Restructuring the Corpus Makes RAG Work for Math

Negar Arabzadeh[1] **Wenjie Ma**[1] **Sewon Min**[1] **Matei Zaharia**[1]

[1]UC Berkeley
{negara, windsey, sewonm, matei}@berkeley.edu

Abstract

Large Language Models (LLMs) achieve strong performance on mathematical problem solving when guided by chain-of-thought prompting or trained on reasoning traces. Yet it remains unclear whether Retrieval-Augmented Generation (RAG) which shows a lot of success on knowledge-intensive tasks, can also provide benefits for math reasoning. We show that with regular text datastores, vanilla RAG provides no or little benefit on benchmarks such as MATH and AIME. However, it is possible to redesign datastore contents to be more RAG-friendly, and we examine which types of content and organizational structures most effectively support mathematical reasoning. We run experiments on different corpora building from generic text to structured "thinking traces" and explore how offline restructuring can transform raw material into reasoning-friendly retrieval units. Results show that restructuring documents into step-by-step reasoning units consistently boosts accuracy, with average gains of 17.7% and 8.8% for general-purpose models such as `LLaMA-3.1-8B` and `Qwen-2.5-32B`. Notably, even math-finetuned models benefit from structured external reasoning traces: `Mathstral-7B-v0.1` improves by 30.3%, while `OpenMath2-LLaMA-3.1-8B` gains 15.7%. These findings highlight the central role of corpus design: retrieval supports math reasoning only when paired with well-structured, reasoning-oriented data.

1 Introduction

Large Language Models (LLMs) are increasingly applied to reasoning intensive tasks, such as mathematical problem solving, theorem proving, complex scientific QA and so on Wang et al. (2025, 2024); Patel et al. (2025); Lála et al. (2023); Auer et al. (2023). Researchers have deployed a variety of prompt design and inference strategies to coax models into multi-step reasoning instead of directly outputting the answer Wei et al. (2022); Wang et al. (2022); Kojima et al. (2022). Some recent advances go beyond prompting by training or finetuning models on reasoning traces to further boost correctness Yang et al. (2024); Ho et al. (2023); Magister et al. (2023). Nevertheless, much of the improvement is believed to come from internalizing knowledge that is, the model's parameters encode facts, lemmas, patterns, and reasoning heuristics derived from the training data Roberts et al. (2020); Kaplan et al. (2020); Hoffmann et al. (2022). Thus, a strong model "knows" many theorem facts and solution templates implicitly, and uses those to guide its chain-of-thought.

Because of this, the community typically views success in reasoning-heavy domains as evidence of the model's internal reasoning and knowledge capacity Schaeffer et al. (2023). In contrast, while Retrieval-Augmented Generation (RAG) has been widely used in knowledge-intensive NLP tasks (e.g. open-domain QA, summarization) Lewis et al. (2020); Fan et al. (2024), the potential of RAG to assist with mathematical reasoning is far less explored. A few works apply RAG to math QA or tutoring settings and observe that retrieval can enhance answer quality when well integrated, though grounding overly rigidly to textbooks may reduce fluency or flexibility in explanations Levonian et al. (2023); Han et al. (2024). Still, the question remains: *Can retrieval actually help LLMs with math*

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: MATH-AI.

problem solving? Under what conditions, and with what kinds of retrieved content, can RAG improve reasoning, especially for models that lack strong implicit reasoning skills?

To answer this, in this work, we investigate the following three Research Questions (RQ)s:

- **RQ1: Does retrieval help math problem solving?** We hypothesize that retrieving helpful knowledge (e.g. similar examples, useful lemmas) might provide hints for solving problems.
- **RQ2: What kind of content is helpful for RAG on math problems?** We compare generic corpus of textbooks and web articles against collections that are only build from the "thinking trajectory" of other LLMs on decontaminated math problems. Our hypothesis is that a retrieving reasoning examples might provide guidance that aligns with the model's own reasoning steps.
- **RQ3: Can we restructure a corpus to make it more RAG-friendly for math?** Math documents often interleave text, formulas, and noisy context that may not directly support reasoning. Retrieved chunks are sometimes truncated, incomplete, or too short to convey full solutions. We therefore ask whether offline processing such as extracting concise reasoning steps can reorganize raw documents into retrieval units that better support RAG for mathematical problem solving.

We conduct experiments on the MATH benchmark and AIME competition problems (2022–2024). We evaluate the performance of different LLMs under different settings: with vs. without retrieval, across datastore types, and under various corpus reconstruction strategies. Preliminary results show that *retrieval can improve math problem solving, but the content of the datastore is crucial.* Simply retrieving from a large generic corpus often yields little or no benefit, depending on the LLM. In contrast, when the datastore contains high-quality restructured reasoning traces, models can use these cues to reach correct solutions. Reformatting retrieved context into concise, step-by-step hints further boosts effectiveness, especially for smaller models. This suggests that reasoning knowledge from larger models can indeed help weaker models when provided as external context. More broadly, making the retrieval corpus "RAG-friendly", both in relevance and format, is key to unlocking retrieval benefits for mathematical reasoning.

2 Methodology

Let $q \in \mathcal{Q}$ be a math problem, and let an LLM L produce a solution y. A retriever R takes a query q and a corpus $\mathcal{C}$ and returns k documents ranked by similarity $D = R(q; \mathcal{C}, k) = \{d_1, \ldots, d_k\}$. In this work, we study the following four variants in the *source* and *form* of the retrieved context:

(1) No-RAG. No external retrieval is used. Therefore: $y \sim L\big(q\big)$.

(2) Vanilla-RAG-Text. A text corpus $\mathcal{C}_{\text{text}}$ (e.g., textbooks, web math) is indexed. The top-k passages are retrieved and concatenated with the query, denoted $[\cdot \oplus \cdot]$:

$$D_{\text{text}} = R(q; \mathcal{C}_{\text{text}}, k), \qquad y \sim L\big(D_{\text{text}} \oplus q\big).$$

(3) Vanilla-RAG-Trace. We construct a *thinking-trace* corpus from an auxiliary problem set Q', chosen to be related in topic to $\mathcal{Q}$ and ideally much larger ($|Q'| \gg |\mathcal{Q}|$). An auxiliary "thinker" model L' generates a reasoning trace $\tau(q')$ for each $q' \in Q'$. The trace corpus is $\mathcal{C}_\tau = \big\{\, \tau(q') : q' \in Q' \,\big\}$. At test time we retrieve from $\mathcal{C}_\tau$ and condition the LLM on the retrieved traces:

$$D_\tau = R(q; \mathcal{C}_\tau, k), \qquad y \sim L\big(D_\tau \oplus q\big).$$

(4) RAG-Restruct. We apply a *restructuring* operator $\mathcal{F} : (d) \mapsto \tilde{d}$ that converts raw retrieved content into a canonical, math-friendly scaffold (e.g., numbered steps, lemma, application, symbol-normalized equations, truncation repair). Using either $\mathcal{C}_\tau$ or $\mathcal{C}_{\text{text}}$ as the source corpus, we obtain:

$$D = R(q; \mathcal{C}, k), \qquad \tilde{D} = \{\mathcal{F}(d_i) | d_i \in D\}, \qquad y \sim L\big(\tilde{D} \oplus q\big).$$

We note that since the restructuring function $\mathcal{F}$ is independent of the query and depends only on the corpus chunks, it can be applied entirely offline. Thus, it is possible to incur a one-time cost by using a larger model to restructure the corpus.

Table 1: Accuracy of different models on MATH under various RAG and no-RAG settings. Best values in each column are in bold. Best value for each model is shown in blue.

Method	LLaMA-3.1-8B				Qwen2.5-32B			
	CompactDS MATH	CompactDS Full	Traces Qwen2-32B	Traces Gemini-2.0	CompactDS MATH	CompactDS Full	Traces Qwen3-2B	Traces Gemini-2.0
No-RAG	45.9	45.9	45.9	45.9	73.6	73.6	73.6	73.6
Vanilla-RAG	47.6	44.3	**47.6**	48.1	76.2	75.6	74.4	74.7
RAG-Restruct-8B	**54.4**	48.0	44.0	**54.5**	**77.5**	**78.0**	**74.8**	79.8
RAG-Restruct-32B	51.8	**48.3**	46.2	53.5	76.1	77.2	74.4	**80.4**

Method	OpenMath2-Llama3.1-8B				Mathstral-7B-v0.1			
	CompactDS MATH	CompactDS Full	Traces Qwen32B	Traces Gemini	CompactDS MATH	CompactDS Full	Traces Qwen32B	Traces Gemini
No-RAG	63.0	63.0	**63.0**	63.0	47.5	47.5	**47.5**	47.5
Vanilla-RAG	**64.2**	61.4	61.4	61.0	43.3	42.3	45.6	45.0
RAG-Restruct-8B	62.8	64.5	61.3	**73.5**	**51.7**	52.1	43.8	**62.7**
RAG-Restruct-32B	64.1	**65.2**	62.6	72.2	50.9	**53.0**	47.4	61.1

3 Experimental setup

Datasets and Evaluation: Following Lyu et al. (2025), we evaluate on MATH Hendrycks et al. (2021) and AIME 2022-2024 using zero-shot chain-of-thought (CoT) with the same prompt and decoding settings across both datasets. Evaluation details and prompts are provided in Appendix B

Models: We evaluate four models to study RAG effects across general-purpose vs. math-specialized LLMs: `Llama3.1-8B-Instruct` and `Qwen2.5-32B-Instruct` as general models, alongside two math-tuned variants, `Mathstral-7B-v0.1`, specialized for mathematical and scientific tasks, and `OpenMath2-Llama3.1-8B`, obtained by finetuning `Llama-3.1-8B-Base` on OpenMathInstruct-2 and reported to markedly improve MATH accuracy over the vanilla `Llama 3.1-8B-Instruct` Toshniwal et al. (2024). All models are run with a decoding temperature of 0.6 and a maximum generation length of 32K tokens.

Corpora. We evaluate RAG using four corpora: two large-scale text collections (CompactDS–Math and CompactDS–Full) and two reasoning-trace collections (S1–Traces-Gemini and S1–Traces-Qwen3-32B). The text corpora provide raw math-related content from web and academic sources, while the trace corpora consist of worked-out step-by-step solutions generated by large models. The S1–Traces-Gemini traces are taken from the SimpleScaling dataset Muennighoff et al. (2025), while the S1–Traces-Qwen3-32B traces are generated by us on the same decontaminated MATH portion using the released S1 prompt. A detailed description of each corpus is provided in Appendix C.

Retrieval. For all corpora indexed locally, we use FAISS Johnson et al. (2019) with Contriever embeddings Izacard et al. (2022), 256-token chunking, and using top-3 retrieved chunks via a FLAT exact index. For the CompactDS-Full corpus we query via their provided API.

Corpus Restructuring. We compare restructuring via a smaller model (`Qwen3-8B`) with a larger one (`Qwen3-32B`), in order to assess the impact of restructure model size on downstream performance. The restructuring prompt is inspired by the solution CoT prompt shown in Figure 3.

4 Results

In this section, we report results on the MATH problems; due to space constraints, results on AIME 2022–2024 are presented in Appendix A. Table 1 summarizes Average@4 accuracy across four models under different retrieval settings, and Figure 1 shows relative improvements compared to the No-RAG baseline. Across all settings, we find that **RAG on raw text alone provides no significant benefit over the No-RAG baseline**. For example, `LLaMA-3.1-8B` improves only marginally, from 45.9 without retrieval to 47.6 with CompactDS-Math, while `Mathstral-7B-v0.1` even drops from 47.5 (No-RAG) to 43.3 (Vanilla-RAG-Text) with CompactDS-Full. In general, simply appending raw text passages does not consistently improve performance. We further observe that **increasing corpus size or diversity does not guarantee better results**. CompactDS-Full, despite being much larger and more diverse than CompactDS-Math, performs comparably or worse across models. For instance, `Qwen2.5-32B` improves only slightly, from 73.6 (No-RAG) to 76.2(Vanilla-RAG-Text) with CompactDS-Math and 75.6 with CompactDS-Full.

Comparing Vanilla-RAG-Text and Vanilla-RAG-Trace, we find that retrieval from thinking traces does not provide a significant advantage over raw text corpora. The key improvements only emerge once a

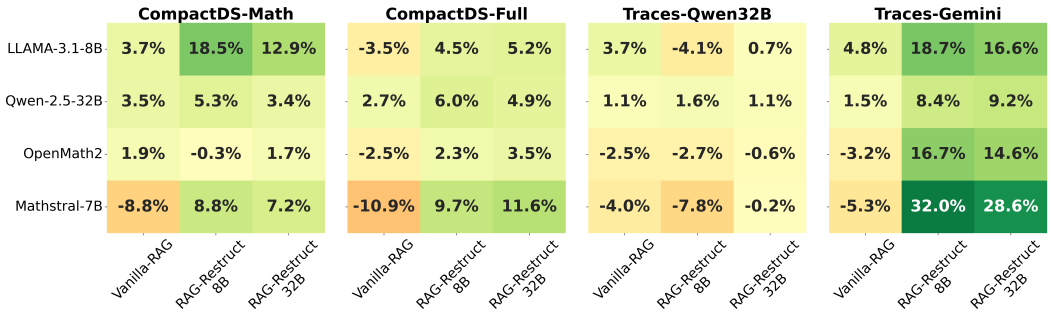

Figure 1: Relative improvements (%) across different RAG strategies and corpora w.r.t NO-RAG baselines. Each heatmap corresponds to a different retrieval corpus and each row represents a different model. Within each heatmap, columns correspond to improvements from Vanilla-RAG, RAG-Restruct with `Qwen3-8B`, and RAG-Restruct with `Qwen3-32B` (from left to right). As shown, *restructuring retrieved items from traces* produced by larger models such as `Gemini` consistently improves performance for both general-purpose and math-specialized models.

restructuring step is applied. Looking at the `Restruct.` rows, we observe that **offline restructuring into step-by-step reasoning units yields consistent and substantial gains**. For example, with `LLaMA-3.1-8B`, restructuring retrieved context from `CompactDS-Math` boosts accuracy from 45.9 to 54.4 (+18.5%), and to 54.5 with restructured `Gemini` traces (+18.7%). `Qwen2.5-32B` improves from 73.6 to 79.8 (+8.4%) when restructured by Qwen3-8B, and to 80.4 (+9.2%) when restructured by Qwen3-32B. More interestingly, math-specialized models also benefit considerably. `OpenMath2` rises from 63.0 to 73.5(+16.7%) with restructured `Gemini` traces, while `Mathstral-7B-v0.1` jumps from 47.5 to 62.7 (+32%; the largest relative improvement among all models) . Restructuring quality also appears largely **robust to the size of the restructuring model**. Differences between `RAG-Restruct-8B` by `Qwen3-8B` vs `RAG-Restruct-32B` by `Qwen3-32B` are minor across corpora. However, **the source of the traces matters substantially**. Restructured `Gemini` traces (RAG-Trace-Gemini) consistently outperform `CompactDS-Math` (RAG-Text), while as shown in Figure 1 `RAG-Trace-Qwen32B` shows no or little advantage over No-RAG. This highlights that the choice of the "thinker" model strongly influences downstream RAG effectiveness.

In summary, retrieval can support math problem solving, but only when paired with right format and right content. Effective corpus restructuring is the key to unlocking RAG benefits, even for models already fine-tuned on math. Results on AIME (Appendix A) suggest that extending these benefits to more challenging benchmarks remains an open direction.

5 Takeaway and Ongoing Work

We investigate under what conditions RAG can be helpful for solving mathematical problems. In response to **RQ1**, we showed that simple vanilla RAG does not necessarily benefit math problem solving. However, when we curate a *RAG-friendly* datastore, LLMs can indeed benefit from restructured retrieved content. In response to RQ2 we showed that retrieval by itself, whether over raw text or unprocessed traces, provides little to no benefit, and in some cases even hurts performance. *Retrieval becomes useful only when combined with proper corpus preparation and restructuring.* In response to **RQ3**, we demonstrated that restructuring the retrieved content is essential for RAG to be effective in math. A simple offline restructuring of the corpus creates a RAG-friendly datastore, where content is optimized and prepared for the generator. Offline restructuring into step-by-step reasoning units consistently improves accuracy across all models. Notably, even math-specialized models benefit substantially from this restructuring. Restructured thinking traces prove especially effective, outperforming raw text and establishing themselves as the most beneficial retrieval resource.

This work represents an initial step toward designing RAG-friendly datastores. Our experiments are limited in scope, relying on a small set of models and benchmarks. Extending the analysis to larger reasoning-intensive models, additional datastores, and broader task suites will be important for generalization. Moreover, we explored only one method of restructuring. Future research should investigate more generalizable restructuring strategies, adaptive reconstruction methods and richer trace formats that extend to more complex reasoning benchmarks. Our results show that it is feasible to restructure the same corpus into a math-friendly format, but further exploration is required to fully realize the potential of RAG for reasoning-intensive tasks.

References

Sören Auer, Dante AC Barone, Cassiano Bartz, Eduardo G Cortes, Mohamad Yaser Jaradeh, Oliver Karras, Manolis Koubarakis, Dmitry Mouromtsev, Dmitrii Pliukhin, Daniil Radyush, et al. 2023. The sciqa scientific question answering benchmark for scholarly knowledge. *Scientific Reports* 13, 1 (2023), 7240.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv e-prints* (2024), arXiv–2407.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. 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*. 6491–6501.

Yuling Gu, Oyvind Tafjord, Bailey Kuehl, Dany Haddad, Jesse Dodge, and Hannaneh Hajishirzi. 2025. OLMES: A Standard for Language Model Evaluations. arXiv:2406.08446 [cs.CL] `https://arxiv.org/abs/2406.08446`

Zifei FeiFei Han, Jionghao Lin, Ashish Gurung, Danielle R Thomas, Eason Chen, Conrad Borchers, Shivang Gupta, and Kenneth R Koedinger. 2024. Improving assessment of tutoring practices using retrieval-augmented generation. *arXiv preprint arXiv:2402.14594* (2024).

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. *CoRR* abs/2103.03874 (2021). arXiv:2103.03874 `https://arxiv.org/abs/2103.03874`

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2023. Large Language Models Are Reasoning Teachers. arXiv:2212.10071 [cs.CL] `https://arxiv.org/abs/2212.10071`

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556* (2022).

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised Dense Information Retrieval with Contrastive Learning. arXiv:2112.09118 [cs.IR] `https://arxiv.org/abs/2112.09118`

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data* 7, 3 (2019), 535–547.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling Laws for Neural Language Models. arXiv:2001.08361 [cs.LG] `https://arxiv.org/abs/2001.08361`

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems* 35 (2022), 22199–22213.

Jakub Lála, Odhran O'Donoghue, Aleksandar Shtedritski, Sam Cox, Samuel G Rodriques, and Andrew D White. 2023. Paperqa: Retrieval-augmented generative agent for scientific research. *arXiv preprint arXiv:2312.07559* (2023).

Zachary Levonian, Chenglu Li, Wangda Zhu, Anoushka Gade, Owen Henkel, Millie-Ellen Postle, and Wanli Xing. 2023. Retrieval-augmented Generation to Improve Math Question-Answering: Trade-offs Between Groundedness and Human Preference. arXiv:2310.03184 [cs.CL] `https://arxiv.org/abs/2310.03184`

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

Xinxi Lyu, Michael Duan, Rulin Shao, Pang Wei Koh, and Sewon Min. 2025. Frustratingly Simple Retrieval Improves Challenging, Reasoning-Intensive Benchmarks. arXiv:2507.01297 [cs.CL] https://arxiv.org/abs/2507.01297

Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2023. Teaching Small Language Models to Reason. arXiv:2212.08410 [cs.CL] https://arxiv.org/abs/2212.08410

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. arXiv:2501.19393 [cs.CL] https://arxiv.org/abs/2501.19393

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. 2023. OpenWebMath: An Open Dataset of High-Quality Mathematical Web Text. arXiv:2310.06786 [cs.AI] https://arxiv.org/abs/2310.06786

Liana Patel, Negar Arabzadeh, Harshit Gupta, Ankita Sundar, Ion Stoica, Matei Zaharia, and Carlos Guestrin. 2025. DeepScholar-Bench: A Live Benchmark and Automated Evaluation for Generative Research Synthesis. *arXiv preprint arXiv:2508.20033* (2025).

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910* (2020).

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2023. Are Emergent Abilities of Large Language Models a Mirage? arXiv:2304.15004 [cs.AI] https://arxiv.org/abs/2304.15004

Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. 2024. Openmathinstruct-2: Accelerating ai for math with massive open-source instruction data. *arXiv preprint arXiv:2410.01560* (2024).

Peng-Yuan Wang, Tian-Shuo Liu, Chenyang Wang, Yi-Di Wang, Shu Yan, Cheng-Xing Jia, Xu-Hui Liu, Xin-Wei Chen, Jia-Cheng Xu, Ziniu Li, and Yang Yu. 2025. A Survey on Large Language Models for Mathematical Reasoning. arXiv:2506.08446 [cs.AI] https://arxiv.org/abs/2506.08446

Ruida Wang, Jipeng Zhang, Yizhen Jia, Rui Pan, Shizhe Diao, Renjie Pi, and Tong Zhang. 2024. Theoremllama: Transforming general-purpose llms into lean4 experts. *arXiv preprint arXiv:2407.03203* (2024).

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* (2022).

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

Sean Welleck, Jiacheng Liu, Ronan Le Bras, Hannaneh Hajishirzi, Yejin Choi, and Kyunghyun Cho. 2021. NaturalProofs: Mathematical Theorem Proving in Natural Language. arXiv:2104.01112 [cs.IR] https://arxiv.org/abs/2104.01112

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-Math Technical Report: Toward Mathematical Expert Model via Self-Improvement. arXiv:2409.12122 [cs.CL] https://arxiv.org/abs/2409.12122

Table 2: AIME 2022–2024 results (per year). Each cell shows Pass@4 for AIME 2022/2023/2024.

Method	LLaMA-3.1-8B				Qwen2.5-32B			
	CompactDS MATH	CompactDS Full	Traces Qwen2-32B	Traces Gemini	CompactDS MATH	CompactDS Full	Traces Qwen2-32B	Traces Gemini
No-RAG	0/1/1	0/1/1	0/1/1	0/1/1	6/3/6	6/3/6	6/3/6	6/3/6
Vanilla-RAG	5/2/3	5/2/3	4/0/3	3/0/2	7/3/6	5/7/4	8/4/6	5/5/7
RAG-Restruct-8B	3/3/0	1/1/0	7/1/0	8/1/1	4/5/7	6/4/10	7/4/8	9/4/7
RAG-Restruct-32B	6/2/1	3/3/0	8/1/1	5/2/1	5/5/6	4/5/5	9/5/6	3/6/8

Method	OpenMath2-LLaMA-3.1-8B				Mathstral-7B-v0.1			
	CompactDS MATH	CompactDS Full	Traces Qwen2-32B	Traces Gemini	CompactDS MATH	CompactDS Full	Traces Qwen2-32B	Traces Gemini
No-RAG	1/3/4	1/3/4	1/3/4	1/3/4	1/4/3	1/4/3	1/4/3	1/4/3
Vanilla-RAG	5/1/2	7/3/2	7/3/5	7/1/3	4/3/0	2/1/0	3/2/0	3/2/2
RAG-Restruct-8B	5/3/1	4/4/2	6/3/4	6/3/3	1/3/2	1/3/2	5/1/2	6/2/3
RAG-Restruct-32B	3/3/1	3/5/3	6/2/0	6/3/3	5/2/1	5/2/1	9/2/1	5/1/3

Table 3: Pass@4 over 90 questions in AIME 2022–2024 results (sum over three years, out of 90). Best values in each column are in bold. Best value for each model is shown in blue.

Method	LLaMA-3.1-8B				Qwen2.5-32B			
	CompactDS MATH	CompactDS Full	Traces Qwen2-32B	Traces Gemini	CompactDS MATH	CompactDS Full	Traces Qwen2-32B	Traces Gemini
No-RAG	2	2	2	2	15	15	15	15
Vanilla-RAG	**10**	**10**	7	5	**16**	16	18	17
RAG-Restruct-8B	6	2	8	**10**	**16**	**20**	19	**20**
RAG-Restruct-32B	9	6	**10**	8	**16**	14	**20**	17

Method	OpenMath2-LLaMA-3.1-8B				Mathstral-7B-v0.1			
	CompactDS MATH	CompactDS Full	Traces Qwen2-32B	Traces Gemini	CompactDS MATH	CompactDS Full	Traces Qwen2-32B	Traces Gemini
No-RAG	8	8	8	8	**8**	**8**	8	8
Vanilla-RAG	8	**12**	**15**	11	7	3	5	7
RAG-Restruct-8B	**9**	10	13	**12**	6	6	8	**11**
RAG-Restruct-32B	7	11	8	12	**8**	**8**	**12**	9

A AIME results

Here, we present results on the AIME benchmark for the years 2022, 2023, and 2024, with each year consisting of 30 questions (90 questions in total). We evaluate models using Pass@4, where the model is given up to four attempts at temperature 0.6 and is counted correct if any attempt matches the gold answer. Table 2 reports per-year results for AIME 2022, 2023, and 2024, while Table 3 aggregates these into totals across all three years. Together, these tables highlight both year-by-year performance and aggregate trends across AIME 2022–2024. We note that percentages are not reported, as the absolute numbers are small and percentage values would exhibit high variance.

From the results, we observe that vanilla RAG can sometimes help, as with LLaMA-3.1-8B, where performance improved from 2/90 without retrieval to 10/90 with vanilla RAG. However, restructuring yields more consistent gains. For the stronger Qwen2.5-32B, vanilla RAG offered little benefit, but restructuring the retrieved content improved performance from 15 to 20 correct answers, showing that careful content design is crucial for larger models. For Mathstral-7B, which is already math-specialized, vanilla RAG did not provide improvements, whereas restructuring with Traces was only marginally helpful. Overall, these findings suggest that achieving improvements with RAG on more challenging benchmarks like AIME (compared to MATH benchmark) requires systematic exploration of how to restructure content to make retrieval more effective.

B Evaluation

For MATH, we follow the dataset's seven categories and sample 100 problems per category (700 total); answers are judged by exact numeric match after normalization using the `MINERVA_MATH::LLAMA3.1` configuration from OLMES Gu et al. (2025), which reproduces the evaluation setup used by `LLaMA-3.1` in Dubey et al. (2024). To account for the non-deterministic nature of decoding, we report Average@4 accuracy on MATH, where the model is sampled four times per question. For AIME, we report Pass@4, i.e., whether the correct solution appears in any of four sampled generations.

Prompt for Solution Generation

Solve the following math problem efficiently and clearly:
- For simple problems (2 steps or fewer): Provide a concise solution with minimal explanation.
- For complex problems (3 steps or more): Use this step-by-step format:
Step 1: [Concise description] [Brief explanation and calculations]
Step 2: [Concise description] [Brief explanation and calculations]
...
Regardless of the approach, always conclude with:
Therefore, the final answer is: `$\boxed{answer}$`. I hope it is correct.
Where [answer] is just the final number or expression that solves the problem.
Problem:

Figure 2: Prompt used for solution generation with chain-of-thought.

Prompt. We use the CoT prompt shown in Figure 2 to generate solutions.

C Corpora

We retrieve from four different corpora to study how the nature of retrieved material impacts RAG for mathematical problem solving. To ensure the robustness of our evaluation, following Lyu et al. (2025), all corpora have been decontaminated by filtering out any paragraph with more than 70% 13-gram Jaccard similarity to queries in our evaluation datasets.

1. **CompactDS–Math:** The math-only portion of the CompactDS raw-text release on Hugging Face,[1] which provides combines OpenWebMath Paster et al. (2023), a collection of filtered math webpages from Common Crawl, and NaturalProofs Welleck et al. (2021), a corpus of theorems, proofs, definitions, and related content.
2. **CompactDS–Full:** the full web-scale datastore introduced in, which combines diverse high-quality sources (web crawls, curated math, academic papers, textbooks). We access this corpus via the CompactDS API.[2] Compared to the math-only split, the full CompactDS is substantially larger and more diverse.
3. **S1–Gemini:** thinking traces released in the SimpleScaling ablation study dataset Muennighoff et al. (2025) [3] generated by Gemini 2.0 Flash. We use these as retrievable worked-step exemplars.
4. **S1–Qwen3-32B:** we additionally generate thinking traces on the same decontaminated MATH portion of S1 using Qwen3-32B, in order to study the effect of corpus quality on RAG for mathematical problem solving. We use the same prompt/template as the released S1 data.

D Restructuring Example

Figure 4 shows an example of retrieved chunk and restructured format of it.

[1] `https://huggingface.co/datasets/alrope/CompactDS-102GB-raw-text`
[2] `https://github.com/berkeleyljj/Massive-Serve-Jinjian`
[3] `https://huggingface.co/datasets/simplescaling/data_ablation_full59K/`

Prompt for Restructuring the Corpus

Instruction. You are given a math problem and its solution. Your goal is to rewrite the solution into a clearly labeled, step-by-step concise format that teaches how to solve the problem.

Guidelines.
- Each step should reflect a logical phase in solving the problem.
- Use a concise "cheatsheet" style so learners can generalize the strategy to harder or related problems.
- If the problem or solution is incomplete or noisy, infer reasonable steps and still provide guidance.
- Focus on key ideas, strategies, or techniques the learner can reuse.
- Keep at most 7 steps; be concise and avoid verbosity.
- Avoid over-fragmentation; combine actions logically where appropriate.

Output format.

```
Problem: [Copy the problem exactly as given]
Step 1: [Short label]
[Brief explanation of what's being done and why, with any relevant
calculations]
Step 2: [Next short label]
[Brief explanation and math]
...
Step N: [Final step]
[Brief explanation and final processing or insight]
Therefore, the final answer is: $\boxed{[final answer]}$
```

Given problem: `{retrieval_text}`

Figure 3: Prompt used to restructuring the retrieved chunks.

Problem

A band is marching in a rectangular formation with dimensions $n-2$ and $n+8$ (members). In the second stage of their performance, they re-arrange to form a different rectangle with dimensions n and $2n-3$, excluding all the drummers. If there are at least 4 drummers, then find the sum of all possible values of n.

(a) Problem

Retrieved Text

Difference between revisions of "2005 AIME I Problems/Problem 4" Problem The director of a marching band wishes to place the members into a formation that includes all of them and has no unfilled positions. If they are arranged in a square formation, there are 5 members left over. The director realizes that if he arranges the group in a formation with 7 more rows than columns, there are no members left over. Find the maximum number of members this band can have.
Solution 1 If $n > 14$ then $n^2 + 6n + 14 < n^2 + 7n < n^2 + 8n + 21$ and so $(n+3)^2 + 5 < n(n+7) < (n+4)^2 + 5$. If n is an integer there are no numbers which are 5 more than a perfect square strictly between $(n+3)^2 + 5$ and $(n+4)^2 + 5$. Thus, if the number of columns is n, the number of students is $n(n+7)$ which must be 5 more than a perfect square, so $n \leq 14$. In fact, when $n = 14$ we have $n(n+7) = 14 \cdot 21 = 294 = 17^2 + 5$, so this number works and no larger number can. Thus, the answer is `$\boxed{[294]}$`.
Solution 2 Define the number of rows/columns of the square formation as s, and the number of rows of the rectangular formation r (so there are $r-7$ columns). Thus, $s^2+5 = r(r-7) \implies r^2 - 7r-$"

(b) Retrieved Text

Restructured Retrieved Text

Problem: The director of a marching band wishes to place the members into a formation with no unfilled positions. In a square formation there are 5 members left over. In a rectangular formation with 7 more rows than columns, there are none left over. Find the maximum number of members.
Step 1: Let columns $= n$, rows $= n+7$, so members $= n(n+7)$. Also members $= s^2 + 5$.

$$n(n+7) = s^2 + 5$$

Step 2: For $n > 14$,

$$(n+3)^2 + 5 < n(n+7) < (n+4)^2 + 5$$

contradiction. So $n \leq 14$.
Step 3: Check $n = 14$:

$$14 \cdot 21 = 294 = 17^2 + 5 \quad (\text{since } 17^2 = 289)$$

Answer: `$\boxed{[294]}$`

(c) Restructured Retrieved Text

Figure 4: Example restructured retrieved content for Problem shown in Figure (a), Figure (b) shows the Retrieved Text, and (c) shows the Restructured version of Retrieved Text.