`.

S1.3.5 Short-Answer Design and Rationale

To minimize the impact of writing style and summarization on accuracy evaluation, all questions were deliberately designed to elicit short, concise answers. Answering a multi-hop question requires decomposing it into its constitutive one-hop sub-questions, retrieving each intermediate answer, and then combining them to arrive at the final answer. Even when full context is available, a correct response cannot be produced if a model fails to infer and integrate multiple pieces of knowledge. By keeping answers brief and focusing on discrete factual steps, we ensure that performance evaluation reflects a model's ability to conduct multi-hop inference rather than its capacity to paraphrase or generate lengthy explanations.

S1.4 Statistical Details of the Generated Graph and Dataset

Table S1 provides a concise overview of both our dataset-level and graph-level statistics. In Table S1a, we summarize key properties of the 971 multi-hop questions, including average question and answer lengths (in characters and tokens), the mean number of hops per question, total and pooled context lengths, and the proportion of questions containing at least one shortcut edge.

On average, questions were 319.4 chars long (SD = 129.2) or 45.5 tokens (SD = 18.6), while answers averaged just 16.7 chars (SD = 9.7) or 1.76 tokens (SD = 0.96). Each question required a mean of 2.45 hops (SD = 1.12). Summing all hops per question yields a total context length of $5,993$ characters (SD = $5,009$) or 849 tokens (SD = 726), and the pooled hop lengths average $2,447$ characters (SD = $2,222$) and 346 tokens (SD = 324). Only 96 questions (9.9 %) include at least one shortcut edge (mean = 0.12, SD = 0.38). The full hop-count breakdown appears in the "Hop-count Distribution" block of Table S1a.

Table S1b then reports the main network characteristics of the underlying knowledge graph: its size (nodes and edges), sparsity (density), degree distribution (min, max, and average), number of connected components, and the size of the largest component, as well as clustering and assortativity coefficients. The graph contains $14,523$ nodes and $13,419$ edges (density = 0.000127), with node degrees ranging from 0 to 257 (mean = 1.85). It splits into 4,684 connected components, the largest of which spans $7,318$ nodes. The average clustering coefficient is 0.0298, and the degree assortativity coefficient is –0.0265. Finally, the five highest-degree nodes, hydrogen, carbon, oxygen, CO_2, and lithium, are listed to highlight the most central concepts in the graph.

Table S2 compares our chemical dataset (ChemKGMultiHopQA) with HotpotQA-Chemistry and ChemLitQA-multi across question count, bridged entities, entity types, answer format, domain, and source. ChemKGMultiHopQA comprises 971 ChemRxiv questions enhanced by PubChem and Wikipedia (1–4 hops, auto-built KG with expert-verified subset), offering richer multi-hop chemical

QA Metric	Mean	Std. Dev.
Question length (tokens)	80.0	32.2
Answer length (tokens)	5.4	6.0
Mean # hops per question	2.45	1.12
Total context length (tokens)	1480.2	1244.5
Shortcut count per question	0.12	0.38
Hop-count Distribution (of 1188 questions)		
1 hop	299 (26.6%)	
2 hops	299 (25.2%)	
3 hops	300 (24.9%)	
4 hops	300 (23.3%)	
Questions w/ $\geq$ 1 shortcut	96 (9.9%)	

(a) Dataset-level statistics for multi-hop questions

Graph Metric	Value
Number of nodes	14 523
Number of edges	13 419
Density	0.000127
Degree (min / max / avg)	0 / 257 / 1.85
Connected components	4 684
Largest component size	7 318
Avg. clustering coefficient	0.0298
Degree assortativity coefficient	–0.0265
Top 5 nodes by degree	
hydrogen (257), carbon (250), oxygen (232), CO_2 (220), lithium (155)	

(b) Key network-level properties of the loaded knowledge graph

Table S1: Overview of both dataset-level and graph-level statistics. Left panel: Table S1a summarizes the dataset-level statistics for our 971 multi-hop questions. Right panel: Key properties of the underlying knowledge graph are reported in Table S1b.

reasoning than HotpotQA-Chemistry (980 Wikipedia questions, 2 hops, no chemical entities) and ChemLitQA-multi (742 ChemRxiv questions, 1 entity, LLM+expert verification).

S1.5 Summary of models' performance

The summary of tested models and their performance is provided in Table S3.

S1.6 Detailed Performance Based on Context Availability

Figure S1 shows that model performance is strongly influenced by whether context is provided in the input. In particular, Claude 3.7 with extended thinking achieved a correctness rate of 84% with context, whereas `o3-mini` recorded the highest correctness rate (48%) when the context was absent. Note that `o3-mini` was primarily used to generate the questions, which may have introduced a slight bias, resulting in its minor improvement in correctness. Figures S2 and S3 illustrate the token usage and latency of the models, respectively.

S1.7 Performance of models on Chemistry Subset of HotpotQA

Table S4 shows the details of each model's performance, latency, and tokens used in both setups of context provided and not provided for the chemistry subset of HotpotQA (Yang et al., 2018) questions.

Dataset	# Qs	# bridged entities	# entity type	Answer type	Domain	Source
HotpotQA-Chemistry	980	2	General	Short	Wikipedia	Crowd (Wiki)
ChemLitQA-multi	742	1	Chemistry	Long & Short	ChemRxiv	LLM + expert verified
ChemKGMultiHopQA	971	1–4	Chemistry	Short	ChemRxiv enhanced by PubChem & Wikipedia	LLM + NER + KG (auto) + An expert verified subset

Table S2: Comparison of HotpotQA, ChemLitQA-multi, and ChemKGMultiHopQA data sets.

Model	Correctness Rate (%)		Avg Duration (s)		Avg Input Tokens		Avg Output Tokens	
	No Ctx	Gold Ctx	No Ctx	Gold Ctx	No Ctx	Gold Ctx	No Ctx	Gold Ctx
Llama 3.3 70B Instruct	25.13	53.54	0.33	0.40	330	1770	11	11
Mistral Large	32.29	72.60	0.41	0.57	178	1898	14	15
QwQ-32B	31.11	71.08	69.23	27.66	168	1653	2194	830
R1	39.04	72.51	21.54	9.35	159	1540	1498	624
R1 Distill Qwen-32B	30.27	70.24	31.67	12.54	159	1624	1060	418
Sonnet 3.5	32.21	62.90	1.53	1.67	567	2196	30	30
Sonnet 3.7	37.52	68.13	1.61	1.83	567	2196	30	30
Sonnet 3.7 Thinking	39.80	73.27	39.88	16.11	583	2214	1816	754
gemma-3-27b-it	27.57	61.30	0.89	0.99	164	1576	12	13
gpt-4o	32.29	56.32	0.64	0.69	205	1617	10	10
gpt-4o-mini	25.38	52.78	0.63	0.69	205	1617	10	10
gpt-5	45.11	71.68	22.45	9.89	204	689	1565	713
o1-mini	32.55	61.47	7.96	5.84	161	1598	1070	742
o3-mini	39.38	69.48	11.95	6.43	211	1623	1231	601

Table S3: Summary of tested models' performance in terms of Correctness rate, average latency, average input tokens, and average output tokens for both Contextual and Non-Contextual Setups

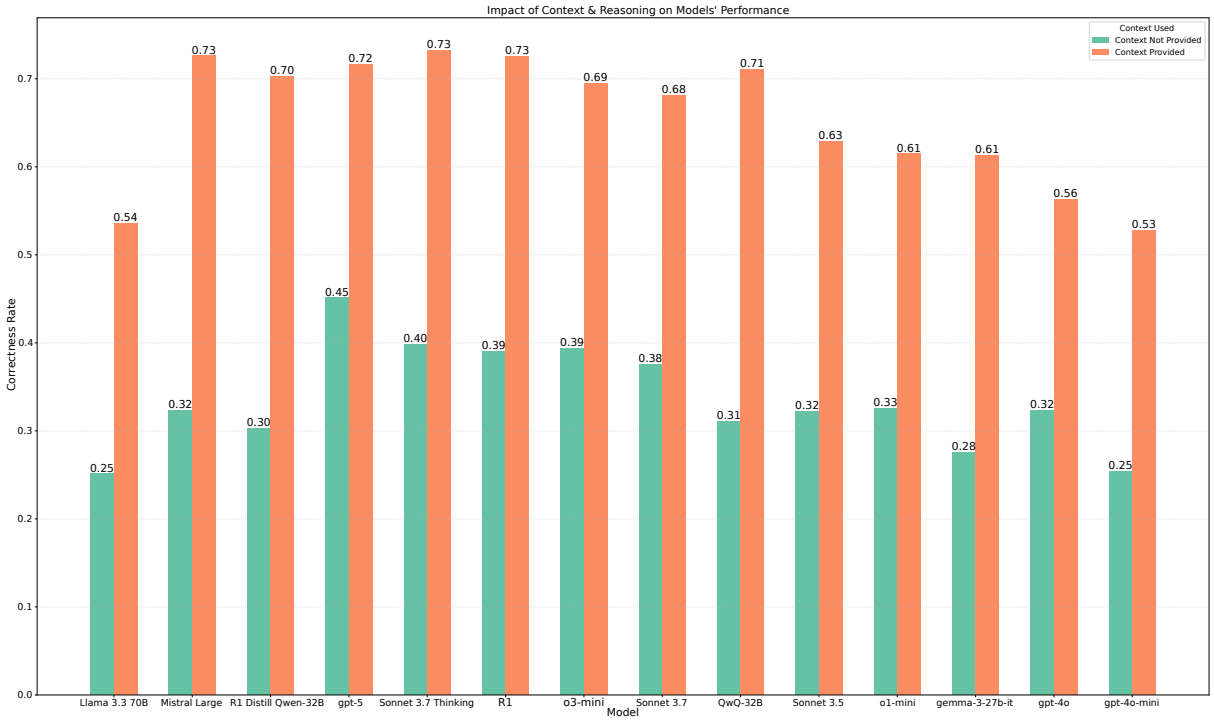

Figure S1: Correctness rate of models with respect to context.

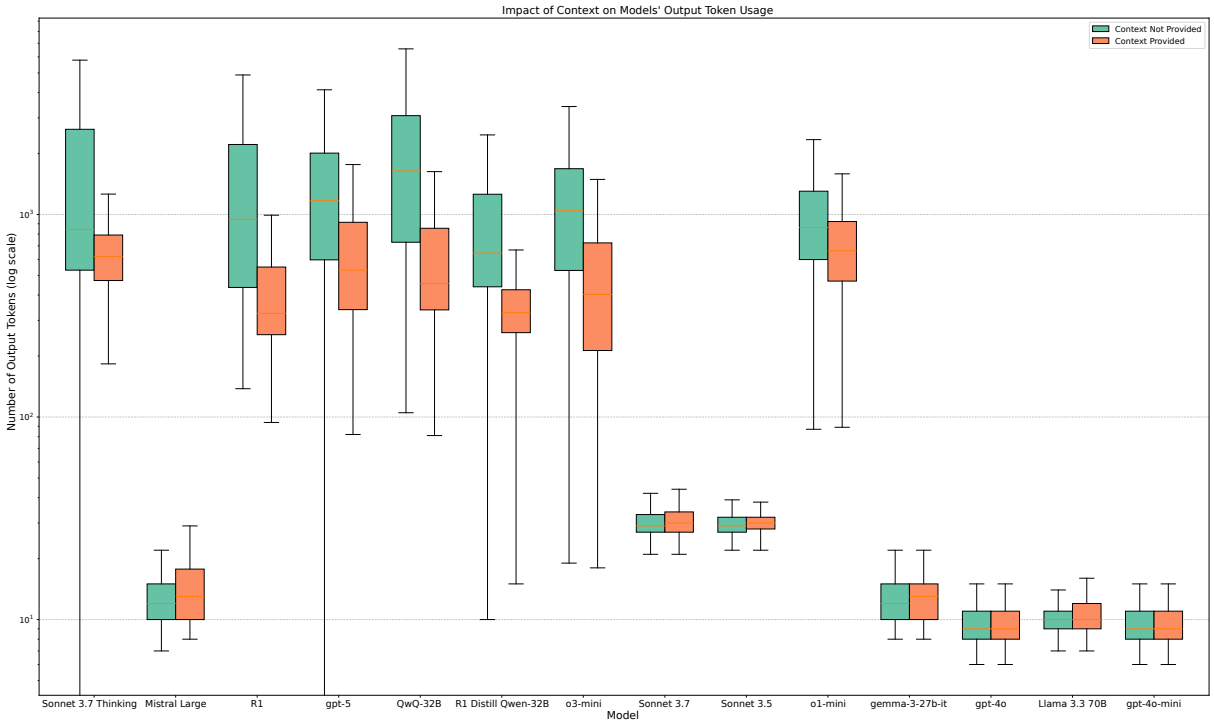

Figure S2: Token usage of models with respect to context

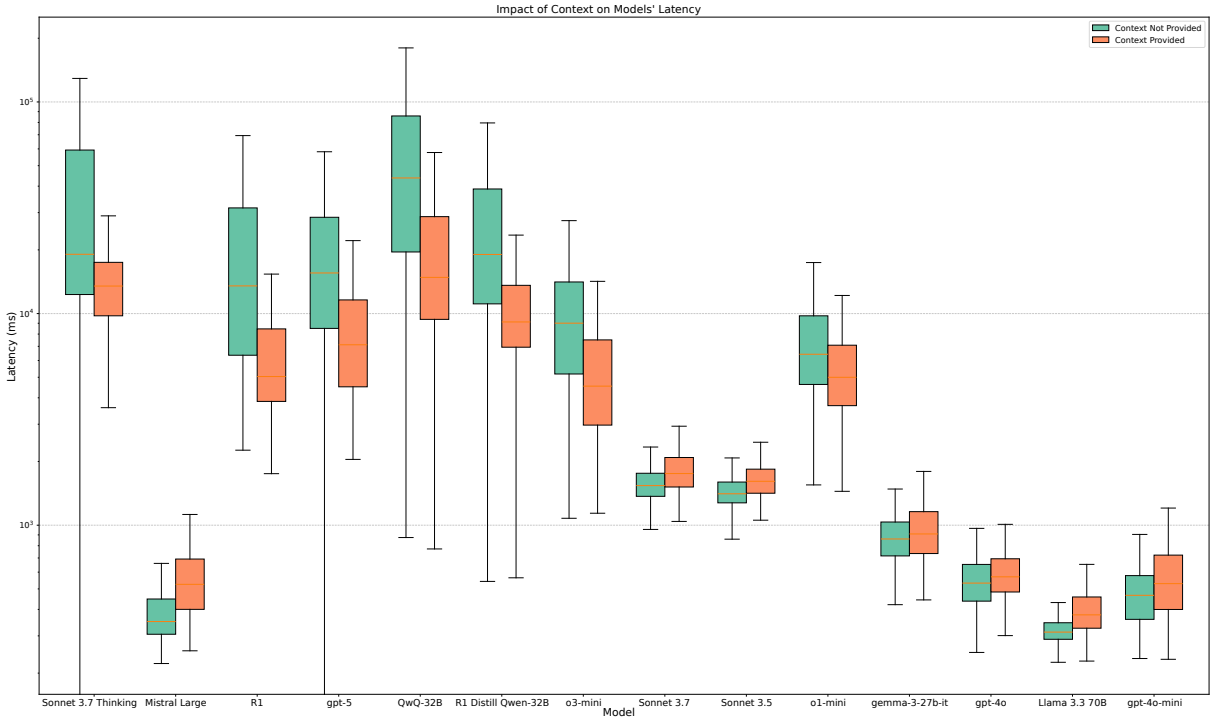

Figure S3: Latency with respect to context

Model	Correctness Rate (%)		Avg Duration (s)		Avg Input Tokens		Avg Output Tokens	
	No Ctx	Gold Ctx	No Ctx	Gold Ctx	No Ctx	Gold Ctx	No Ctx	Gold Ctx
Anthropic Claude Sonnet 3.5 V2	53.78	85.31	1.26	1.32	517	618	29	29
Anthropic Claude Sonnet 3.7	59.49	86.73	1.87	1.94	517	618	29	29
Anthropic Claude Sonnet 3.7 (Thinking)	66.33	87.96	17.54	10.10	539	640	726	390
OpenAI GPT-4o-mini	45.41	80.41	0.43	0.50	170	257	7	8
OpenAI GPT-4o	55.51	82.24	0.62	0.63	170	257	8	8
OpenAI o1-mini	51.94	86.12	4.82	3.41	127	217	719	426
OpenAI o3-mini	60.00	87.86	9.65	3.74	166	253	977	247
Mistral Large	5.00	1.12	0.49	0.36	198	303	18	12
Llama 3.3 70B Instruct	44.90	79.59	0.32	0.29	284	373	9	9
Google Gemma 3 27B	40.31	80.31	0.77	0.82	127	218	10	11
DeepSeek R1	60.10	86.12	8.93	5.12	125	212	612	358
Qwen QwQ 32B	52.76	89.39	27.91	11.01	126	219	865	412
DeepSeek R1 Distill Qwen 32B	46.84	87.55	16.20	7.98	119	208	565	287

Table S4: Summary of tested models' performance on the HotpotQA chemistry subset in terms of Correctness rate, average latency, average input tokens, and average output tokens for both Contextual and Non-Contextual Setups

S1.8 Collected Feedback on questions quality from domain experts

Table S5 summarize the collected feedbacks on questions quality from domain experts and the average number of hops presented in questions categorized in each group.

Cat.	Num. Que.	Avg. len.
Good	83 (69%)	2.46
Ok	25 (21%)	2.55
Poor	12 (10%)	2.60

Table S5: The 40 high-confidence questions are grouped by expert-rated quality (*Good*, *Ok*, *Poor*). **Num. Que.** shows the count and percentage of questions in each category. **Avg. len.** is the average number of reasoning hops per question.

For failure analysis, we selected 50 questions from the hard subset (answered incorrectly by at least 14 models) and the medium subset (answered incorrectly by 6–8 models). Across these questions, models relied on the provided text in 61% of cases but still chose an incorrect answer (e.g., selecting the wrong component). In the remaining 39%, models appeared to rely on internal knowledge and produced answers that do not occur in the text.

S1.9 A Multi-Hop QA Generation Example

Figure S1.9 illustrates a typical multi-hop QA example derived from our knowledge-graph-based methodology. The context is drawn from chemical literature discussing the use of *carbon dioxide* as a renewable feedstock for *formic acid*, which then serves as a non-gaseous CO surrogate in *carbonylation reactions*. By chaining these facts together, our approach constructs a question that requires integrating multiple pieces of information to arrive at the correct answer. This demonstrates how multi-hop reasoning, guided by entity relations and supplemented with descriptive metadata, enables the generation and evaluation of more complex questions by large language models. Additionally, Figure 2 shows the step-by-step process of deriving multi-hop questions from a knowledge graph, illustrating how entities, relations, and descriptive metadata are combined to construct more complex queries.

Context:

[Source 1*]: Carbonylation reactions constitute a potent tool to manufacture carboxylic acids and their derivatives both in industry and academic organic synthesis. In general, carbonylation requires the use of toxic carbon monoxide, which thus usually demands certified high-pressure reaction vessels. Therefore, developing non-gaseous CO surrogate for conducting safe and facile-operation carbonylation is an important and ongoing research topic. Among these established CO surrogates, formic acid is one

kind of versatile atom.

[Source 2*]: The utilization of carbon dioxide as a C1 feedstock for the generation of industrially relevant chemicals is also an interesting approach. CO_2 is an attractive renewable C1 source, which can lead to formic acid. Those approaches would not only reduce carbon dioxide emissions through carbon capture but also compensate sequestration costs by producing chemicals in global demand.

Question:
What is the process that uses a substance, produced from carbon dioxide and known as the simplest carboxylic acid with antibacterial and preservative properties, as a non-gaseous surrogate to safely form carboxylic acids and their derivatives under mild conditions?

Answer: carbonylation reactions

Sentence-level supporting facts:
1) formic acid can be produced from carbon dioxide.
2) formic acid is the simplest carboxylic acid with antibacterial and preservative properties.
3) formic acid can act as a non-gaseous CO surrogate.
4) carbonylation reactions safely produce carboxylic acids under mild conditions using formic acid as a CO surrogate.

Path (multi-hop chain of reasoning):
carbon dioxide → formic acid → carbonylation reactions

* Source 1 and source 2 are coming from different documents.

Context:

[Source 1*]: The most common way to functionalise two-dimensional materials such as graphene is through reactions occurring in solution.

[Source 2*]: Radiolytic shielding via graphene membranes has been demonstrated, highlighting the role of the two-dimensional allotrope graphene.

[Source 3*]: A variety of chemisorptionbased solid sorbents exist, such as amines grafted onto porous solids like silica, cellulose, or metalorganic frameworks (MOFs). Membrane-based DAC captures CO_2 by selectively allowing CO_2 to permeate membranes while excluding other gases like nitrogen.

$Cr_3(Cr_4Cl)_3(BTT)_{8,2}(BTT_3$, 1,3,5-benzenetristetrazolate$)_{12}$
MOF exhibits very high selectivity for O_2 over nitrogen. While multiple factors can influence gas adsorption in MOFs, the origin of very high selectivity for O2 is perplexing because structurally similar metal ion MOFs are unselective.

Question:
What is the metal–organic framework that exhibits very high oxygen selectivity over the major atmospheric diatomic gas, which is typically excluded by the selective barriers used for isolating carbon dioxide, and which act as radiolytic shields when formed from a two-dimensional carbon allotrope that is often functionalised in solution?

Answer:
$Cr_3(Cr_4Cl)_3(BTT)_{82}(BTT_3$ 1,3,5-benzenetristetrazolate$)_{12}$

Sentence-level supporting facts:
1) Solution-phase chemistry is the standard route to functionalise graphene.
2) Graphene can form membranes that provide radiolytic shielding.
3) Membranes used for DAC selectively exclude nitrogen.
4) $Cr_3(Cr_4Cl)_3(BTT)_{82}$ MOF shows very high O_2 selectivity over nitrogen.

Path (multi-hop chain of reasoning):
solution → graphene → membranes → nitrogen → $Cr_3(Cr_4Cl)_3(BTT)_{82}$

*Sources 1–4 are extracted from four different documents.

S1.10 Impact of Context and Reasoning on output tokens count

Figure S4 illustrates the impact of context availability on the average number of output tokens generated by reasoning and non-reasoning models when answering questions. Non-reasoning models produce a similar number of tokens, as they do not engage in test-time reasoning. In contrast, for reasoning models, the number of tokens generated to answer questions decreases with the availability of context, suggesting a potential requirement for less cognitive effort when the context is available.

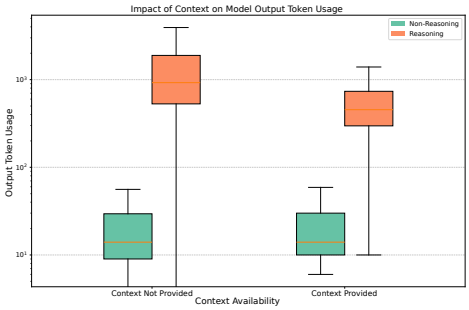

Figure S4: Visualization of the token usage distribution based on input context availability and model reasoning capability. The y-axis is log-scaled.

S1.11 Performance Analysis Based on Number of Hops

The following figures illustrate the details of each model's performance, latency, and tokens used for question clusters requiring a different number of hops to be answered.

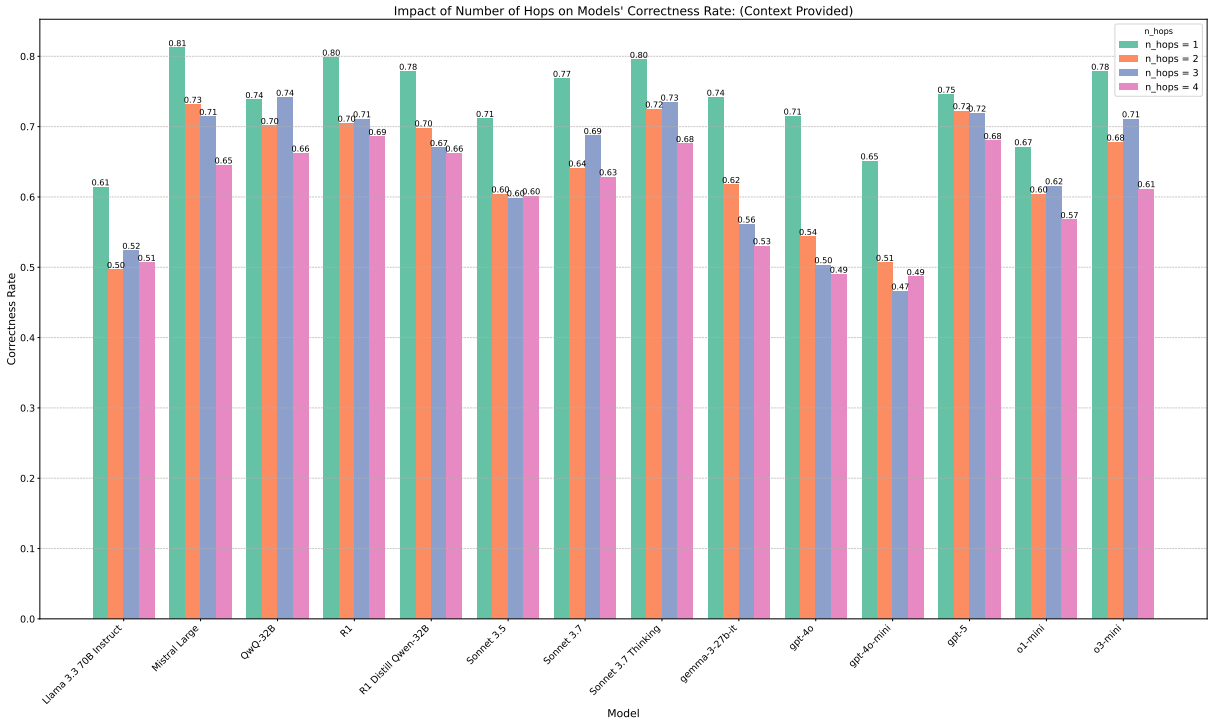

Figure S5: Overall performance of models as a function of the number of hops When context is provided.

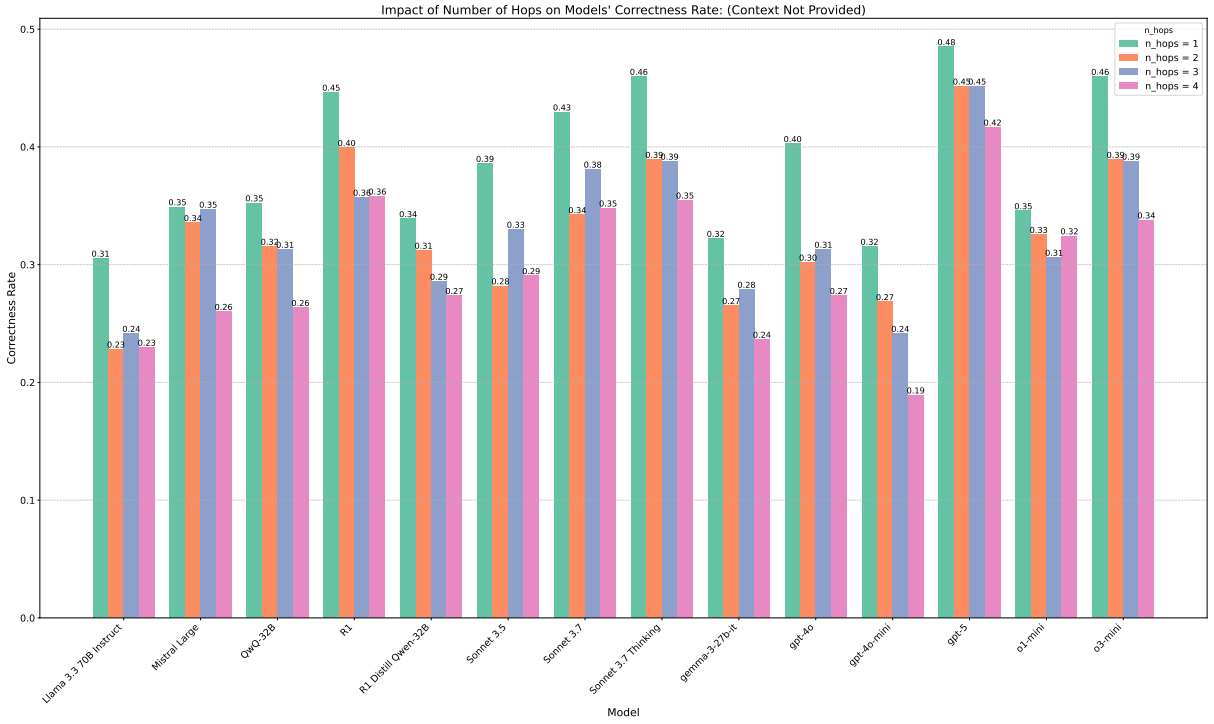

Figure S6: Overall performance of models as a function of the number of hops when context is not provided.

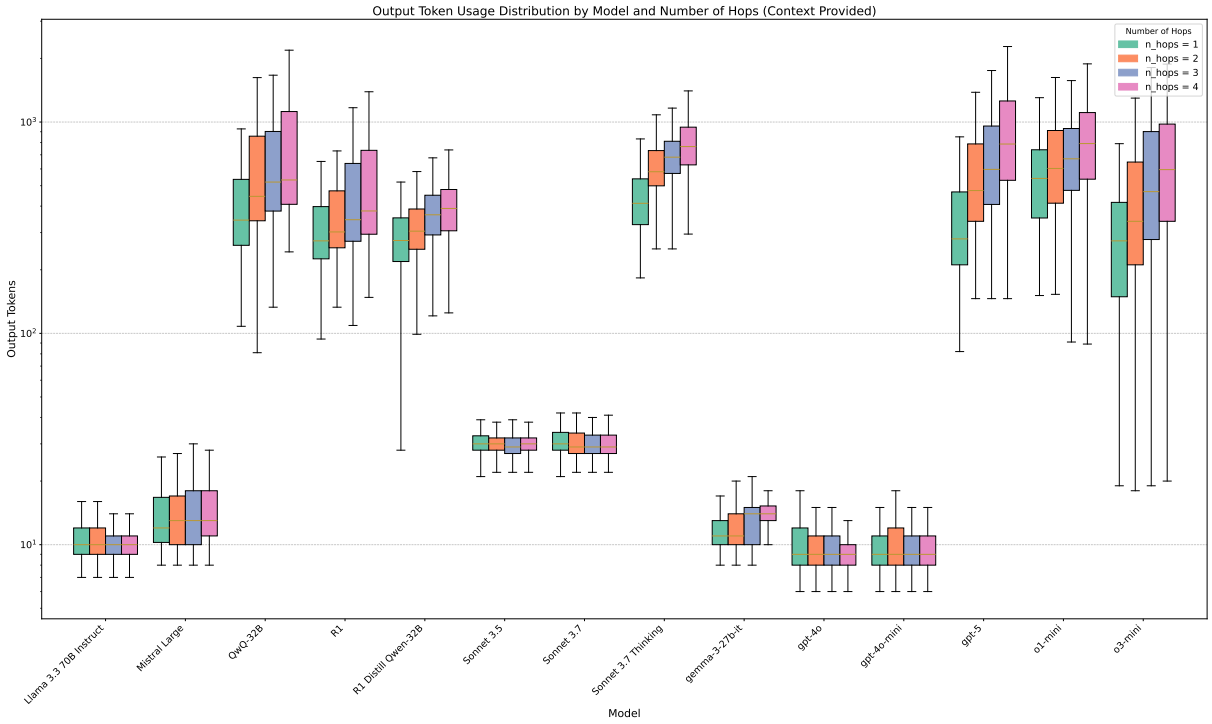

Figure S7: Token usage across different numbers of hops when context is provided.

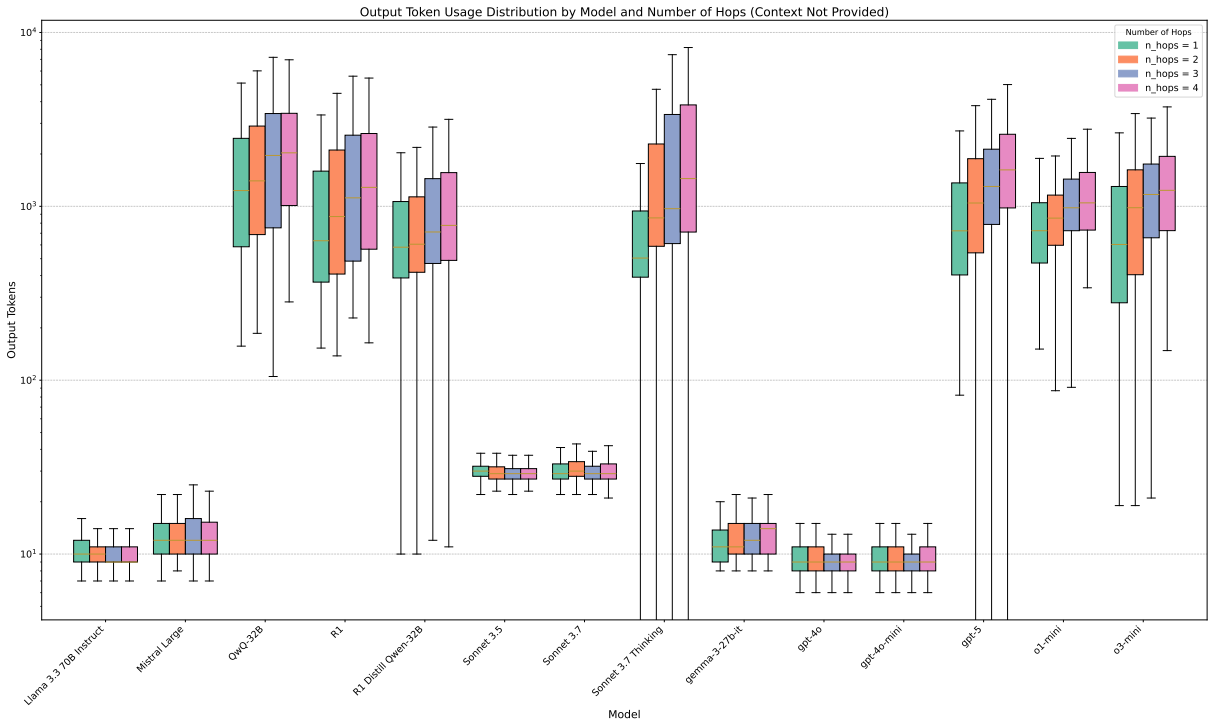

Figure S8: Token usage across different numbers of hops when context is not provided.

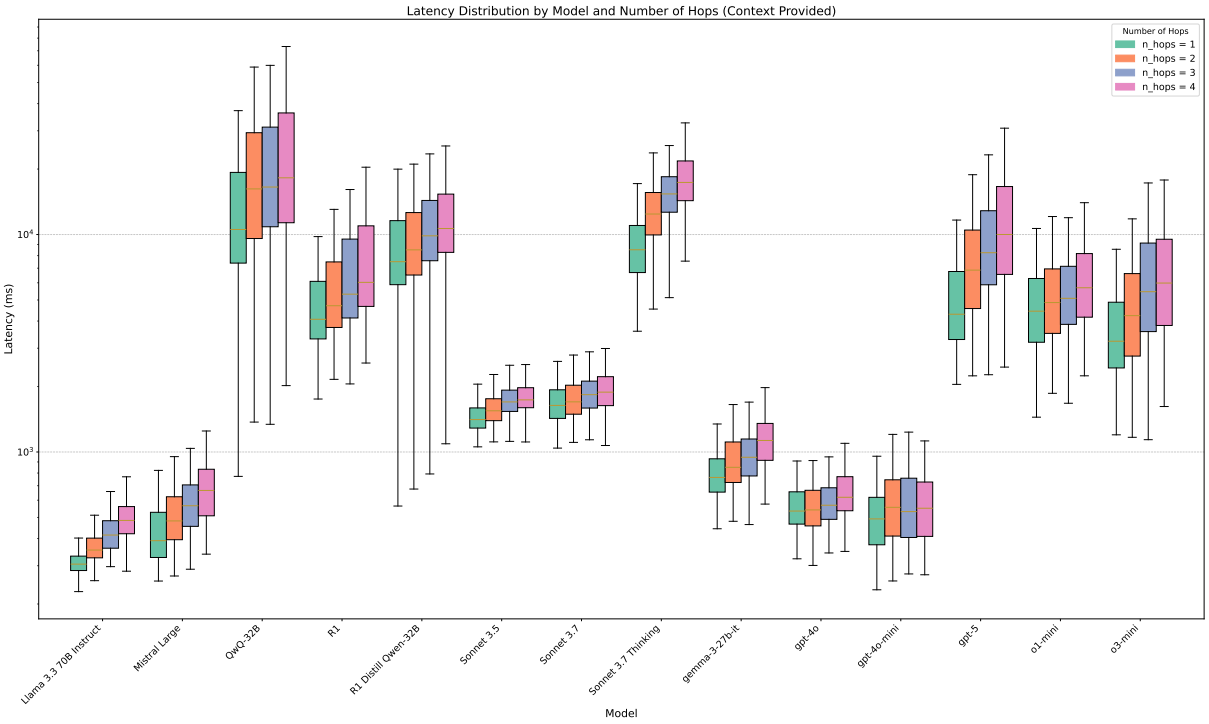

Figure S9: Latency measurements across different numbers of hops when context is provided.

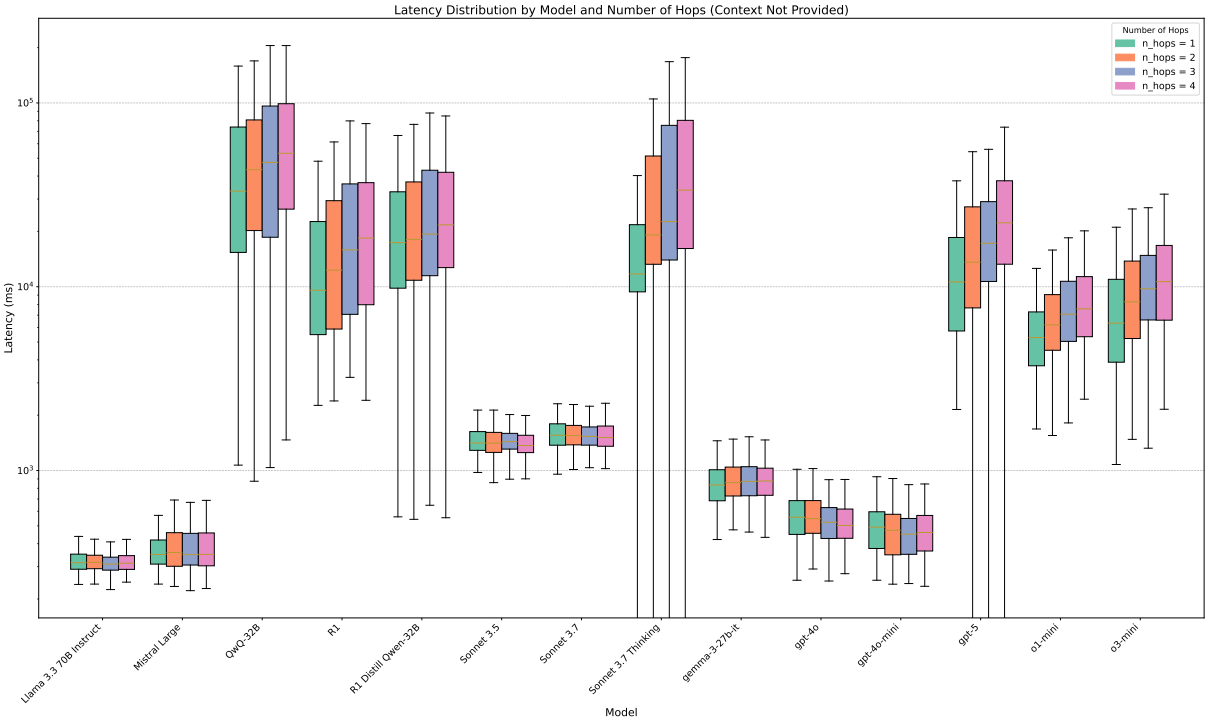

Figure S10: Latency measurements across different numbers of hops when context is not provided.