Benchmarking Retrieval-Augmented Generation for Chemistry

Xianrui Zhong[1], **Bowen Jin**[1], **Siru Ouyang**[1], **Yanzhen Shen**[1], **Qiao Jin**[2], **Yin Fang**[2], **Zhiyong Lu**[2] **& Jiawei Han**[1]
[1]Siebel School of Computing and Data Science, University of Illinois Urbana-Champaign
[2]National Library of Medicine, National Institutes of Health
{xzhong23, bowenj4, siruo2, yanzhen4, hanj}@illinois.edu
{qiao.jin, yin.fang, zhiyong.lu}@nih.gov

Abstract

Retrieval-augmented generation (RAG) has emerged as a powerful framework for enhancing large language models (LLMs) with external knowledge, particularly in scientific domains that demand specialized and dynamic information. Despite its promise, the application of RAG in the chemistry domain remains underexplored, primarily due to the lack of high-quality, domain-specific corpora and well-curated evaluation benchmarks. In this work, we introduce CHEMRAG-BENCH, a comprehensive benchmark designed to systematically assess the effectiveness of RAG across a diverse set of chemistry-related tasks. The curated chemistry corpus integrates heterogeneous knowledge sources, including scientific literature, the PubChem database, PubMed abstracts, textbooks, and Wikipedia entries. In addition, we present CHEMRAG-TOOLKIT, a modular and extensible RAG toolkit that supports five retrieval algorithms and eight LLMs. Using CHEMRAG-TOOLKIT, we demonstrate that RAG yields a substantial performance gain—achieving an average relative improvement of 17.4% over direct inference methods. We further conduct in-depth analyses on retriever architectures, corpus selection, and the number of retrieved passages, culminating in practical recommendations to guide future research and deployment of RAG systems in the chemistry domain. The code and data is available at https://chemrag.github.io.

1 Introduction

Retrieval-augmented generation (RAG) (Gao et al., 2023) has emerged as a powerful paradigm for enhancing large language models (LLMs) with external knowledge sources. By incorporating retrieval into the generation process, RAG can effectively mitigate hallucinations (Zhang et al., 2023b) and inject up-to-date domain-specific information into LLMs (Siriwardhana et al., 2023). These capabilities are particularly valuable in scientific domains, where factual accuracy and timely knowledge are critical. A typical scientific RAG system consists of two components: (1) a retriever that selects relevant documents or facts from a scientific knowledge base, and (2) a generator, often an LLM, that integrates the retrieved content to produce informed and coherent responses. Such frameworks have shown promising applications in domains like biomedicine (Xiong et al., 2024).

In chemistry, LLMs have shown remarkable potential across various tasks, including molecular captioning (Li et al., 2024), chemical reasoning (Tang et al., 2025), and reaction prediction (Shi et al., 2023). However, chemistry is a highly specialized and dynamic discipline, characterized by complex terminologies, domain-specific conventions, and rapidly-evolving knowledge. As a result, LLMs trained on general corpora often fail to generate grounded and accurate responses, instead producing hallucinated or outdated content (Zhang et al., 2023a; Wang et al., 2024b). RAG presents a natural solution to these limitations, allowing models to retrieve and incorporate trusted chemical knowledge during inference.

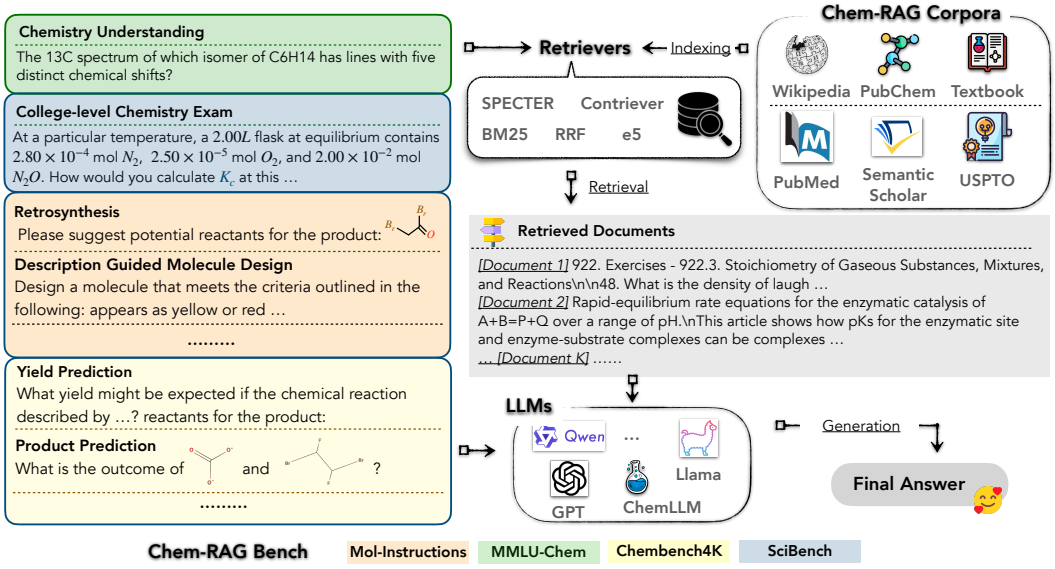

Figure 1: Overview of the CHEMRAG toolkit. Retrievers are first constructed from CHEMRAG corpora. For each question in the CHEMRAG-BENCH benchmark, we retrieve related documents as additional contexts for LLMs to predict the final answer.

Despite the growing interest in applying RAG to the chemistry domain, there remains a lack of standardized benchmarks and curated domain-specific resources to support rigorous evaluation and design of RAG systems. To address this gap, we introduce CHEMRAG-BENCH, a novel evaluation benchmark comprising 1,932 expert-curated question-answer pairs covering diverse chemistry tasks. These include description-guided molecular design, retrosynthesis, chemical calculations, molecule captioning, name conversion, and reaction prediction. This benchmark provides a foundation for systematically evaluating the effectiveness of RAG systems in chemistry and guiding future research in this direction.

To facilitate comprehensive and reproducible evaluation on CHEMRAG-BENCH, we introduce CHEMRAG-TOOLKIT, a user-friendly and extensible toolkit that supports 6 chemistry-related corpora, 5 retrieval methods, and 8 LLMs, encompassing both general-purpose and domain-specific LLMs. Based on the proposed CHEMRAG-BENCH benchmark, we conduct a systematic evaluation of various CHEMRAG solutions and analyze the impact of individual components on overall performance from multiple perspectives. Across a range of LLMs, we observe an average relative performance improvement of 17.4% when using CHEMRAG compared to direct inference without retrieval.

Along the retrieval corpus dimension, we find that different chemistry tasks exhibit distinct preferences for specific corpora. *For instance, molecule design and reaction prediction tasks benefit more from literature-derived corpora, while nomenclature and conversion tasks favor structured chemical databases.* These observations suggest that task-aware corpus selection is crucial for maximizing RAG performance. Moreover, we show that combining all available corpora often yields the most robust results, serving as a comprehensive retrieval base. In terms of the retriever component, Contriever (Izacard et al., 2021) demonstrates consistently strong performance across tasks. We further find that performance can be enhanced by leveraging ensemble retrieval strategies that combine the strengths of multiple retrievers. Beyond standard evaluation metrics, we uncover a log-linear scaling trend between the number of retrieved passages and downstream model performance, indicating that retrieval depth plays a key role in generation quality. Additionally, we investigate the proportion of external knowledge utilized per task and provide in-depth analysis on retriever selection for chemistry discovery scenarios. Finally, we distill a set of practical recommendations from our findings, offering actionable insights for deploying and advancing RAG systems in the chemistry domain. In summary, our key contributions are fourfold:

- We introduce CHEMRAG-BENCH, a comprehensive benchmark comprising 1,932 expert-curated question-answer pairs across six chemistry-related knowledge sources, enabling systematic evaluation of RAG methods in the chemistry domain.
- We curate CHEMRAG-CORPUS, a large-scale and comprehensive corpora construction.
- We develop CHEMRAG-TOOLKIT, an easy-to-use and extensible framework that integrates five retrieval algorithms and eight large language models, and demonstrate an average relative improvement of 17.4% when applying CHEMRAG over direct inference.
- We conduct comprehensive empirical analyses to examine the impact of retrieval corpus selection, retriever architecture, the number of retrieved documents, etc. Based on our findings, we provide practical guidelines to inform future research and the real-world deployment of chemistry-focused RAG systems.

2 Related Work

2.1 Retrieval-Augmented Generation

Retrieval-augmented generation (RAG) enhances large language models by incorporating external knowledge sources (Lewis et al., 2020). It has been shown to reduce hallucinations (Ayala & Bechard, 2024) and provide access to up-to-date information (Fan et al., 2024). Recent work has sought to improve RAG performance through various enhancements, including more effective retrieval mechanisms (Glass et al., 2022), iterative retrieval-and-reasoning pipelines (Trivedi et al., 2022; Jin et al., 2025), and the integration of long-context language models to better handle extended inputs (Jin et al., 2024). While substantial progress has been made in general-domain RAG benchmarks (Asai et al., 2023; Yu et al., 2024; Kwiatkowski et al., 2019; Yang et al., 2018), relatively little attention has been given to the scientific domain. Although recent efforts, such as Xiong et al. (2024), begin to explore this direction in the medicine domain, the application of RAG to the chemistry domain remains underdeveloped. Notably, TextReact (Qian et al., 2023) applies text retrieval to tasks like reaction condition recommendation and one-step retrosynthesis. ChemLit-QA (Wellawatte et al., 2024) introduces a dataset for chemistry-oriented RAG, but its questions are generated from isolated paper excerpts and may lack real-world utility. Importantly, there remains a gap in the availability of high-quality, domain-specific corpora and comprehensive RAG benchmarks tailored to chemistry.

2.2 Large Language Models for Chemistry

The rapid advancement of large language models (LLMs) has opened up new opportunities across various scientific domains (Ouyang et al., 2023), spurring the development of numerous benchmarks (Lu et al., 2022; Wang et al., 2023; Zhang et al., 2024). Among these domains, chemistry stands out as a particularly challenging yet promising area for LLM applications Fang et al. (2023). Recent efforts, such as ChemCrow (Bran et al., 2023), have demonstrated the potential of integrating LLMs with specialized tools to address a wide range of downstream tasks. In addition, LLMs have been employed to improve performance on specific chemistry tasks, including reaction prediction (Zhong et al., 2023; 2024), drug discovery (Edwards et al., 2023), and SMILES recognition (Edwards et al., 2021). Despite growing interest, existing benchmarks often fall short in capturing the unique demands of the chemistry domain, which is inherently knowledge-intensive. In contrast to general NLP tasks that frequently involve surface-level reasoning, chemistry requires precise retrieval and synthesis of complex, domain-specific knowledge. These characteristics make it a compelling testbed for RAG, where the incorporation of external knowledge sources can substantially enhance LLM reasoning and decision-making.

3 The CHEMRAG-BENCH Benchmark

3.1 Evaluation Settings

The primary goal of this work is to assess RAG systems in a setting that closely mirrors real-world information needs in the chemistry domain while remaining feasible and scalable

Dataset	Type	Task	Size	Avg. Length
MMLU-Chem	Multi-Choice	Chemistry Understanding	303	31
SciBench-Chem	Calculation	College-level Examination	229	94
ChemBench4K	Multi-Choice	Caption2Mol	100	72
		Mol2Caption	100	
		Name Conversion	100	
		Product Prediction	100	
		RetroSynthesis	100	
		Solvent Prediction	100	
		Temperature Prediction	100	
		Yield Prediction	100	
Mol-Instructions	Open-Ended	Desc.-guided Molecule Design	100	54
		Forward Reaction Prediction	100	
		Molecular Desc. Generation	100	
		Property Prediction	100	
		Reagent Prediction	100	
		RetroSynthesis	100	

Table 1: Statistics of CHEMRAG-BENCH, including question type, task type, data size, and the average length of each question.

in practice. To this end, the proposed CHEMRAG-BENCH benchmark is designed around four core evaluation scenarios:

- **Zero-Shot Learning:** In real application, demonstrations are hard to find when conducting novel chemistry discovery. Therefore, we do not use any demonstration when evaluating the RAG systems.
- **Open-ended Evaluation:** Most chemistry tasks are open-ended and do not have answer options, including description-guided molecule design, retrosynthesis, and reagent prediction. To better align with chemists' needs, the RAG system should be evaluated in an open-ended setting. In this setting, no answer options will be provided.
- **Multi-Choice Evaluation:** Multiple choice questions are common in LLM-related system evaluation. We adopt a multiple-choice setting to be consistent with previous work, and to make the evaluation more comprehensive. Many open-ended questions can be converted to multiple-choice questions by adding incorrect options.
- **Question-Only Retrieval:** To mimic real-world usage, for multiple-choice questions, only the question is used as the query for RAG.

3.2 Question Datasets

Our CHEMRAG-BENCH contains four datasets that cover a wide range of chemistry tasks, including three multi-choice benchmarks, MMLU-Chem (Hendrycks et al., 2021), SciBench (Wang et al., 2024c), and ChemBench4K (Zhang et al., 2024), and one open-ended benchmark, Mol-Instructions (Fang et al., 2024). MMLU-Chem consists of college chemistry questions collected online. SciBench collects questions from chemistry textbooks. ChemBench4K contains multiple chemical analysis and prediction tasks, but in a multiple-choice fashion. Mol-Instructions is a collection of molecule design, retrosynthesis, and prediction tasks. The statistics of the datasets are shown in Table 1.

Metric For multi-choice questions, we use accuracy as the metric. For open-ended questions, the generated molecule is evaluated by exact match (EM), validity, MACCS FTS, RDK FTS, Morgan FTS, and BLEU. To evaluate the generated text, we use BLEU and ROUGE. For numerical results, we use accuracy with a 5% relative error tolerance. Please refer to Appendix A for more details on molecule evaluation metrics.

4 The CHEMRAG-TOOLKIT

CHEMRAG-TOOLKIT analyzes how RAG systems perform on CHEMRAG-BENCH. The CHEMRAG-TOOLKIT contains three major components: Corpora, Retrievers, and LLMs.

Corpora By working with researchers in chemistry, we construct a large and comprehensive collection of corpora covering a wide range of chemistry information from six sources: [1]PubChem for molecule information (English name, SMILES, IUPAC name, weight, mocular formula, and synonyms), [2]PubMed for biochemistry abstracts, [3]USPTO for chemical patents information, [4]Semantic Scholar for chemistry full-text papers, and [5]OpenStax for chemistry textbooks. Data processing is described in Appendix C, and the statistics of the corpora are shown in Table 2.

Corpus	# Snippets	Avg. Length	Domain
PubChem	14.6M	72	Chemistry
PubMed	23.9M	305	Biochemistry
USPTO	143K	140	Chemistry
Semantic Scholar	32.7M	403	Chemistry
OpenStax	5521	273	Chemistry
Wikipedia	29.9M	163	General

Table 2: Statistics of corpora in CHEMRAG-TOOLKIT.

Retrievers In CHEMRAG-TOOLKIT, we select four representative retrievers for the retrieval process in RAG: BM25 (Robertson & Zaragoza, 2009), Contriever (Izacard et al., 2022), SPECTER (Cohan et al., 2020), and e5 (Wang et al., 2024a). In addition, we implement Reciprocal Rank Fusion (RRF, Cormack et al. (2009)) to combine the results from different retrievers.

LLMs We choose a few representative LLMs to be used in CHEMRAG-TOOLKIT: Llama-3.1-8B-Instruct, Llama-3.1-70B-Instruct, and Mistral-7B-Instruct-v0.2 for general open-source models, ChemLLM for chemistry open-source model, GPT-3.5-turbo and GPT-4o for closed-source models, Deepseek-R1-Llama-8B and o1 for reasoning models.

5 Experiment Result

5.1 Comparison of Backbone LLMs

To systematically study how LLMs perform on chemistry tasks and how the proposed CHEMRAG-TOOLKIT affects models, we benchmark various LLMs on CHEMRAG-BENCH with the same ChemRAG-Corpora. The top 5 documents retrieved by the RRF retriever are prepended to each question. The results are in Table 3, Appendix E, and Appendix F. More implementation details could be found in Appendix B.

As shown in Table 3, different models behave differently when CHEMRAG-TOOLKIT is in use. On average, most models benefit from using CHEMRAG-TOOLKIT, Llama-3.1-8B-Instruct gains 25.86%, Llama-3.1-70B-Instruct gains 24.5%, Mistral-7B-Instruct gains 36.9%, GPT-3.5-turbo gains 28.43%, GPT-4o gains 20.92%, and o1 gains 16.38%. The largest improvement often comes from the one in Mol-Instructions and ChemBench4K. Among the backbone LLMs, o1 achieves the highest performance in both baseline and RAG settings.

Although most models benefit from CHEMRAG-TOOLKIT, the performance of ChemLLM decreases slightly (-12.6%) and Deepseek-R1-Llama barely improves. They still gain some performance on certain question datasets. Both ChemLLM and DeepSeek-R1-Llama benefit from RAG on MMLU-Chem ($+14.91\%$ and $+3.59\%$). DeepSeek-R1-Llama also performs slightly better on SciBench and Mol-Instructions with the proposed toolkit ($+0.78$ and $+4.07$). In our experiments, we notice that DeepSeek-R1-Llama-8B does not follow our instructions and generates its answers in various forms, which poses difficulty in parsing its answers and may lead to poor performance in calculation.

We observe that larger models have consistent gains in chemistry-specific benchmarks (SciBench-Chem, ChemBench4K, and Mol-Instructions). This suggests that larger models

LLM	Method	MMLU	SciBench	ChemBench4K	Mol-Instruct.	Avg.
Llama3.1 (8b)	Baseline	42.90	3.30	27.25	23.99	24.36
	Ours	52.15	3.56	25.88	41.05	30.66
Llama3.1 (70b)	Baseline	62.38	5.99	24.25	28.33	30.24
	Ours	61.05	13.63	26.25	49.67	37.65
Mistral (7b)	Baseline	45.21	2.09	12.63	4.66	16.15
	Ours	42.57	0	11.13	34.73	22.11
ChemLLM (7b)	Baseline	37.62	8.72	23.5	17.74	21.90
	Ours	43.23	2.03	16.75	14.56	19.14
Deepseek-r1 -llama(8b)	Baseline	55.44	3.09	35.38	3.75	24.42
	Ours	57.43	3.87	29.13	7.82	24.56
GPT3.5	Baseline	49.17	9.66	30.5	29.00	29.58
	Ours	52.81	8.80	44.5	45.83	37.99
GPT-4o	Baseline	74.59	4.97	59.5	28.79	41.96
	Ours	73.92	8.59	67.25	53.18	50.74
o1	Baseline	85.81	40.82	41.63	31.55	49.95
	Ours	85.48	43.61	58.38	45.04	58.13

Table 3: Benchmark results of different LLMs on CHEMRAG-BENCH.

have a better understanding of the retrieved documents. In MMLU-Chem, most large models (Llama-3.1-70b, GPT-4o, and o1) do not benefit from our toolkit. This may be because MMLU is a common benchmark when evaluating LLMs, and these models are trained on related knowledge. The toolkit may not be able to bring new knowledge to larger models. In SciBench-Chem, many models suffer from using the toolkit, this reflects that these models may not understand the retrieved documents well, since advanced models (Llama-3.1-70b, GPT-4o, and o1) all benefit from the toolkit, and o1 even reaches the highest performance when using the toolkit. In ChemBench4K, similar patterns occur: smaller models have worse results, but larger models gain from the toolkit.

Since Mol-Instructions contains multiple sub-tasks, and each sub-tasks require multiple metrics, we select description-guided molecule design as a representative to analyze in detail how models perform after using our toolkit. The comparison is shown in Figure 2, with more details in the appendix. From Figure 2, we observe that with our toolkit, all models improve in all aspects, except ChemLLM.

5.2 Comparison of Retrievers and Corpora

To understand the effect of each component in CHEMRAG-TOOLKIT, we benchmark different retrievers with different corpora on CHEMRAG-BENCH. The experiments are conducted with GPT-3.5-turbo since it is one of the models that benefit most from our toolkit, and it is also efficient and inexpensive for inference. The results are in Table 4.

Comparison between Corpora From Table 4, we observe that the performance of a RAG system is correlated to the selected corpus. The model performs the best with OpenStax on MMLU-Chem and SciBench-Chem, but OpenStax barely has benefit for Mol-Instructions. USPTO helps the model to achieve its best on ChemBench4K and Mol-Instructions, but it provides little benefit on MMLU-Chem and SciBench-Chem. When using the combined CHEMRAG Corpus, the model achieves the best on MMLU-Chem and ChemBench4K, surpassing leveraging only one corpus, which demonstrates the significance of combining multiple corpora. The CHEMRAG Corpus also helps the model to perform better on Mol-Instructions, only not as good as USPTO. Our corpus is also beneficial for SciBench when using Contriever as the retriever.

Comparison between Retrievers The Retriever plays another critical role as it decides how the documents rank. From our experiments shown in Table 4, all retrievers have their best

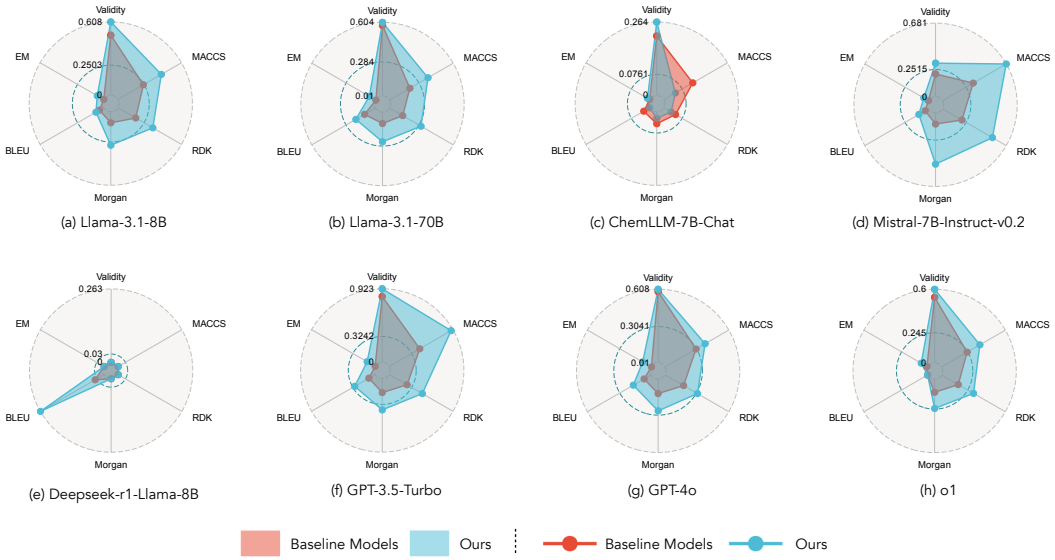

Figure 2: Performance comparison on description-guided molecule design w.r.t evaluation metrics for molecule generation. Ours outperforms the baseline in almost all the scenarios.

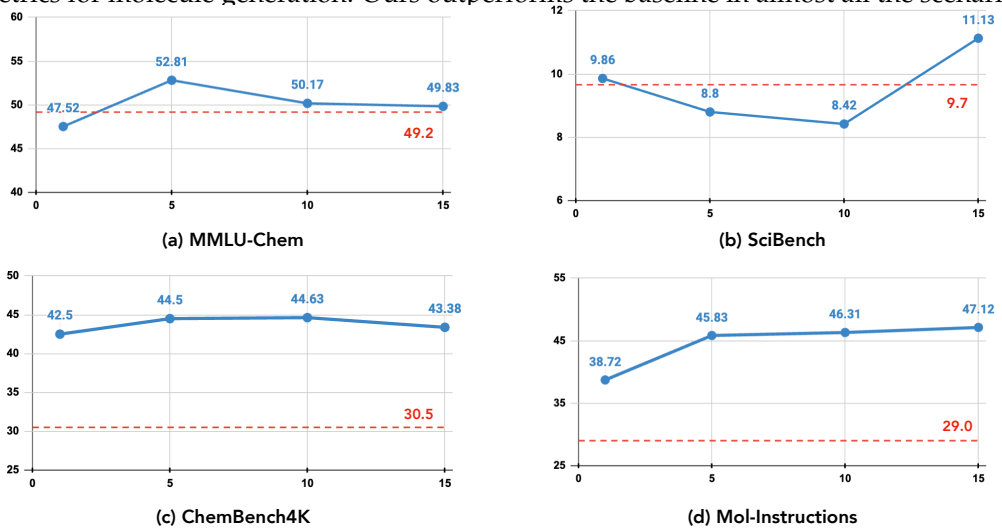

Figure 3: Performance comparison on different numbers of retrieved documents. The red dotted line represents the baseline. The experiments are conducted on GPT-3.5-turbo.

performance on a specific corpus and task. BM25 shows a very strong performance when using USPTO on Mol-Instructions, and using PubChem on SciBench-Chem. Contriever outperforms other retrievers when incorporating CHEMRAG Corpus on MMLU-Chem, it also works well with PubChem and the CHEMRAG Corpus. SPECTER and e5 have mixed performances but still can excel in certain corpora. The RRF retriever, combining the results of the four retrievers, usually improves the performance, even though it might not be the best, and sometimes results in the best performance. For instance, RRF helps the model achieve the best on MMLU-Chem and ChemBench4K.

6 Discussion and Analyses

6.1 Performance Scaling

The number of retrieved documents k is an important factor in RAG systems. When k is too small, RAG systems may lack critical information; on the other hand, when k is too large,

Corpus	Retriever	MMLU	SciBench	Chem-Bench4K	Mol-Instructions	Avg.
None	*None*	49.17	9.66	30.5	29.00	29.58
PubChem	BM25	47.19	12.98	36.00	27.73	30.98
	Contriever	48.18	10.02	39.50	29.72	31.86
	SPECTER	49.83	9.98	36.75	26.80	30.84
	e5	46.86	8.61	40.50	30.65	31.66
	RRF	48.84	9.08	37.38	29.58	31.22
PubMed	BM25	46.86	12.02	38.63	28.14	31.41
	Contriever	49.17	10.37	37.13	27.51	31.05
	SPECTER	47.19	9.08	36.63	27.69	30.15
	e5	46.53	10.36	39.63	25.07	30.40
	RRF	48.18	8.98	37.13	25.70	30.00
USPTO	BM25	49.50	11.55	44.00	56.17	40.31
	Contriever	49.50	10.92	42.25	37.40	35.02
	SPECTER	47.85	9.44	37.00	31.71	29.00
	e5	47.52	11.05	38.13	37.55	33.56
	RRF	49.17	10.70	43.38	56.68	39.98
Semantic Scholar	BM25	45.54	7.18	37.25	29.72	29.92
	Contriever	47.85	12.65	38.88	31.73	32.78
	SPECTER	49.17	10.45	37.00	26.44	30.77
	e5	45.21	10.78	38.75	31.52	31.57
	RRF	44.55	8.91	39.13	31.76	31.09
OpenStax	BM25	50.17	10.04	37.88	28.34	31.61
	Contriever	49.50	12.66	36.38	27.95	31.62
	SPECTER	50.50	11.88	37.13	28.20	31.93
	e5	49.83	11.57	38.5	29.96	32.47
	RRF	52.48	11.55	37.25	29.35	32.66
Wiki	BM25	49.17	8.06	38.75	27.67	30.91
	Contriever	48.84	10.33	37.25	29.14	31.39
	SPECTER	47.52	9.21	39.25	27.22	30.8
	e5	50.83	8.93	37.13	29.54	31.61
	RRF	50.17	10.70	38.00	27.66	31.63
Chem -RAG Corpus	BM25	49.83	6.51	38.13	34.99	32.37
	Contriever	53.46	12.58	42.63	42.08	37.69
	SPECTER	48.18	8.57	41.63	32.35	32.69
	e5	47.19	7.56	37.13	42.24	33.53
	RRF	52.81	8.80	44.5	45.83	37.99

Table 4: Experiment results of various retrievers and corpora on CHEMRAG-BENCH. Compared with the baseline (first row), the intensity of the shade represents the magnitude of the decreases and increases.

RAG systems may suffer from too much irrelevant information. To better understand how this factor affects RAG systems, we conduct experiments on $k = 1, 5, 10, 15$. The results are shown in Figure 3. The phenomenon where performance first increases and then decreases as k increases is clearly observed in MMLU-Chem and ChemBench4K. In SciBench-Chem, the performance first decreases but then increases. This suggests that a better retriever is needed or a reranker should be used. In our opinion, a better retriever should be developed since current retrievers only consider semantic similarity, however, semantic similarity may not be sufficient in reasoning tasks like SciBench. Overall, $k = 5$ is a good choice since it provides sufficient information in most cases.

6.2 Proportion in the CHEMRAG Corpus

We investigate the proportion of different sources used across various tasks. Figure 4 shows the proportions of six sources in CHEMRAG-Corpus, and the actual proportions in the top 50 retrieved chunks in CHEMRAG-BENCH. A task-specific pattern of proportion is observed.

OpenStax has a larger proportion in SciBench and a relatively large proportion in MMLU-Chem. This is natural since the questions in SciBench and MMLU are derived from academic settings. PubChem has the largest proportion in both ChemBench4K and Mol-Instructions, which can be explained by the fact that these two tasks focus on molecule-related questions.

6.3 Retrievers in Chemistry

In our observation from Table 4 and Figure 3, we believe that a better retriever is needed for retrieving documents for chemistry downstream tasks. In Table 4, the model always performs better with USPTO and OpenStax corpora, but it performs worse on the combined corpus, which suggests the retriever ranks the helpful snippets to a lower place. This is also validated by the sudden rise in Figure 3 (b).

In addition, chemistry retrieval faces a "multi-modality" issue. One chemical compound may have multiple representations, including SMILES strings, IUPAC names, and English names, and each of them has variants.

Finally, current retrievers only consider keyword matching and semantic similarities, but chemistry tasks require more. More discussion and analysis are in Appendix D.

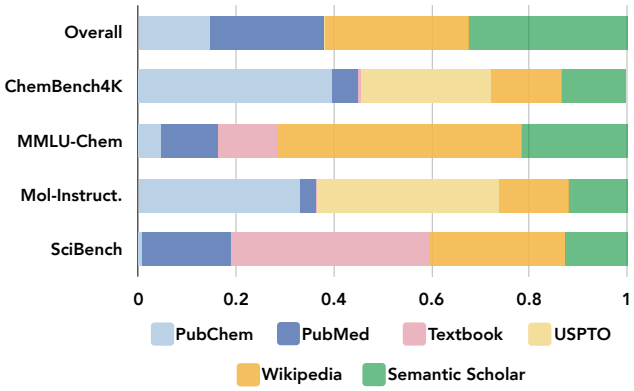

Figure 4: The overall corpus composition of CHEMRAG corpora and the actually retrieved proportion in different tasks.

6.4 Practical Recommendations

Based on our experiments, we provide some practical recommendations:

- **Corpus Selection** The proposed CHEMRAG-Corpus is a good start and is likely to outperform using only one corpus source. This is confirmed in Table 4, MMLU-Chem and ChemBench4K in particular. When working on molecule-related tasks, one may want to try USPTO since it reaches high performance in both ChemBench4K and Mol-Instructions. As for questions in school, OpenStax (textbook) may be preferred, but the performance is still lower than using CHEMRAG-Corpus in MMLU-Chem, illustrated in Table 4.
- **Retriever Selection** Contriever is the most stable retriever in the four individual retrievers, but its performance still fluctuates across tasks and corpora. The proposed RRF retriever is recommended since it usually performs close to the best individual retriever and sometimes outperforms them.
- **LLM Selection** o1 is the best model for all the tasks. Considering the cost and inference speed, GPT-3.5-turbo and GPT-4o are good options. For open-source models, Llama-3.1-8B-Instruct is preferred since it achieves the second among the five open-source models and performs similar to the best model, Llama-3.1-70B-Instruct. Llama-3.1-70B only performs 24% better, but with 775% more parameters and much higher computation cost.

7 Conclusion

We propose CHEMRAG-BENCH and CHEMRAG-TOOLKIT to systematically evaluate RAG systems in chemistry. Based on our extensive experiments, we provide some novel findings, practical recommendations, and future directions for the community to better leverage RAG systems in chemistry in real world.

Acknowledgments

Research was supported in part by US DARPA INCAS Program No. HR0011-21-C0165 and BRIES Program No. HR0011-24-3-0325, National Science Foundation IIS-19-56151, the Molecule Maker Lab Institute: An AI Research Institutes program supported by NSF under Award No. 2019897, the Institute for Geospatial Understanding through an Integrative Discovery Environment (I-GUIDE) by NSF under Award No. 2118329, and Apple PhD Fellowship. This research was also supported in part by the Division of Intramural Research (DIR), National Library of Medicine (NLM), National Institutes of Health (NIH). This research has used the Delta/DeltaAI advanced computing and data resource, a joint effort of the University of Illinois Urbana-Champaign and its National Center for Supercomputing Applications, supported by the National Science Foundation (award OAC 2320345) and the State of Illinois. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and do not necessarily represent the views, either expressed or implied, of DARPA or the U.S. Government.

References

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection, 2023. URL https://arxiv.org/abs/2310.11511.

Orlando Ayala and Patrice Bechard. Reducing hallucination in structured outputs via retrieval-augmented generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pp. 228–238. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-industry.19. URL http://dx.doi.org/10.18653/v1/2024.naacl-industry.19.

Andres M Bran, Sam Cox, Andrew D White, and Philippe Schwaller. Chemcrow: Augmenting large-language models with chemistry tools. *arXiv preprint arXiv:2304.05376*, 2023.

Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. SPECTER: Document-level Representation Learning using Citation-informed Transformers. In *ACL*, 2020.

Gordon V. Cormack, Charles L A Clarke, and Stefan Buettcher. Reciprocal rank fusion outperforms condorcet and individual rank learning methods. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pp. 758–759, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605584836. doi: 10.1145/1571941.1572114. URL https://doi.org/10.1145/1571941.1572114.

Joseph L. Durant, Burton A. Leland, Douglas R. Henry, and James G. Nourse. Reoptimization of MDL keys for use in drug discovery. *J. Chem. Inf. Comput. Sci.*, 42(5):1273–1280, 2002. doi: 10.1021/CI010132R. URL https://doi.org/10.1021/ci010132r.

Carl Edwards, ChengXiang Zhai, and Heng Ji. Text2Mol: Cross-modal molecule retrieval with natural language queries. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 595–607, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.47. URL https://aclanthology.org/2021.emnlp-main.47.

Carl N Edwards, Aakanksha Naik, Tushar Khot, Martin D Burke, Heng Ji, and Tom Hope. Synergpt: In-context learning for personalized drug synergy prediction and drug design. *bioRxiv*, pp. 2023–07, 2023.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. A survey on rag meeting llms: Towards retrieval-augmented large language

models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pp. 6491–6501, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400704901. doi: 10.1145/3637528.3671470. URL https://doi.org/10.1145/3637528.3671470.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. *arXiv preprint arXiv:2306.08018*, 2023.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. In *ICLR*. OpenReview.net, 2024. URL https://openreview.net/pdf?id=Tlsdsb6l9n.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2, 2023.

Michael Glass, Gaetano Rossiello, Md Faisal Mahbub Chowdhury, Ankita Rajaram Naik, Pengshan Cai, and Alfio Gliozzo. Re2g: Retrieve, rerank, generate. *arXiv preprint arXiv:2207.06300*, 2022.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021. URL https://arxiv.org/abs/2009.03300.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *arXiv preprint arXiv:2112.09118*, 2021.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning, 2022. URL https://arxiv.org/abs/2112.09118.

Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan O Arik. Long-context llms meet rag: Overcoming challenges for long inputs in rag. In *The Thirteenth International Conference on Learning Representations*, 2024.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Greg Landrum et al. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum*, 8:31, 2013.

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.

Jiatong Li, Yunqing Liu, Wenqi Fan, Xiao-Yong Wei, Hui Liu, Jiliang Tang, and Qing Li. Empowering molecule discovery for molecule-caption translation with large language models: A chatgpt perspective. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Yujian Li and Bi Liu. A normalized levenshtein distance metric. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):1091–1095, 2007. doi: 10.1109/TPAMI.2007.1078. URL https://doi.org/10.1109/TPAMI.2007.1078.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

Siru Ouyang, Shuohang Wang, Yang Liu, Ming Zhong, Yizhu Jiao, Dan Iter, Reid Pryzant, Chenguang Zhu, Heng Ji, and Jiawei Han. The shifted and the overlooked: A task-oriented investigation of user-GPT interactions. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://openreview.net/forum?id=qS1ip2dGH0.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*, pp. 311–318. ACL, 2002. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

Yujie Qian, Zhening Li, Zhengkai Tu, Connor W. Coley, and Regina Barzilay. Predictive chemistry augmented with text retrieval, 2023. URL https://arxiv.org/abs/2312.04881.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009. ISSN 1554-0669. doi: 10.1561/1500000019. URL http://dx.doi.org/10.1561/1500000019.

Nadine Schneider, Roger A. Sayle, and Gregory A. Landrum. Get your atoms in order - an open-source implementation of a novel and robust molecular canonicalization algorithm. *J. Chem. Inf. Model.*, 55(10):2111–2120, 2015. doi: 10.1021/ACS.JCIM.5B00543. URL https://doi.org/10.1021/acs.jcim.5b00543.

Yaorui Shi, An Zhang, Enzhi Zhang, Zhiyuan Liu, and Xiang Wang. Relm: Leveraging language models for enhanced chemical reaction prediction. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pp. 5506–5520. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-EMNLP.366. URL https://doi.org/10.18653/v1/2023.findings-emnlp.366.

Shamane Siriwardhana, Rivindu Weerasekera, Elliott Wen, Tharindu Kaluarachchi, Rajib Rana, and Suranga Nanayakkara. Improving the domain adaptation of retrieval augmented generation (rag) models for open domain question answering. *Transactions of the Association for Computational Linguistics*, 11:1–17, 2023.

Xiangru Tang, Tianyu Hu, Muyang Ye, Yanjun Shao, Xunjian Yin, Siru Ouyang, Wangchunshu Zhou, Pan Lu, Zhuosheng Zhang, Yilun Zhao, Arman Cohan, and Mark Gerstein. Chemagent: Self-updating memories in large language models improves chemical reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=kuhIqeVg0e.

Taffee T Tanimoto. Elementary mathematical theory of classification and prediction. 1958.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. *arXiv preprint arXiv:2212.10509*, 2022.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training, 2024a. URL https://arxiv.org/abs/2212.03533.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. Knowledge editing for large language models: A survey. *ACM Computing Surveys*, 57(3):1–37, 2024b.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. *arXiv preprint arXiv:2307.10635*, 2023.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R. Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. SciBench: Evaluating College-Level Scientific Problem-Solving Abilities of Large Language Models. In *Proceedings of the Forty-First International Conference on Machine Learning*, 2024c.

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*, 2024. URL https://openreview.net/forum?id=6PoHVQeeHU.

Guangzhi Xiong, Qiao Jin, Zhiyong Lu, and Aidong Zhang. Benchmarking retrieval-augmented generation for medicine. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 6233–6251, 2024.

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

Tian Yu, Shaolei Zhang, and Yang Feng. Auto-rag: Autonomous retrieval-augmented generation for large language models, 2024. URL https://arxiv.org/abs/2411.19443.

Di Zhang, Wei Liu, Qian Tan, Jingdan Chen, Hang Yan, Yuliang Yan, Jiatong Li, Weiran Huang, Xiangyu Yue, Dongzhan Zhou, Shufei Zhang, Mao Su, Hansen Zhong, Yuqiang Li, and Wanli Ouyang. Chemllm: A chemical large language model, 2024.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. Siren's song in the AI ocean: A survey on hallucination in large language models. *CoRR*, abs/2309.01219, 2023a. doi: 10.48550/ARXIV.2309.01219. URL https://doi.org/10.48550/arXiv.2309.01219.

Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023b.

Ming Zhong, Siru Ouyang, Minhao Jiang, Vivian Hu, Yizhu Jiao, Xuan Wang, and Jiawei Han. ReactIE: Enhancing chemical reaction extraction with weak supervision. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 12120–12130, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.767. URL https://aclanthology.org/2023.findings-acl.767.

Xianrui Zhong, Yufeng Du, Siru Ouyang, Ming Zhong, Tingfeng Luo, Qirong Ho, Hao Peng, Heng Ji, and Jiawei Han. Actionie: Action extraction from scientific literature with programming languages. In *the Association for Computational Linguistics: ACL 2024*. Association for Computational Linguistics, August 2024.

A Evaluation Metrics for Molecules

To assess the quality of generated molecules, we first employ general text-based generation metrics such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004), which compare generated outputs against reference answers.

For molecular generation, we begin by verifying the validity of generated molecules using RDKit (Landrum et al., 2013) and then compute their exact match with reference solutions. However, a single textual description can correspond to multiple molecular structures, making exact matching a limited evaluation criterion. Moreover, expecting an LLM—even one fine-tuned with LoRA on specific instructions—to consistently generate outputs that perfectly match reference molecules is often unrealistic.

To address these challenges and provide a more comprehensive evaluation, we incorporate molecular similarity metrics, including similarity scores based on RDKit, MACCS, and Morgan fingerprints (Tanimoto, 1958; Schneider et al., 2015; Durant et al., 2002), alongside Levenshtein (Li & Liu, 2007) and BLEU scores.

For tasks that need to compare numbers, following previous work (Wang et al., 2024c), we compare the generated output with the ground truth, allowing a 5% relative error. This makes sure that the score is within 0 and 1, making it more suitable for combining with other scores. These results can be found in Appendix F.

B Implementation Details

Since DeepSeek-R1-Llama and o1 are reasoning models, following their guidelines, we set 0.6 and 1 as their temperatures respectively. The temperatures for other models are set to 0 for reproducibility. For each experiment, we run three rounds and report the mean for DeepSeek-R1-Llama and o1 models. We run one round for other models. The number of max generation tokens is set to 10,000 for DeepSeek-R1-Llama and o1 since their reasoning requires more tokens, and the numbers for other models are set to 512.

C Corpus Data Processing

PubChem For each chemical compound in PubChem, we collect its names, properties, and description and transform them into JSON format.

Semantic Scholar We collect 1,849,956 full-text chemistry papers from Semantic Scholar, then chunk them into chunks of 512 tokens with a 50-token overlap.

USPTO USPTO contains a large amount of information about reactions: reactants, reagents, products, and yields. We transform these attributes into JSON format.

OpenStax We parse the PDF textbooks to texts with Mathpix[1], and then split the parsed textbook into chunks of 512 tokens with a 50-token overlap.

PubMed and Wikipedia The data is from Xiong et al. (2024).

D Retrieval Error Case Study

We present two representative errors in chemistry retrieval observed in our experiments.

D.1 Prioritize too much on molecule matching

For instance, the query "Which ingredients are commonly selected for creating Cc1oc(-c2ccccc2)nc1COc1ccc2cc(CC3SC(=O)NC3=O)cnc2c1?" is asking about reagents information

[1] https://mathpix.com/

Model Size	Method	MMLU	SciBench	ChemBench4K	Mol-Instruct.	Avg.
8B	Baseline	42.90	3.30	27.25	23.99	24.36
	CoT	61.38	1.92	32.75	6.67	25.68
	Ours	52.15	3.56	25.88	41.05	30.66
70B	Baseline	62.38	5.99	24.25	28.33	30.24
	CoT	67.66	4.36	32.13	25.97	32.53
	Ours	61.05	13.63	26.25	49.67	37.65

Table 5: Comparison with Chain-of-Thought (CoT) on Llama-3.1.

for generating the mentioned compound as a product. When there is no such information, retrievers usually give high scores to irrelevant documents that contain the same SMILES. This may introduce noise to the retrieved documents and mislead LLMs. A better retrieval system may identify this situation, then search for the synonyms, and search for similar compounds if synonyms still fail.

D.2 Often fail when the document only mention one name

A molecule may have many names, including SMILES, IUPAC, and English names. This makes retrieving the right document more difficult as the question may only contain English names, but there may only be SMILES in the relevant documents. For example,

Query: What is the molecular weight of aspirin?

Document1: The molecular weight of CH3COOC6H4COOH is 180.16 g/mol.

Document2: Aspirin can cause developmental toxicity.

In this example, CH3COOC6H4COOH is the formula of aspirin, but current retrievers don't give Document1 high scores because it doesn't know CH3COOC6H4COOH is aspirin. By training with synonyms, the texts with CH3COOC6H4COOH and aspirin will be closer in the embedding space. Alternatively, when using an LLM to generate queries, the LLM may first find the formula of aspirin, and then use the formula to search for the molecular weight.

E More Experiment Results

E.1 Chain-of-Thought Comparison

We deploy Chain-of-Thought (CoT) on Llama3.1-8B and Llama3.1-70B, the results are in Table 5. CoT does better than baseline and solely using RAG in multiple-choice settings (MMLU and ChemBench4K), but performs worse than baseline and solely using RAG in open QA (SciBench and Mol-Instructions).

F Detailed Experiment Results

F.1 SciBench-Chemistry

Table 6 shows the performances of different models on SciBench-Chemistry tasks.

F.2 ChemBench4K

Table 7 and Table 8 demonstrate the performances of different models on ChemBench4K tasks.

F.3 Mol-Instructions

Table 9, Table 10, Table 11, and Table 12 demonstrate some results in Mol-Instructions.

LLM	Method	SciBench				Avg.
		atkins	chemmec	matter	quan	
Llama3.1 (8b)	Baseline	0	10.26	0	2.94	3.30
	Ours	3.74	2.56	2.04	5.88	3.56
Llama3.1 (70b)	Baseline	2.56	5.99	0	17.65	5.99
	Ours	6.54	15.38	6.12	26.47	13.63
Mistral (7b)	Baseline	3.74	2.56	2.04	0	2.09
	Ours	0	0	0	0	0
ChemLLM (7b)	Baseline	11.21	12.82	2.04	8.82	8.72
	Ours	0.93	5.13	2.04	0	2.03
Deepseek-r1 -llama(8b)	Baseline	4.67	7.69	0	0	3.09
	Ours	2.80	7.69	2.04	2.94	3.87
GPT3.5	Baseline	5.61	23.08	4.08	5.88	9.66
	Ours	5.61	20.51	6.12	2.94	8.80
GPT-4o	Baseline	3.74	10.26	0	5.88	4.97
	Ours	10.28	10.26	2.04	11.76	8.59
o1	Baseline	38.32	46.15	34.69	44.12	40.82
	Ours	44.86	48.72	36.73	44.12	43.61

Table 6: Detailed benchmark results of different LLMs on SciBench-Chemistry. The accuracy is computed by comparing the generated answer with the ground truth, allowing a 5% relative error.

LLM	Method	ChemBench4K			
		Caption2Mol	Mol2Caption	Name Conversion	Product Prediction
Llama3.1 (8b)	Baseline	0	88	57	13
	Ours	7	70	59	15
Llama3.1 (70b)	Baseline	3	86	68	0
	Ours	5	87	64	11
Mistral (7b)	Baseline	3	26	40	15
	Ours	7	19	33	11
ChemLLM (7b)	Baseline	24	46	48	2
	Ours	21	33	38	0
Deepseek-r1 -llama(8b)	Baseline	15	81	70	19
	Ours	18	68	65	12
GPT3.5	Baseline	20	89	48	17
	Ours	36	87	60	39
GPT-4o	Baseline	41	98	79	93
	Ours	61	98	81	83
o1	Baseline	6	99	76	26
	Ours	27	96	80	59

Table 7: Detailed benchmark results of different LLMs on ChemBench4K, Part 1.

G Prompt

The prompts used in our experiments can be found in Table 13, 14, 15, 16, 17, 18, 19, 20.

LLM	Method	ChemBench4K			
		Retrosynthesis	Solvent Prediction	Temp. Prediction	Yield Prediction
Llama3.1 (8b)	Baseline	0	21	17	22
	Ours	6	20	15	15
Llama3.1 (70b)	Baseline	0	24	9	4
	Ours	2	22	1	18
Mistral (7b)	Baseline	0	2	9	6
	Ours	2	8	0	9
ChemLLM (7b)	Baseline	1	25	21	21
	Ours	0	28	4	10
Deepseek-r1 -llama(8b)	Baseline	14	36	31	17
	Ours	3	33	17	17
GPT3.5	Baseline	4	23	18	25
	Ours	26	41	28	39
GPT-4o	Baseline	54	35	33	43
	Ours	76	49	43	47
o1	Baseline	5	42	48	31
	Ours	50	50	63	42

Table 8: Detailed benchmark results of different LLMs on ChemBench4K, Part 2.

LLM	Method	Description-Guided Molecule Deisgn					
		EM↑	Validity↑	MACCS FTS↑	RDK FTS↑	Morgan FTS↑	BLEU↑
Llama3.1 (8b)	Baseline	0	73	35.71	25.01	13.37	6.02
	Ours	9	89	60.78	48.85	40.64	10.92
Llama3.1 (70b)	Baseline	1	95	32.61	21.67	16.72	18.34
	Ours	11	99	60.35	49.65	40.89	31.56
Mistral (7b)	Baseline	0	21	32.74	20.66	10.34	3.61
	Ours	5	31	68.14	53.26	47.52	10.35
ChemLLM (7b)	Baseline	0	47	26.40	10.70	9.41	5.41
	Ours	2	58	10.48	6.59	5.10	0
Deepseek-r1 -llama(8b)	Baseline	0	0	0	0	0	3.70
	Ours	0	0	0	0	0	26.25
GPT3.5	Baseline	0	85	45.53	26.48	18.08	9.41
	Ours	12	95	92.28	49.35	40.45	30.56
GPT-4o	Baseline	1	93	47.33	28.88	20.32	11.88
	Ours	14	96	60.84	49.47	42.45	27.92
o1	Baseline	1	89	40.12	25.72	17.55	-
	Ours	12	97	57.59	46.01	39.78	-

Table 9: Detailed benchmark results of different LLMs on Mol-Instructions – Description-guided molecule design.

LLM	Method	Forward Reaction Prediction					
		EM↑	Validity↑	MACCS FTS↑	RDK FTS↑	Morgan FTS↑	BLEU↑
Llama3.1 (8b)	Baseline	0	38	59	60.24	43.19	6.63
	Ours	17	92	63.29	53.61	45.46	29.50
Llama3.1 (70b)	Baseline	0	68	63.74	60.68	44.94	17.47
	Ours	22	91	72.70	62.74	57.14	43.89
Mistral (7b)	Baseline	0	0	0	0	0	2
	Ours	3	29	76.68	77.54	63.33	11.31
ChemLLM (7b)	Baseline	0	29	45.88	34.61	28.15	3.18
	Ours	0	57	22.55	17.55	12.68	0
Deepseek-r1 -llama(8b)	Baseline	0	33	1.56	0.39	0.65	12.98
	Ours	0	59	0	0	0	1.77
GPT3.5	Baseline	0	57	58.37	52.03	40.63	23.43
	Ours	16	96	72.11	67.80	56.21	39.43
GPT-4o	Baseline	2	96	66.35	62.6	50.84	50.7
	Ours	26	89	78.31	73.88	68.3	61.44
o1	Baseline	13	87	81.95	78.95	72.06	-
	Ours	30	90	87.35	82.89	78.92	-

Table 10: Detailed benchmark results of different LLMs on Mol-Instruction – Forward Reaction Prediction.

LLM	Method	Molecule Description Generation	
		BLEU	Rouge-L
Llama3.1 (8b)	Baseline	0	8.98
	Ours	8.24	32.79
Llama3.1 (70b)	Baseline	0.83	15.25
	Ours	4.30	27.6
Mistral (7b)	Baseline	0.63	18.64
	Ours	4.48	32.09
ChemLLM (7b)	Baseline	5.33	34.04
	Ours	0	0
Deepseek-r1 -llama(8b)	Baseline	0	0
	Ours	0	0
GPT3.5	Baseline	3.18	20.58
	Ours	3.75	21.51
GPT-4o	Baseline	1.23	18.25
	Ours	2.98	30.06
o1	Baseline	0	0
	Ours	1.02	14.44

Table 11: Detailed benchmark results of different LLMs on Mol-Instructions – Moleclue Description Generation.

LLM	Method	Property Prediction
		Accuracy
Llama3.1 (8b)	Baseline	0.14
	Ours	0
Llama3.1 (70b)	Baseline	0
	Ours	0
Mistral (7b)	Baseline	0
	Ours	0
ChemLLM (7b)	Baseline	60
	Ours	15
Deepseek-r1 -llama(8b)	Baseline	1
	Ours	1
GPT3.5	Baseline	18
	Ours	1
GPT-4o	Baseline	2
	Ours	0
o1	Baseline	0
	Ours	0

Table 12: Detailed benchmark results of different LLMs on Mol-Instructions – Property Prediction. The accuracy is computed by comparing the generated answer with the ground truth, allowing a 5% relative error.

Table 13: Baseline prompt template for general open-ended questions.

Open-ended Baseline Prompt

Answer the question directly.
Only give me the answer and do not output any other words.
Question: { *Instruction* }
Answer:

Table 14: Multi-choice baseline prompt template for general open-ended questions.

Multi-choice Baseline Prompt

Answer the question directly.
Only give me the answer and do not output any other words.
Question: { *Instruction* }
Choices: { *Choices* }
Make prediction from the given choices.
Answer:

Table 15: Numerical baseline prompt template for general open-ended questions.

Numerical Baseline Prompt

Answer the question directly.
Conclude the answer by stating "The answer is therefore [ANSWER]"
Only give me the answer and do not output any other words.
Question: { *Instruction* }
Answer:

Table 16: Generation baseline prompt template for general open-ended questions.

Generation Baseline Prompt

Answer the question directly.
Your answer should be surrounded by [ANSWER] and [/ANSWER]. When generating a molecule, please generate a valid SMILES string.
Only give me the answer and do not output any other words.
Question: { *Instruction* }
Answer:

Table 17: RAG prompt template for general open-ended questions.

Open-ended RAG Prompt

Answer the question based on the given document.
Only give me the answer and do not output any other words.
The following are given documents.
{ reference }
Question: { *Instruction* }
Answer:

Table 18: RAG Prompt template for multiple-choice questions.

Multi-choice RAG Prompt

Answer the question based on the given document.
Only give me the answer and do not output any other words.
The following are given documents.
{ reference }
Question: { *Instruction* }
Choices: { *Choices* }
Make prediction from the given choices.
Answer:

Table 19: Prompt template for numerical questions.

Numerical RAG Prompt

Answer the question based on the given document.
Conclude the answer by stating "The answer is therefore [ANSWER]"
Only give me the answer and do not output any other words.
The following are given documents.
{ reference }
Question: { *Instruction* }
Answer:

Table 20: Prompt template for generation questions.

Generation RAG Prompt

Answer the question based on the given document.
Your answer should be surrounded by [ANSWER] and [/ANSWER]. When generating a molecule, please generate a valid SMILES string.
The following are given documents.
{ reference }
Only give me the answer and do not output any other words.
Question: { *Instruction* }
Answer: