

SEMANTIC SEARCH OVER 9 MILLION MATHEMATICAL THEOREMS

Luke Alexander^{1,2,*}, **Eric Leonen**^{1,3,*}, **Sophie Szeto**^{1,2,4,*}, **Artemii Remizov**^{1,5},
Ignacio Tejada^{1,2}, **Giovanni Inchiostro**^{1,2}, **Vasily Ilin**^{1,2,†*}

¹Math AI Lab, University of Washington, Seattle, WA, USA

²Department of Mathematics, University of Washington, Seattle, WA, USA

³Department of Applied and Computational Mathematical Sciences,
University of Washington, Seattle, WA, USA

⁴Paul G. Allen School of Computer Science and Engineering, University of Washington, Seattle, WA, USA

⁵Lake Washington High School, Kirkland, WA, USA

*Equal contribution

†Correspondence to: vilin@uw.edu

ABSTRACT

Searching for mathematical results remains difficult: most existing tools retrieve entire papers, while mathematicians and theorem-proving agents often seek a specific theorem, lemma, or proposition that answers a query. While semantic search has seen rapid progress, its behavior on large, highly technical corpora such as research-level mathematical theorems remains poorly understood. In this work, we introduce and study semantic theorem retrieval at scale over a unified corpus of 9.2 million theorem statements extracted from arXiv and seven other sources, representing the largest publicly available corpus of human-authored, research-level theorems. We represent each theorem with a short natural-language description as a retrieval representation and systematically analyze how representation context, language model choice, embedding model, and prompting strategy affect retrieval quality. On a curated evaluation set of theorem-search queries written by professional mathematicians, our approach substantially improves both theorem-level and paper-level retrieval compared to existing baselines, demonstrating that semantic theorem search is feasible and effective at web scale. The project page, search tool, dataset, REST API, and MCP server are available at theoremsearch.com

1 INTRODUCTION

Mathematical knowledge is organized around discrete results: theorems, lemmas, propositions, and corollaries. These statements serve as the fundamental units of reasoning for both human mathematicians and automated proof systems Polu & Sutskever (2020); Yang et al. (2023); Wu et al. (2022). A researcher proving a new result must first determine whether the statement already exists in the literature, and similarly, an AI agent generating formal proofs benefits from retrieving relevant lemmas to guide its search. Yet most existing tools – Google Scholar, arXiv, and even modern LLMs with web access – operate at the level of entire documents, forcing users to manually scan papers when they seek a specific statement.

This gap is increasingly significant. arXiv hosts over 2.4 million papers, including more than 690,000 in mathematics Ginsparg (1994). A study of over 14,000 withdrawn arXiv preprints found that 2.5% were retracted because the authors’ results already appeared in prior literature Rao et al. (2024). For example, Popescu-Pampu (2007), Zhang (2012), and Shahryari (2020) were withdrawn

*We thank Michele Pernice and Dori Bejleri for contributing several entries to the validation set. CPU and GPU computing, LLM inference and data storage were in part done using AWS credits from the UW eScience School and UW IT. CPU and GPU computing was in part done using the UW Research Computing Club funded from the UW Student Technology Fee Committee. GPU computing was in part done on UWIT’s GPU cluster Tillicum. GPU computing and model inference were in part done using Nebius and TokenFactory.

after discovering their main results had been previously established. AI systems face the same problem: the Erdős Problems Project documented cases where AI tools “solved” open problems that had been established decades earlier Erdős & Newman (1977); Wirsing & Schwarz (1961); Klarner (1966) – underscoring the need for theorem-level search.

In this work, we construct a corpus of over 9 million theorem statements from arXiv, the Stacks Project, ProofWiki, and five other sources, and study semantic retrieval at scale. We represent each theorem using a natural-language “slogan” generated by an LLM, then embed slogans and queries into a shared semantic space. Our main contributions are:

1. **A large-scale theorem corpus.** We release over 9 million theorem statements with rich metadata, the *largest* collection of informal mathematical theorems to date.
2. **A systematic study of representation choices.** We analyze how the context, LLM choice, embedding model, and prompting strategy affect retrieval. Notably, we find that embedding theorems via natural-language slogans significantly outperforms embedding their raw \LaTeX formulations.
3. **State-of-the-art retrieval.** On 111 queries from professional mathematicians, we achieve 45.0% Hit@20 at the theorem level, outperforming ChatGPT 5.2 with search (19.8%) and Gemini 3 Pro (27.0%). For paper-level retrieval, we achieve 56.8% Hit@20 compared to 37.8% for Google Search.

Our results demonstrate that semantic theorem search is feasible at web scale. The project page is at theoremsearch.com. The dataset is available at huggingface.co/datasets/uw-mathai/theorem-search-dataset. A datasheet for our dataset is provided in Appendix A.

2 RELATED WORK

Mathematical Information Retrieval. Early MathIR work focused on formula-level retrieval, with the NTCIR Math Tasks Aizawa et al. (2014); Zanibbi et al. (2016) establishing benchmarks for formula search over arXiv and Wikipedia. The ARQMath shared tasks Mansouri et al. (2022) extended this to mathematical question answering over Math Stack Exchange. A recent survey Dadure et al. (2024) notes that while formula retrieval has progressed, semantic understanding of mathematical statements remains an open challenge – the gap our work addresses.

Dense Retrieval and LLM-Augmented Search. Dense Passage Retrieval Karpukhin et al. (2020) showed that dual-encoder architectures can outperform sparse methods like BM25. Sentence-BERT Reimers & Gurevych (2019) enabled efficient semantic similarity via Siamese networks, while E5 Wang et al. (2022), Qwen3-Embedding Zhang et al. (2025), and Gemma Embedding Vera et al. (2025) have pushed embedding quality further. ColBERT Khatib & Zaharia (2020) introduced late interaction for fine-grained token matching. Retrieval-augmented generation (RAG) Lewis et al. (2020) combines retrieval with language model generation, enabling systems to ground responses in retrieved documents. These advances underpin our approach, though mathematical text poses unique challenges due to symbolic notation.

Search for Formal Mathematics. LeanSearch Gao et al. (2024) provides semantic search over Mathlib4’s 230,000+ theorems by generating natural-language descriptions and using dense retrieval. ReProver Yang et al. (2023) uses retrieval-augmented generation to select premises during proof search, and Numina-Lean-Agent Liu et al. (2026) integrates multiple search tools, including LeanDex, to retrieve lemmas across libraries. LeanFinder Lu et al. (2025) focuses on user intent, while LeanExplore Asher (2025) combines embeddings with BM25+ and PageRank. Jiang et al. (2023) demonstrate that language models can translate between formal and natural language mathematics. However, formal libraries cover only a fraction of mathematical knowledge; our work extends informalization-based retrieval to millions of \LaTeX theorem statements.

Scientific Literature Search. General academic search engines (Google Scholar, Semantic Scholar) and arXiv provide paper-level retrieval but cannot target individual theorems. The zbMATH database Steinfeldt & Mihaljević (2024) offers curated paper-level indexing. Large language models with web access, such as GPT-4 Achiam et al. (2023) and Gemini Google DeepMind (2025), can

Table 1: Distribution of theorem types in our dataset.

Theorem Type	Count
Lemma	3,280,463
Theorem	2,864,668
Proposition	2,067,371
Corollary	1,034,259

3.1 THEOREM PARSING FROM ARXIV

We source 99.5% (9.2 million) of our theorems from papers in the arXiv database tagged with `math`, `stat`, `cs`, `physics`, `eess`, `econ`, `q-fin`, or `q-bio`. A full breakdown of the statement types and arXiv categories across the dataset are provided in Table 1, Figure 2, and Table 7 in Appendix E.

Our objective in parsing theorems is to extract theorem *names* and *bodies*. A theorem *name* has three components:

1. The theorem type (e.g., Theorem, Proposition, Lemma, or Corollary),
2. An optional reference number, and
3. An optional note. For example, a theorem name might look like “Theorem 3.9 (Shokurov reduction)”.

We define the *body* as the \LaTeX content of the theorem, in which basic author-defined macros are expanded. For instance, if an author defines `\R` as a macro for `\mathbb{R}` in the \LaTeX preamble, we replace all occurrences of `\R` accordingly.

The variety of ways authors define theorem environments makes parsing theorems difficult. Thus, we parse theorems for each paper using three strategies:

1. *Node Search with plasTeX*: We use the Python library `plasTeX` Arnold (2009) to convert \LaTeX sources into a structured node tree. Every command and environment corresponds to a node. We traverse this tree to find environments corresponding to theorem-like structures. Then, we extract the theorem type, reference number, note, and body from the node metadata. This method successfully parses 422 thousand papers, yielding approximately 6.9 million theorems. However, `plasTeX` occasionally truncates theorem bodies when a paper’s source relies on \LaTeX packages unknown to the parser. To remove such malformed extractions, we apply simple heuristics, such as filtering out theorems shorter than 8 characters or ending in “and” or “let”.
2. *TeX Logging*: As a fallback to `plasTeX`, we generate and inject a custom \LaTeX package that logs theorem data. When a paper’s source is compiled, the package records the theorem type, reference number, note, and body for all theorem environments. We parse 137 thousand papers this way, yielding approximately 1.8 million theorems.
3. *Regex-based Parsing*: As a fallback to TeX logging, we use regular expressions to identify theorem delimiter tokens such as `\begin{...}` and `\end{...}` or `\proclaim` and `\endproclaim`. We then parse the content between delimiters to extract the theorem’s note and body. Reference numbers are not captured unless they are explicitly stated. We parse 30,000 papers this way, yielding approximately 542,000 theorems.

3.2 THEOREM PARSING FROM OTHER SOURCES

The remaining 0.5% (38,974) of our theorems come from a variety of other sources: ProofWiki ProofWiki Contributors (23,871), the Stacks Project The Stacks Project Authors (12,693), the Open Logic Project The Open Logic Project (745), the CRing Project The CRing Project Contributors (546), Stacks and Moduli Alper (2026) (506), the HoTT Book The Univalent Foundations Program (2013) (382), and An Infinitely Large Napkin Chen (231). These sources were chosen for their verifiability through open-source contributions and structured, accessible \LaTeX files. We favored

graduate-level texts, as their theorems would be useful to a mathematician and are often cited by researchers.

The Stacks Project, Open Logic Project, CRing Project, HoTT Book, and Infinitely Large Napkin are all hosted as open-source repositories on GitHub. Theorems from these sources were exclusively delimited in the source by `\begin{...}` and `\end{...}` tags. This uniform structure allowed us to build a single regex-based parser, similar to our regex-based parser for arXiv, that could extract theorems from all five sources. Our parser first normalizes shorthand environment names, then extracts the theorem name, body, label, and type from each environment. When a theorem body begins with a bracketed note, we append it to the theorem name in parentheses for easier identification. Labels defined with `\label{...}` are captured to preserve cross-referencing information from the original text. Since these repositories use GitHub’s file structure, we generate source URLs pointing to the original `.tex` files, enabling users to verify theorems against their source material. In addition, a custom theorem counter was used to reliably generate reference numbers.

ProofWiki required a different approach, as it is a wiki-based encyclopedia built on MediaWiki rather than \LaTeX . ProofWiki stores content in wikitext format with mathematical expressions enclosed in `$...$` tags. We developed a separate parser that interfaces with the MediaWiki API to programmatically retrieve pages from relevant categories. For each page, we extract the theorem statement by identifying section headers and capturing content up to the proof section. The wikitext is then cleaned by removing MediaWiki-specific markup such as `<onlyinclude>` tags, template calls, and wiki links, while converting `<math>` blocks to standard delimiters. Since ProofWiki assigns a unique URL to each theorem, we preserve these links to allow direct verification.

3.3 THEOREM REPRESENTATION

Theorems in our corpus are exclusively represented in \LaTeX , and often lack concise natural-language summaries. To obtain searchable textual representations and improve retrieval performance, we generate a short natural language description, or *slogan*, for each theorem using a large language model (LLM). This converts the retrieval task from a symmetric search over formal notation to an asymmetric task where informal queries retrieve formalized content Wang et al. (2022). Given the parsed theorem body, we prompt the DeepSeek V3 model Liu et al. (2024) to produce a concise, declarative English description of the theorem’s main result. Prompts instruct the LLM to avoid symbolic notation, proof details, and references to surrounding document structure in the generated slogans. The resulting slogans are stored as the primary textual representation for theorem retrieval.

To study the effect of additional mathematical context on slogan quality and retrieval performance, we evaluate three slogan-generation strategies:

1. **Body Only:** The prompt includes only the parsed theorem body.
2. **Body + Abstract:** The prompt includes the theorem body together with the paper abstract.
3. **Body + Introduction:** The prompt includes the theorem body together with the paper introduction.

Each strategy uses a similar prompt template as shown in Table 13 in Appendix G. Their major differences are in the contextual fields provided to the model. Model temperature is fixed at 0.2, and the maximum output tokens is fixed at 1024. Slogans are generated independently for each variant and treated as separate representations of the same underlying theorem during evaluation. The total cost of building the corpus was approximately \$6,000 USD: \$4,000 for LLM API calls to generate slogans across all 9.2 million theorems, and \$2,000 for compute and S3 storage.

3.4 RESULT RETRIEVAL

We embed theorem slogans and user queries using the Qwen3-Embedding-0.6B model Zhang et al. (2025), which maps natural language inputs to fixed-dimensional vectors. All theorem slogans are embedded offline following their generation by the LLM and stored in a PostgreSQL database Stonebraker & Rowe (1986) with the pgvector extension pgvector Contributors (2024). The database employs a Hierarchical Navigable Small World (HNSW) index Malkov & Yashunin (2020), enabling

fast approximate nearest-neighbor search. User queries are embedded at inference time using the same model. Within this shared embedding space, we retrieve the top- k theorems ranked by their cosine distance to the query.

3.5 VALIDATION SET

Enlisting the help of three research mathematicians, we curated a set of 111 distinct math queries across 14 arXiv tags, mainly Algebraic Geometry, Analysis, and PDEs. See Table 6 in Appendix E for a detailed breakdown. These queries search for distinct theorems, lemmas, corollaries, and propositions written by a small number of authors whose work our subject matter experts are well-acquainted with. Crucially, queries were written *blind*: each mathematician composed natural-language descriptions of theorems they knew from memory, without access to the corpus or its slogans, to prevent any leakage of retrieval representations into the evaluation queries. We then asked them to select a larger dataset of papers in which they were confident that the results of the queries they wrote appeared at most once; this larger dataset consisted of 7,356 papers. We used this larger dataset to guide our decisions on which embedders to use, which LLM to use for the slogans, how to prompt the embedder, and how much context to give to the LLM when generating the slogans (body of the result only, body+abstract, body+first section). Every query–theorem pair was subsequently verified in a two-stage quality check: an LLM confirmed that the target theorem exists in the corpus and semantically matches the query, and a second mathematician independently reviewed each pair to ensure correctness.

We remark that while small, this size is typical of mathematical information retrieval datasets, such as the query set of LeanSearch for Mathlib4 Gao et al. (2024) and ARQMath-3 Mansouri et al. (2022), which contain 50 and 78 queries, respectively.

4 EXPERIMENT METHODOLOGY

In this section, we report the evaluation performance of existing retrieval methods against our search engine. We outline the experiment setup, compare the performance against existing retrieval tools, and perform an ablation study on both the context window and the LLM used for slogan generation.

4.1 EXPERIMENTS

To measure the performance of our search engine against existing literature review tools, we employ three commonly used information retrieval metrics, following standard practice in mathematical retrieval benchmarks Mansouri et al. (2022); Dadure et al. (2024). Firstly, Precision@ k calculates the proportion of relevant documents in the first k results of a query.

$$\text{Precision@}k = \frac{1}{|\mathcal{Q}|} \sum_{i=1}^{|\mathcal{Q}|} \frac{1}{k} \sum_{j=1}^k \mathbb{I}(i, j)$$

where \mathcal{Q} is the set of queries and $\mathbb{I}(i, j)$ is an indicator function that equals one if and only if the j -th result of the i -th query is an exact match, and is zero otherwise. Similar to Precision, we calculate Hit@ k , also known as the hit rate:

$$\text{Hit@}k = \frac{1}{|\mathcal{Q}|} \sum_{i=1}^{|\mathcal{Q}|} \mathbb{I}(i, d_i)$$

where $d_{i,k}$ denotes the top- k results of the i -th query. This indicator function is 1 if and only if at least one of the top- k results is an exact match, and is zero otherwise. Because each query targets at most one exact match, Precision@ k and Hit@ k target complementary behaviors: early accuracy and robustness to ranking noise.

Another metric we use is Mean Reciprocal Rank (MRR@ k) Craswell (2009), which takes the reciprocal of the rank at which an exact match is discovered:

$$\text{MRR@}k = \frac{1}{|\mathcal{Q}|} \sum_{i=1}^{|\mathcal{Q}|} \frac{1}{\text{rank}_i}$$

where rank_i is the position of the *first* relevant result.

4.2 SETUP

We chose three embedders to use in our experiments: Google’s Gemma 0.3B Vera et al. (2025), and both Qwen3 0.6B and 8B Zhang et al. (2025). Our baseline methods consist of the following: Filtered Google Search (`site:arxiv.org + Query`), arXiv advanced search, ChatGPT 5.2 w/ Search, and Gemini 3 Pro Google DeepMind (2025); the prompts used for the LLM baselines are provided in Table 14 in Appendix G.

Both Google and arXiv are unable to return specific math statements, so their performance is graded on their ability to locate the correct paper. Conversely, LLM-based systems like ChatGPT and Gemini occasionally return correct results with incorrect reference numbers; we grade these as misses on theorem-level retrieval and treat them as paper matches for paper-level retrieval.

We emphasize that these baselines are not designed for theorem-level retrieval, but we evaluate them to reflect the tools currently used by LLMs and research mathematicians in the absence of specialized theorem-level search systems.

4.3 MAIN RESULTS

Table 2: Results on the validation set, comparing embedder performance against existing literature search tools. Values are reported as **theorem-level** / **paper-level**. Google Search and arXiv search cannot return specific theorem statements, so only paper-level results are reported. Our corpus was restricted to arXiv papers for evaluation.

Model	P@1	Hit@10	Hit@20	MRR@20
Baselines				
Google Search [†]	0.162	0.378	0.378	0.237
arXiv Search [†]	0.009	0.018	0.027	0.011
Chat-GPT 5.2	0.117	0.180	0.198	0.139
Gemini 3 Pro	0.171	0.252	0.270	0.196
Our Methods				
Gemma 0.3B (\LaTeX)	0.027 / 0.054	0.090 / 0.117	0.090 / 0.135	0.041 / 0.070
Gemma 0.3B	0.081 / 0.108	0.189 / 0.225	0.252 / 0.306	0.118 / 0.154
Qwen3 0.6B	0.081 / 0.153	0.234 / 0.342	0.270 / 0.351	0.132 / 0.215
Qwen3 8B	0.171 / 0.243	0.387 / 0.505	0.450 / 0.568	0.243 / 0.328
Qwen3 8B w/ Reranker	0.189 / 0.324	0.432 / 0.613	0.450 / 0.631	0.270 / 0.416

[†]Paper-level only: these tools cannot return individual theorems.

Our results are presented in Table 2. We find that the embedder Qwen3 8B outperforms existing literature review methods across all metrics. In particular, Qwen3 8B achieves substantially higher Hit@10 and Hit@20, indicating that the correct theorem is frequently retrieved within the top candidate set even when it is not ranked first. This behavior is desirable in large-scale theorem retrieval, where downstream reranking or human inspection can refine results once relevant candidates have been surfaced. We further apply a cross-encoder reranker (Qwen3-Reranker-0.6B Zhang et al. (2025)) to rescore the top-100 candidates retrieved by Qwen3 8B; this improves theorem-level P@1 from 17.1% to 18.9% and MRR@20 from 24.3% to 27.0%, confirming that late interaction over full query–slogan pairs captures fine-grained semantic distinctions that the bi-encoder alone misses.

For search methods at the paper-level, Qwen3 8B continues to outperform the baseline models, achieving higher Precision and MRR. The arXiv search is limited in the fields it can search over and does not search within the paper body, so it was only able to locate two papers across the whole query set in the top-20 results.

We also observed that Chat-GPT and Gemini tend to cluster their outputs by paper, returning multiple theorems from the same paper before moving to the next. This reduces result diversity and lowers Hit@ k : Gemini returns only 10.98 distinct papers per query on average, compared to 16.89

for Qwen3 8B. Our approach is particularly advantageous for retrieving auxiliary lemmas and technical results that appear deep within a paper, far from the title and abstract. Since our system indexes every theorem independently via its slogan, it is equally capable of retrieving a minor lemma as a paper’s headline theorem.

4.4 RETRIEVAL-AUGMENTED LLM REASONING

Beyond quantitative retrieval metrics, we investigate whether our theorem corpus can improve LLM reasoning on research-level mathematical questions. We posed a question about the boundary of a KSBA compactification of elliptic surface pairs to Claude (Opus 4.5). Without retrieval augmentation, Claude produced a confidently stated but *incorrect* answer, concluding that all boundary pairs must retain a fibration structure. The argument, while superficially plausible, relied on incorrect claims about the behavior of slc surface pairs under KSBA compactification.

When the same question was posed with access to our theorem database as a RAG tool, Claude retrieved relevant results from Ascher & Bejleri (2017); Ascher et al. (2018); Inchiostro (2018); Ascher & Bejleri (2016) and arrived at the *correct* answer: such boundary pairs do exist, and they are pseudoelliptic surfaces obtained by contracting the section of an elliptic fibration. The RAG-augmented response cited specific theorem numbers and provided a logically valid chain of reasoning grounded in retrieved results. This illustrates a key failure mode of LLM reasoning—without access to the relevant literature, the model confabulates plausible-sounding but incorrect mathematical arguments—and demonstrates that retrieval augmentation over our corpus can directly address it. The full query, both responses, and detailed citations are reported in Appendix D.

4.5 ABLATION STUDIES

In this subsection, we perform ablation studies on the context windows used for slogan generation, the LLMs used to generate slogans, and the document preparation process. For each ablation, we limit the corpus to arXiv papers labeled with the algebraic geometry tag (math.AG) and authored by individuals whose work contains an exact match to one of our queries. This filtering reduces the corpus to 7,356 statements written by 8 primary authors.

4.5.1 ABLATION OF CONTEXT WINDOWS

Table 3 presents the effects of context on slogan generation. Increasing or decreasing the amount of context provided to LLMs when generating math slogans considerably affects our search engine’s performance, while holding the embedder constant. We find that supplying the LLM with additional paper context improves the retrieval performance of its generated slogans. Performance falters when the LLM is only given the abstract, but improves significantly with the inclusion of the first section of the paper, defined by `\section{}`. We attribute this improvement to the LLM making better sense of the statement with more context, especially since theorems alone tend to be at most 3–4 sentences long and may reference earlier sections of the paper. Furthermore, the introduction of most math papers typically outlines the results and the existing research upon which they build. As a result, context-rich slogans better capture the semantic intent of the theorem, yielding more reliable and robust retrieval across queries. Results for an expanded ablation covering additional embedders are reported in Table 8 in Appendix F.

Table 3: Comparing context window size in retrieval performance. Embedded with prompt using Qwen3 8B Zhang et al. (2025). Slogans generated with DeepSeek V3 Liu et al. (2024).

Model	P@1	Hit@10	Hit@20	MRR@20
Body Only	0.342	0.658	0.737	0.451
w/ Abstract	0.316	0.645	0.737	0.429
w/ First Section	0.368	0.737	0.763	0.496

4.5.2 ABLATION OF SLOGAN GENERATOR

The LLM used to generate the slogans can also significantly affect our engine’s search performance, as some LLMs are better trained on mathematical texts than others and thus generate better informal versions of math statements. We find that leading proprietary models, such as Claude’s Opus 4.5 and Gemini 3 Pro Google DeepMind (2025), outperform open-source models such as DeepSeek-V3 Liu et al. (2024), as summarized in Table 4. A broader comparison across additional slogan generators and embedding models is provided in Table 9 in Appendix F.

Table 4: Comparing LLM slogans in retrieval performance. Embedded with task instruction using Qwen3 8B Zhang et al. (2025) Body + Abstract.

Model	P@1	Hit@10	Hit@20	MRR@20
Deepseek V3.1	0.316	0.645	0.737	0.429
Deepseek R1	0.276	0.671	0.697	0.388
Gemini 3 Pro	0.368	0.750	0.816	0.507
Claude Opus 4.5	0.395	0.776	0.842	0.536

4.5.3 ABLATION OF TASK INSTRUCTION

We examine the impact of the task instructions used during embeddings on retrieval performance in Table 5. We find that by providing a math retrieval instruction, both Qwen embedders achieve higher performance than without any instruction at all. However, this effect is reversed with Gemma, with empty instruction performing better, though worse than Qwen 8B with instruction. We use a modification of the prompts made by Gao et al. (2024), the details of which are listed in Table 12.

4.6 EMBEDDING SPACE ANALYSIS

Following Gao et al. (2024); Jiang et al. (2023), we convert \LaTeX theorem statements into natural-language slogans before embedding, since embedders struggle with symbol-heavy notation Bleckmann & Tschisgale (2025); Peng et al. (2021) and mathematicians typically query in informal language. To verify that sloganization induces semantically meaningful structure, we use PCA and UMAP as diagnostic tools on a random sample of 10,000 theorems (1,000 from each of the 10 most common arXiv categories).

Figure 3 shows PCA projections for three categories: Algebraic Geometry, Probability Theory, and Statistics Theory. Conceptually distant fields (Algebraic Geometry vs. Probability) are well separated, while closely related fields (Probability vs. Statistics) overlap substantially – consistent with mathematical intuition. UMAP projections over all ten categories in Figure 4 reveal that Qwen3 8B produces tighter, better-separated clusters than Gemma 0.3B, consistent with the retrieval gap in Table 2.

Table 5: Comparing prompting performance. Slogans generated by DeepSeek V3.1 Liu et al. (2024) on ‘Body+Abstract’ context window.

Model	P@1	Hit@10	Hit@20	MRR@20
Unprompted				
Gemma 0.3B	0.303	0.539	0.566	0.376
Qwen3 0.6B	0.224	0.487	0.526	0.297
Qwen3 8B	0.250	0.553	0.618	0.332
Prompted				
Gemma 0.3B	0.197	0.434	0.487	0.265
Qwen3 0.6B	0.237	0.566	0.658	0.346
Qwen3 8B	0.316	0.645	0.737	0.496



Figure 3: PCA visualizations of theorems in our dataset using slogan embeddings. Left: Gemma embedding. Right: Qwen3 8B embedding.

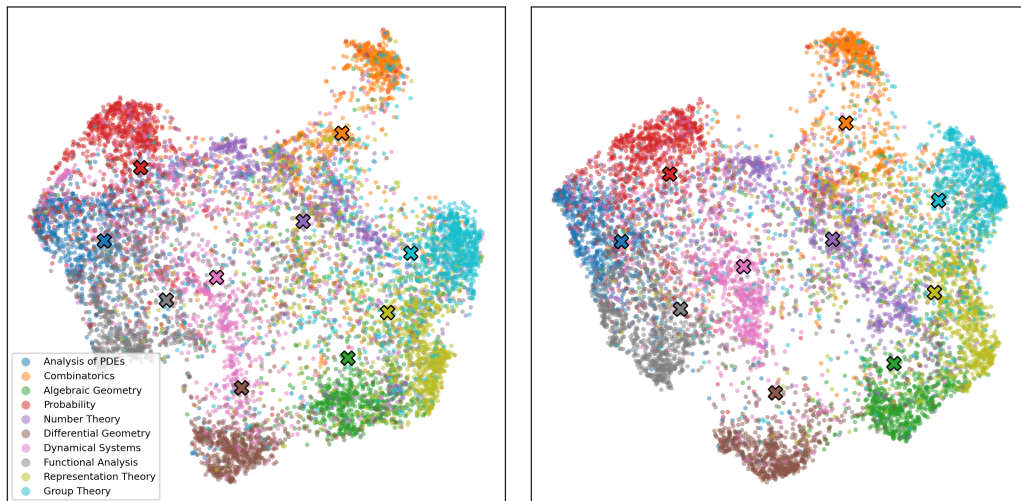


Figure 4: UMAP visualizations of 10,000 theorem slogan embeddings across the ten most common arXiv categories. Gemma 0.3B (top) and Qwen3 8B (bottom). Qwen3 8B produces tighter, better-separated clusters.

5 CONCLUSION

Mathematical knowledge is organized around discrete results – theorems, lemmas, propositions, and corollaries – yet existing search tools operate at the document level, forcing researchers and AI agents alike to manually locate specific statements within papers. In this work, we addressed this gap by constructing a corpus of over 9.2 million theorem statements extracted from arXiv and seven other sources, constituting the largest unified collection of human-authored, research-level theorems to date.

We showed that representing theorems via natural-language slogans generated by an LLM, rather than embedding raw \LaTeX directly, yields substantially better retrieval performance. Our systematic ablations revealed that context matters: slogans generated with access to the paper’s introduction outperform those generated from the theorem body alone, and the choice of both the slogan-generating LLM and the embedding model significantly affects downstream retrieval quality. On a curated evaluation set of 111 queries written by professional mathematicians, our best configuration achieves 45.0% Hit@20 at the theorem level and 56.8% Hit@20 at the paper level, outperforming ChatGPT 5.2 with search, Gemini 3 Pro, and Google Search.

By treating theorems as first-class retrieval objects and enriching them with natural-language descriptions, we enable new forms of access to mathematical knowledge – whether for retrieval-augmented generation by LLMs, for premise selection in formal proof search, or for literature review by expert mathematicians. We have deployed a public search interface described in Appendix B. Qualitative user feedback from research mathematicians, reported in Appendix C, further illustrates the practical utility of our system.

ACKNOWLEDGMENTS

We thank Michele Pernice and Dori Bejleri for contributing several entries to the validation set. CPU and GPU computing, LLM inference and data storage were in part done using AWS credits from the UW eScience School and UW IT. CPU and GPU computing was in part done using the UW Research Computing Club funded from the UW Student Technology Fee Committee. GPU computing was in part done on UWIT’s GPU cluster Tillicum. GPU computing and model inference were in part done using Nebius and TokenFactory.

REFERENCES

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- Akiko Aizawa, Michael Kohlhase, Iadh Ounis, and Moritz Schubotz. NTCIR-11 Math-2 Task Overview. In *Proceedings of the 11th NTCIR Conference*, pp. 88–98, Tokyo, Japan, 2014.
- Jarod Alper. Stacks and moduli. <https://sites.math.washington.edu/~jarod/moduli.pdf>, 2026.
- Tim Arnold. Getting started with plasTeX. *TUGboat*, 30(2):180–185, 2009. URL <https://www.tug.org/TUGboat/tb30-2/tb95arnold.pdf>.
- Kenneth Ascher and Dori Bejleri. Moduli of fibered surface pairs from twisted stable maps, 2016. URL <https://arxiv.org/abs/1612.00792>.
- Kenneth Ascher and Dori Bejleri. Log canonical models of elliptic surfaces, 2017. URL <https://arxiv.org/abs/1610.08033>.
- Kenneth Ascher, Dori Bejleri, and Giovanni Inchiostro. Moduli of weighted stable elliptic surfaces and invariance of log plurigenera, 2018. URL <https://arxiv.org/abs/1702.06107>.
- Justin Asher. Leanexplore: A search engine for lean 4 declarations. *arXiv preprint arXiv:2506.11085*, 2025.
- Yuri Bahturin, Alexander V Mikhalev, Viktor M Petrogradsky, and Mikhail V Zaicev. *Infinite dimensional Lie superalgebras*, volume 7. Walter de Gruyter, 2011.
- Tom Bleckmann and Paul Tschisgale. Evaluating nlp embedding models for handling science-specific symbolic expressions in student texts, 2025. URL <https://arxiv.org/abs/2505.17950>.
- Evan Chen. An infinitely large napkin. <https://venhance.github.io/napkin/>. Accessed 2026-01-25.
- Nick Craswell. *Mean Reciprocal Rank*, pp. 1703–1703. Springer US, Boston, MA, 2009. doi: 10.1007/978-0-387-39940-9_488.
- Pankaj Dadure, Partha Pakray, and Sivaji Bandyopadhyay. Mathematical information retrieval: A review. *ACM Computing Surveys*, 57(3):1–34, 2024. doi: 10.1145/3699953.
- Paul Erdős and Donald J. Newman. Bases for sets of integers. *Journal of Number Theory*, 9(4): 420–425, 1977.

- Guoxiong Gao, Haocheng Ju, Jiedong Jiang, Zihan Qin, and Bin Dong. A semantic search engine for mathlib4. In *Findings of the Association for Computational Linguistics: EMNLP 2024*. Association for Computational Linguistics, 2024. URL <https://arxiv.org/abs/2403.13310>.
- Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92, December 2021. doi: 10.1145/3458723.
- Paul Ginsparg. First steps towards electronic research communication. *Computers in Physics*, 8(4):390–396, 1994. doi: 10.1063/1.4823313.
- Google DeepMind. Gemini 3: Introducing the latest gemini AI model from Google. <https://blog.google/products/gemini/gemini-3/>, November 2025. Accessed 2026-01-25.
- Xinshuo Hu, Zifei Shan, Xinpeng Zhao, Zetian Sun, Zhenyu Liu, Dongfang Li, Shaolin Ye, Xinyuan Wei, Qian Chen, Baotian Hu, Haofen Wang, Jun Yu, and Min Zhang. Kalm-embedding: Superior training data brings a stronger embedding model, 2025. URL <https://arxiv.org/abs/2501.01028>.
- Giovanni Inchiostro. Moduli of weierstrass fibrations with marked section, 2018. URL <https://arxiv.org/abs/1808.03539>.
- Albert Q. Jiang, Wenda Li, and Mateja Jamnik. Multilingual mathematical autoformalization. 2023. URL <https://arxiv.org/abs/2311.03755>.
- Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6769–6781. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.550.
- Omar Khattab and Matei Zaharia. ColBERT: Efficient and effective passage search via contextualized late interaction over BERT. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 39–48. ACM, 2020. doi: 10.1145/3397271.3401075.
- David A Klarner. Representations of n as a sum of distinct elements from special sequences. *The Fibonacci Quarterly*, 4(4):289–306, 1966.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474. Curran Associates, Inc., 2020.
- Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.
- Junqi Liu, Zihao Zhou, Zekai Zhu, Marco Dos Santos, Weikun He, Jiawei Liu, Ran Wang, Yunzhou Xie, Junqiao Zhao, Qiufeng Wang, Lihong Zhi, Jia Li, and Wenda Li. Numina-lean-agent: An open and general agentic reasoning system for formal mathematics. 2026. URL <https://arxiv.org/abs/2601.14027>.
- Jialin Lu, Kye Emond, Kaiyu Yang, Swarat Chaudhuri, Weiran Sun, and Wuyang Chen. Lean finder: Semantic search for mathlib that understands user intents. 2025. URL <https://arxiv.org/abs/2510.15940>.
- Yu. A. Malkov and D. A. Yashunin. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):824–836, 2020. doi: 10.1109/TPAMI.2018.2889473.

- Behrooz Mansouri, Vít Novotný, Anurag Agarwal, Douglas W. Oard, and Richard Zanibbi. Overview of ARQMath-3 (2022): Third CLEF lab on answer retrieval for questions on math. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, volume 13390 of *Lecture Notes in Computer Science*, pp. 286–310. Springer, 2022. doi: 10.1007/978-3-031-13643-6_20.
- Shuai Peng, Ke Yuan, Liangcai Gao, and Zhi Tang. Mathbert: A pre-trained model for mathematical formula understanding, 2021. URL <https://arxiv.org/abs/2105.00377>.
- pgvector Contributors. pgvector: Vector similarity search for Postgres. <https://github.com/pgvector/pgvector>, 2024. Accessed 2026-01-25.
- Stanislas Polu and Ilya Sutskever. Generative language modeling for automated theorem proving. *arXiv preprint arXiv:2009.03393*, 2020.
- Patrick Popescu-Pampu. Stein or Milnor fillability and cohomology, 2007. Withdrawn; merged into arXiv:0712.3484 after discovering a theorem had already been proved by Durfee and Hain.
- ProofWiki Contributors. ProofWiki. <https://proofwiki.org>. Accessed 2026-01-25.
- Delip Rao, Jonathan Young, Thomas Dietterich, and Chris Callison-Burch. Withdrarxiv: A large-scale dataset for retraction study. *arXiv preprint arXiv:2412.03775*, 2024.
- Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1410.
- M. Shahryari. A note on surjunctive groups, 2020. Withdrawn; the main result had already been proved by Arzhantseva et al.
- Christian Steinfeldt and Helena Mihaljević. Evaluation and domain adaptation of similarity models for short mathematical texts. In *Intelligent Computer Mathematics: 17th International Conference, CICM 2024*, Lecture Notes in Computer Science, pp. 241–260, Montréal, QC, Canada, 2024. Springer. doi: 10.1007/978-3-031-66997-2_14.
- Michael Stonebraker and Lawrence A. Rowe. The design of POSTGRES. In *Proceedings of the 1986 ACM SIGMOD International Conference on Management of Data*, pp. 340–355, 1986. doi: 10.1145/16894.16888.
- The CRing Project Contributors. The CRing project. https://math.mit.edu/~hrm/the_cring_project.pdf. Accessed 2026-01-25.
- The Open Logic Project. Open logic. <https://openlogicproject.org>. Accessed 2026-01-25.
- The Stacks Project Authors. The stacks project. <https://stacks.math.columbia.edu>. Accessed 2026-01-25.
- The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study, 2013. URL <https://homotopytypetheory.org/book/>.
- Hamid Usefi. Isomorphism invariants of restricted enveloping algebras, 2009. URL <https://arxiv.org/abs/0804.2281>.
- Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftexhar Naim, Joe Zou, Feiyang Chen, et al. Embeddinggemma: Powerful and lightweight text representations. *arXiv preprint arXiv:2509.20354*, 2025.
- 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. *arXiv preprint arXiv:2212.03533*, 2022. URL <https://arxiv.org/abs/2212.03533>.

- Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. Multi-lingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*, 2024.
- Eduard Wirsing and Wolfgang Schwarz. Approximation mit algebraischen zahlen beschränkten grades. 1961.
- Yuhuai Wu, Albert Qiaochu Jiang, Wenda Li, Markus Rabe, Charles Staats, Mateja Jamnik, and Christian Szegedy. Autoformalization with large language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.
- Kaiyu Yang, Aidan M. Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Prenger, and Anima Anandkumar. LeanDojo: Theorem proving with retrieval-augmented language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023.
- Richard Zanibbi, Akiko Aizawa, Michael Kohlhase, Iadh Ounis, Goran Topić, and Kenny Davila. NTCIR-12 MathIR Task Overview. In *Proceedings of the 12th NTCIR Conference*, pp. 299–308, Tokyo, Japan, 2016.
- Shenxing Zhang. On non-congruent numbers with $\equiv 3 \pmod{8}$ prime factors, 2012. Withdrawn; the result is a corollary of a well-known result by Monsky.
- Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models. 2025. URL <https://arxiv.org/abs/2506.05176>.
- Xinping Zhao, Xinshuo Hu, Zifei Shan, Shouzheng Huang, Yao Zhou, Xin Zhang, Zetian Sun, Zhenyu Liu, Dongfang Li, Xinyuan Wei, Youcheng Pan, Yang Xiang, Meishan Zhang, Haofen Wang, Jun Yu, Baotian Hu, and Min Zhang. Kalm-embedding-v2: Superior training techniques and data inspire a versatile embedding model, 2025. URL <https://arxiv.org/abs/2506.20923>.

A DATASHEET

This section is based on *Datasheets for Datasets* by Gebru et al. (2021).

A.1 MOTIVATION

1. For what purpose was the dataset created?
 - We created this dataset to create the largest unified corpus of mathematical theorems. We use this dataset ourselves to create a theorem search engine.
2. Who created the dataset?
 - This paper’s co-authors collected theorems. All theorems were written by the authors of the theorems’ papers and are cited in our database.
3. Who funded the creation of the dataset?
 - Self-funded.

A.2 COMPOSITION

1. What do the instances that comprise the dataset represent?
 - Each instance of the dataset represents a mathematical statement derived from a paper, textbook, or repository online.
2. How many instances are there in total?
 - As of writing, there are 9.2 million mathematical statements in the dataset.
3. Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?
 - The dataset contains a non-random sample of instances from available math corpora.

4. What data does each instance consist of?
 - Each instance contains the name of the theorem as displayed in the source document (e.g. Theorem 1.1), the raw TeX body, the author-written label inside the body, and the paper it was sourced from (e.g. arXiv’s convention of 1234.56789v1).
 - In addition, we stored metadata from each source document. This data includes the title, the list of authors, the link to the document, the abstract, the journal the document was published in (if it was published), the primary and secondary categories, the number of citations (if available), and a tag of which source it came from (e.g. arXiv, Stacks Project).
5. Is there a label or target associated with each instance?
 - Each theorem uses its name derived from the source document, and lists the document it came from, creating a distinct method of identification.
6. Is any information missing from individual instances?
 - No.
7. Are relationships between individual instances made explicit?
 - Each instance is numbered according to the conventions of the source document. They also contain metadata linking them to a specific paper or document.
8. Are there recommended data splits? (e.g., training, development/validation, testing)
 - Not applicable.
9. Are there any errors, sources of noise, or redundancies in the dataset?
 - Some statement bodies, titles, and labels may be fragmented, as our parsing methods are not applicable to every TeX document in our sources.
10. Is the dataset self-contained, or does it link to or otherwise rely on external resources?
 - The data is derived from sources online.
11. Does the dataset contain data that might be considered confidential?
 - No, all data is publicly available online.
12. Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?
 - No.

A.3 COLLECTION PROCESS

1. How was the data associated with each instance acquired?
 - A LaTeX parser was used to parse all theorem names and bodies from sources.
2. What mechanisms or procedures were used to collect the data?
 - Our LaTeX theorem parser used the Python library `plasTeX` to parse LaTeX documents, with a custom TeX logger and a pure-regex parser as fallbacks.
3. If the dataset is a sample from a larger set, what was the sampling strategy?
 - Our dataset is not a sample. To the best of our knowledge, our dataset includes most mathematical theorems in arXiv. For the other sources, such as ProofWiki and the Stacks Project, we have included all theorems.
4. Who was involved in the data collection process, and how were they compensated?
 - This paper’s co-authors were all involved in designing and implementing the pipeline to collect theorems. The mathematicians who created the validation set are volunteers.
5. Over what timeframe was the data collected?
 - Data was collected over five months, from September 2025 to January 2026.
6. Were any ethical review processes conducted?
 - Not applicable.

A.4 PREPROCESSING/CLEANING/LABELING

1. Was any preprocessing/cleaning/labeling of the data done?
 - We preprocessed theorem bodies by removing unnecessary whitespace and expanding simple author-defined macros. We filtered out truncated or corrupted theorem bodies by employing several heuristics.
2. Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data?
 - No.
3. Is the software that was used to preprocess/clean/label the data available?
 - All code used to preprocess, clean, and label our data is available at [Redacted].

A.5 USES

1. Has the dataset been used for any tasks already?
 - The dataset has been used to create a semantic search engine, available at [Redacted].
2. Is there a repository that links to any or all papers or systems that use the dataset?
 - Not applicable.
3. What (other) tasks could the dataset be used for?
 - Our dataset of theorems is useful for mathematicians and for AI agents for automated theorem proving.
4. Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?
 - If authors of our theorem sources were to significantly change the content of the papers/textbooks/projects, our theorems would be outdated.
5. Are there tasks for which the dataset should not be used?
 - No.

A.6 DISTRIBUTION

1. Will the dataset be distributed to third parties outside of the entity on behalf of which the dataset was created?
2. How will the dataset be distributed?
 - We plan to share the entire dataset on Hugging Face upon acceptance.
3. When will the dataset be distributed?
 - Upon acceptance.
4. Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)?
 - No.
5. Have any third parties imposed IP-based or other restrictions on the data associated with the instances?
 - No.
6. Do any export controls or other regulatory restrictions apply to the dataset or individual instances?
 - No.

A.7 MAINTENANCE

1. Who will be supporting/hosting/maintaining the dataset?
 - The co-authors of this paper plan on continuing to build, host, and maintain this dataset.
2. How can the owner/curator/manager of the dataset be contacted?

- The co-authors of this paper own this dataset and can be contacted at [Redacted].
3. Will the dataset be updated?
 - We plan on continuing to improve and grow the dataset in the future. For example, we plan on adding theorems from nLab and other open-source textbooks and projects.
 4. If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were the individuals in question told that their data would be retained for a fixed period of time and then deleted)?
 - Not applicable.
 5. Will older versions of the dataset continue to be supported/hosted/maintained?
 - Older versions of the dataset will be available for download indefinitely on Hugging Face. However, all support and maintenance efforts will be focused on the latest version of the dataset.
 6. If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?
 - Others can suggest sources to add to the dataset by contacting the co-authors at [Redacted].

B SEARCH TOOL INTERFACE

The search tool is currently accessible via theoremsearch.com. Users can perform semantic searches using natural language (e.g., “a rational variety is simply connected”). The drop-down tab allows results to be filtered by relevant metadata (e.g., author sets, arXiv category tags, and journal publication status). To optimize retrieval time, the search uses a two-stage retrieval architecture. At inference time, the natural language query is encoded using the Qwen3-8B embedding model and binary quantized into 4096 bits. An initial pool of $\text{clamp}(\max(200, 12 \times k), 200, 800)$ candidates is returned using the HNSW index and Hamming distance, where k is a user-specified number of results to return. The candidate pool is then filtered by metadata and reranked using cosine similarity on the original 4096-dimensional embeddings. When weighting by citations is enabled, the candidates are instead re-ranked using the following composite scoring function:

$$\text{score} = \text{cosine similarity} + \lambda \cdot \log(\max(\text{citations}, 1))$$

where λ is the user-specified citation-weight parameter. The interface retrieves the top k results, each of which displays the theorem slogan, the theorem body rendered in LaTeX, any relevant paper metadata, and a link to the paper. Finally, a lightweight feedback system allows users to rate each result’s relevance to their query. The resulting end-to-end latency is approximately 4 seconds per query, compared to 60+ seconds for LLM-based search tools such as ChatGPT and Gemini, which must perform multiple web searches and parse documents at inference time. See Figure 5 for the interface.

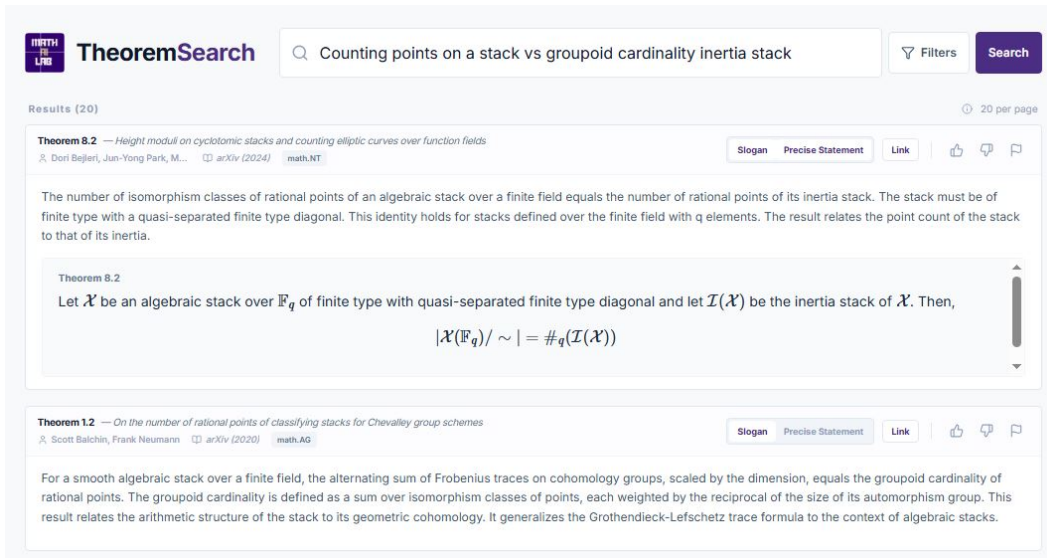


Figure 5: theoremsearch.com interface.

Because our slogans are multi-sentence descriptions, the embedding model produces vectors that are optimized for longer inputs. Short keyword-style queries (e.g., “Landau equation”) can therefore be dominated by padding or generic context, leading to poor retrieval. We find that simply repeating a short query (e.g., “Landau equation Landau equation”) substantially improves result quality by strengthening the signal of the key terms in the embedding. The search tool applies this duplication automatically when the query is below a length threshold.

In addition to the web interface, we provide a REST API at `api.theoremsearch.com/search` that accepts JSON queries and returns ranked theorem results, enabling programmatic integration into research workflows. We also expose a Model Context Protocol (MCP) server at `api.theoremsearch.com/mcp`, allowing AI agents and LLM-based tools to use theorem search as a retrieval tool during reasoning and proof generation. The project page, including links to the search tool, API, MCP server, and dataset, is available at `theoremsearch.com`.

C USER FEEDBACK

In this section, we present feedback from two users on our theorem search engine.

C.1 FEEDBACK ONE

We will first include a high-level description of the first feedback.

The user wanted to check for a specific reference, which we will call “Theorem 1”, which the user knew was true. They thought it follows from another result, which we call “result X,” a bit stronger than Theorem 1. It did, after a small lemma; in other words, (result X + small lemma) implies what the user wanted. This shifted the user’s attention from finding a reference for Theorem 1 to finding a reference for the small lemma. The user searched for the small lemma in our tool, and the first result was a reference for a generalization of the small lemma.

A more detailed description now follows. We advise the non-mathematician reader to skip the next paragraph.

The user wanted to check that a smooth variety over a separably closed field has a k -point. The user thought it follows from the étale local structure of smooth morphisms, which captured the correct idea but reduced the problem to showing that k -points are dense in \mathbb{A}_k^n . In essence, the étale local structure theorem translated the original problem of checking that a smooth variety over a separably

closed field has a k -point, to checking that for k separably closed, the set of k -points of \mathbb{A}^n is dense. The user entered the query “if k is separably closed, the set of k -points of \mathbb{A}^n is dense in \mathbb{A}^n .” The first result was correct: Tag 056U in the Stacks Project.

C.2 FEEDBACK TWO

We now describe a second use case. As before, we first provide a high-level summary.

The user searched for a result known to experts but lacked a specific citable reference; we call this result “Theorem X.” Many existing references point to similar results, but not exactly to Theorem X. The user queried our tool and found a result, which we call “Theorem Y,” whose citation list directly referenced Theorem X. By using our tool, the user located the desired theorem with fewer queries than traditional search methods, which had previously led only to indirect references.

A more detailed description follows. We advise the non-mathematician reader to skip the next two paragraphs.

The user needed a structure theorem for abelian p -Lie algebras, as defined by N. Jacobson. The user conjectured that these algebras are classified analogously to finite abelian groups, provided the base field is algebraically closed. On MathOverflow, an answer directed the user to a book that states the desired result. The book explained that the result follows from the structure theory for modules over a PID, without providing a precise proof, and attributes the technique to Jacobson.

The book *Infinite-Dimensional Lie Superalgebras* Bahturin et al. (2011) contains the result the user was looking for, labeled the 3.2 nil-radical Theorem, attributed to Seligman, 1967. However, the user skimmed through the hundred or so pages of Seligman’s reference several times without finding where the desired result is proved. When the user queried our tool with “over an algebraically closed field, any abelian p -Lie algebra splits as a torus and a sum of nilpotent cyclic Lie algebras,” the first result was the desired statement: Theorem 2.7 in Usefi (2009).

D CLAUDE EXPERIMENT

We now report the experiment coupling Claude with our theorem search tool as a RAG system. This experiment tests whether retrieval-augmented generation over our theorem corpus can correct LLM reasoning failures on a research-level mathematical question. For discussion of this experiment in context, see Section 4.4.

Query to Claude

Consider the closure, in the KSBA moduli space, of the moduli space of pairs $(S, aB + bF)$ where S is an elliptic surface, B is a section of the projection $S \rightarrow C$ which gives the elliptic surface structure to S , the fibers of $S \rightarrow C$ are smooth or nodal, F are some fibers of $S \rightarrow C$, and $0 < a, b < 1$. The coefficients are chosen so that $(S, aB + bF)$ is KSBA-stable (so don't need to reprove it). I would like to know if there exist a surface pair $f : (X', aB' + bF') \rightarrow C'$ appearing on the boundary of the KSBA-compactification, not admitting a fibration over a nodal curve C' , with fibers of f that are of dimension one.

Without access to our tool, Claude answered **incorrectly**:

Claude's response without RAG (incorrect)

No. Every surface pair appearing on the boundary of this KSBA compactification admits a fibration over a (possibly nodal) curve with one-dimensional fibers. The fibration structure cannot be lost in the limit.

Why the fibration persists. The essential point is that the elliptic fibration structure is rigidly encoded in the boundary divisor, and this encoding survives in the KSBA limit.

The fiber class remains nontrivial. In the interior, F is a fiber of $S \rightarrow C$, so $F^2 = 0$ and $(K_S + aB + bF) \cdot F > 0$ (since $K_S \cdot F = 0$, $B \cdot F = 1$, so this equals $a > 0$). In any flat limit, the limiting divisor F' retains $F'^2 = 0$ and $(K_{X'} + aB' + bF') \cdot F' > 0$. So F' is a nontrivial nef class with zero self-intersection, which defines a fibration $f : X' \rightarrow C'$.

Stable reduction preserves the fibered structure. Given a one-parameter degeneration $(\mathcal{S}_t, \Delta_t) \rightarrow \mathcal{C}_t$, after finite base change and birational modification one extends the fibration to the central fiber. Each component of the limiting surface X' either:

- maps surjectively onto a component of the nodal base curve C' with one-dimensional fibers, or
- maps to a point (a “sprouted” component).

But a component mapping to a point would be an entire surface sitting over a single point of C' , and the slc and ampleness conditions prevent this from occurring in the stable model: such a component would need $K + \Delta$ to be positive on it, yet its gluing locus with the rest of X' is along a single fiber curve, and the combinatorics of the slc condition force it to be absorbed (contracted) when passing to the ample model of $K_{X'} + \Delta'$.

The section prevents collapse. The section B' (limit of the section B) meets each fiber in a point, ensuring the map $X' \rightarrow C'$ has generically one-dimensional fibers throughout. If any component lost the fibration structure, B' restricted to that component would be a curve not meeting the fiber class transversally, contradicting the deformation-invariant intersection number $B \cdot F = 1$.

Summary. The divisor data $(aB + bF)$ is sufficiently rigid that all KSBA-stable limits retain a fibration $f : X' \rightarrow C'$ over a nodal curve C' , with all fibers of f being one-dimensional (curves). The boundary pairs differ from interior pairs in that C' may be nodal, X' may be reducible/non-normal (slc), and fibers may be more degenerate—but the fibration itself always exists.

When given access to our theorem database, Claude arrived at the **correct** answer:

Claude's response with RAG (correct)

Yes. Such boundary pairs exist; they are pseudoelliptic surfaces.

- By Corollary 1.3 of Ascher & Bejleri (2017), the log canonical model of an irreducible slc elliptic surface $(X \rightarrow C, S + F_{\mathcal{A}})$ with $K_X + S + F_{\mathcal{A}}$ big is either the relative log canonical model (an elliptic surface over C), or a *pseudoelliptic surface* obtained by contracting the section S .
- By Corollary 6.10 of Ascher & Bejleri (2017), the section is contracted if and only if $(C, f_*F_{\mathcal{A}})$ is not a weighted stable pointed curve; concretely: $C \cong \mathbb{P}^1$ with $\sum a_i \leq 2$, or C has genus 1 with all $a_i = 0$.
- A pseudoelliptic surface admits no morphism to a curve with one-dimensional fibers: the section (which was isomorphic to the base) has been contracted to a point, so all former fibers pass through a single point and cannot be separated by a map to a curve.
- By Proposition 3.22 of Ascher et al. (2018), an explicit family is provided: for an elliptic surface over \mathbb{P}^1 with $\deg \mathcal{L} = 2$ and $\mathcal{A} > 0$, the log canonical model is a pseudoelliptic surface (section contracted), which is KSBA-stable and lies on the boundary.
- By Corollary 6.7 of Inchiostro (2018), in a stable degeneration of elliptic surface pairs, the special fiber maps to a nodal curve C' , but pseudoelliptic components map to a *point* of C' , so f does not have all fibers of dimension one.
- Properness of the moduli space of slc fibered surface pairs (Corollary 5.11 of Ascher & Bejleri (2016)) guarantees these pseudoelliptic limits lie in the closure.

E VALIDATION SET ARXIV CATEGORIES

Table 6 shows the distribution of arXiv categories across the 65 unique papers in our validation set. Out of 32 math tags on arXiv, the validation set covers 12.

Table 6: Distribution of arXiv categories in the validation set. Each cell reports counts as *primary/all*, where “primary” counts only papers whose first-listed arXiv category matches the row, and “all” also includes cross-listed papers.

arXiv Tag	Subject Area	Theorems	Papers
math.AG	Algebraic Geometry	70/75	47/50
math.CA	Classical Analysis and ODEs	26/36	9/12
math.AP	Analysis of PDEs	17/23	4/7
math.FA	Functional Analysis	-/12	-/4
math.MG	Metric Geometry	7/9	2/3
math.DG	Differential Geometry	-/9	-/2
math.NT	Number Theory	2/7	1/4
math.CT	Category Theory	-/4	-/1
math.CO	Combinatorics	1/3	1/2
math.RT	Representation Theory	-/3	-/1
nlin.SI	Exactly Solvable and Int. Sys.	2/2	1/1
math.CV	Complex Variables	-/1	-/1
math.SG	Symplectic Geometry	-/1	-/1
hep-th	High Energy Physics – Theory	-/1	-/1

Table 7: Distribution of arXiv categories in the whole dataset. Each cell reports counts as *primary/all*, where “primary” counts only papers whose first-listed arXiv category matches the row, and “all” also includes cross-listed papers.

Tag	Area	Thms	Papers	Tag	Area	Thms	Papers
math.CO	Combinatorics	727514/1107979	46929/68561	cs.CE	Computational Engineering, Finance, and Science	1425/4295	280/731
math.AG	Algebraic Geometry	761230/1097654	36350/51497	cond-mat.str-el	Strongly Correlated Electrons	580/4123	55/275
math.AP	Analysis of PDEs	791867/1018652	51176/64735	cs.DB	Databases	1521/4043	144/415
math.PR	Probability	606057/868961	38793/55971	physics.data-an	Data Analysis, Statistics and Probability	583/3747	111/541
math.NT	Number Theory	550466/752812	32235/42070	cs.PL	Programming Languages	1120/3515	83/281
math.DG	Differential Geometry	499694/721447	28901/40763	physics.class-ph	Classical Physics	1082/3288	178/459
math.RT	Representation Theory	353605/618985	15219/26364	q-fin.ST	Statistical Finance	1317/3221	197/421
math.DS	Dynamical Systems	374124/574125	21561/33646	nlin.AO	Adaptation and Self-Organizing Systems	558/3204	79/377
math.FA	Functional Analysis	333709/544882	20193/31825	q-bio.NC	Neurons and Cognition	768/3054	102/378
math-ph	Mathematical Physics	127036/533783	8604/32239	cs.PF	Performance	973/3019	107/320
math.GR	Group Theory	303021/488750	14269/22702	cond-mat.mes-hall	Mesoscale and Nanoscale Physics	231/2963	15/209
math.OC	Optimization and Control	298096/470086	28546/43871	cs.CY	Computers and Society	405/2949	69/415
math.GT	Geometric Topology	277653/441500	15348/22650	cond-mat.mtrl-sci	Materials Science	164/2924	29/271
math.RA	Rings and Algebras	202501/352448	10783/17650	q-fin.TR	Trading and Market Microstructure	1200/2919	107/269
math.AT	Algebraic Topology	200267/348076	8886/15601	stat.OT	Other Statistics	804/2794	111/339
cs.LG	Machine Learning	150507/323831	16346/33522	cs.IR	Information Retrieval	564/2488	108/387
math.QA	Quantum Algebra	171100/322466	8258/15419	physics.optics	Optics	299/2033	40/211
math.CA	Classical Analysis and ODEs	194189/309145	14237/21275	nlin.CG	Cellular Automata and Lattice Gases	229/1860	22/136
math.NA	Numerical Analysis	209943/285427	23450/30763	physics.plasm-ph	Plasma Physics	265/1731	42/187
math.LO	Logic	206528/285351	9324/12952	q-fin.GN	General Finance	391/1730	44/156
stat.ML	Machine Learning	95327/281203	10123/28137	cond-mat.soft	Soft Condensed Matter	224/1594	39/168
math.CV	Complex Variables	148649/278361	10203/16899	cond-mat.quant-gas	Quantum Gases	49/1502	6/94
math.OA	Operator Algebras	170685/267922	8033/12677	cs.MS	Mathematical Software	174/1487	55/281
math.ST	Statistics Theory	153124/260628	12979/21634	physics.chem-ph	Chemical Physics	188/1353	32/164
math.AC	Commutative Algebra	158613/253612	8762/13462	eess.IV	Image and Video Processing	349/1350	74/282
math.MG	Metric Geometry	100465/207104	6022/11402	physics.bio-ph	Biological Physics	96/1333	25/172
math.CT	Category Theory	89204/202660	4024/8610	cs.GR	Graphics	254/1289	44/189
math.SG	Symplectic Geometry	110807/191221	4864/8954	physics.geo-ph	Geophysics	79/1258	22/139
math.SP	Spectral Theory	73394/162664	4666/9798	physics.ao-ph	Atmospheric and Oceanic Physics	113/1166	28/166
math.KT	K-Theory and Homology	57905/145433	2441/5961	physics.gen-ph	General Physics	1010/1088	188/206
cs.DM	Discrete Mathematics	35079/134746	2246/8829	hep-ph	High Energy Physics – Phenomenology	31/957	6/93
cs.CC	Computational Complexity	65534/127305	4016/7808	cs.ET	Emerging Technologies	194/876	27/102
cs.DS	Data Structures and Algorithms	53579/121709	3269/7629	hep-lat	High Energy Physics – Lattice	42/831	9/72
stat.ME	Methodology	74267/117914	10991/15438	q-bio.GN	Genomics	179/811	27/125
cs.IT	Information Theory	42758/109029	3422/8646	cs.HC	Human-Computer Interaction	128/760	20/121
hep-th	High Energy Physics – Theory	17512/108924	1753/6809	q-bio.BM	Biomolecules	176/701	24/92
math.GN	General Topology	60256/102259	3223/5212	physics.med-ph	Medical Physics	117/619	24/94
cs.AI	Artificial Intelligence	24644/82598	3072/10378	astro-ph.IM	Instrumentation and Methods for Astrophysics	64/586	15/83
cs.LO	Logic in Computer Science	30251/68125	1772/3945	q-bio.CB	Cell Behavior	99/555	17/83
cs.CG	Computational Geometry	33920/59488	2904/4755	astro-ph.CO	Cosmology and Nongalactic Astrophysics	82/554	4/54
quant-ph	Quantum Physics	25484/58911	1947/4332	q-bio.TO	Tissues and Organs	64/482	12/73
eess.SY	Systems and Control	16997/52647	2549/7239	cond-mat.other	Other Condensed Matter	19/441	2/39
gr-qc	General Relativity and Quantum Cosmology	17771/43279	1125/2721	cond-mat.supr-con	Superconductivity	22/434	3/26
cs.CR	Cryptography and Security	16687/40367	2204/4419	physics.app-ph	Applied Physics	20/431	6/64
cs.GT	Computer Science and Game Theory	17909/36310	1504/3096	cs.SE	Software Engineering	147/406	30/80
nlin.SI	Exactly Solvable and Integrable Systems	8253/32897	744/2414	cond-mat	Condensed Matter	83/383	9/42
math.GM	General Mathematics	27194/29265	2224/2380	physics.hist-ph	History and Philosophy of Physics	116/380	23/54
stat.CO	Computation	9804/28120	1515/3830	cs.SD	Sound	95/373	28/74
econ.TH	Theoretical Economics	19057/26467	1582/2167	astro-ph.EP	Earth and Planetary Astrophysics	121/366	23/55
cond-mat.stat-mech	Statistical Mechanics	2321/24482	287/1834	physics.atom-ph	Atomic Physics	4/316	1/24
q-fin.MF	Mathematical Finance	14999/24160	1273/1960	cs.AR	Hardware Architecture	37/314	6/36
econ.EM	Econometrics	14189/22632	1479/2234	eess.AS	Audio and Speech Processing	7/299	3/66
stat.AP	Applications	5524/20294	1298/3778	astro-ph	Astrophysics	38/272	4/30
cs.FL	Formal Languages and Automata Theory	6655/18854	439/1192	q-bio.SC	Subcellular Processes	61/257	10/33
eess.SP	Signal Processing	10708/18805	2449/3485	cs.OH	Other Computer Science	22/246	3/22
q-bio.PE	Populations and Evolution	7196/17583	740/1743	nucl-th	Nuclear Theory	2/232	2/24
cs.DC	Distributed, Parallel, and Cluster Computing	3689/16269	371/1669	cs.MM	Multimedia	18/190	7/46
cs.SC	Symbolic Computation	5435/14312	395/1113	astro-ph.SR	Solar and Stellar Astrophysics	33/150	7/26
cs.SI	Social and Information Networks	3772/13795	516/1550	cs.DL	Digital Libraries	29/104	11/21
cs.CV	Computer Vision	3247/12628	703/2169	physics.ace-ph	Accelerator Physics	12/104	3/16
cs.MA	Multiagent Systems	1522/12284	208/1386	physics.space-ph	Space Physics	6/82	2/12
physics.flu-dyn	Fluid Dynamics	1836/11450	393/1335	hep-ex	High Energy Physics – Experiment	1/39	1/10
math.HO	History and Overview	5067/10829	427/815	q-bio.OT	Other Quantitative Biology	4/36	1/8
q-fin.CP	Computational Finance	3495/8114	465/928	cs.OS	Operating Systems	16/25	5/7
nlin.CD	Chaotic Dynamics	1715/7802	214/834	physics.pop-ph	Popular Physics	3/17	2/5
cs.NE	Neural and Evolutionary Computing	2010/7620	235/884	physics.ins-det	Instrumentation and Detectors	1/15	1/4
q-fin.RM	Risk Management	2840/7611	239/677				
q-fin.PR	Pricing of Securities	3634/7537	357/710				
cs.RO	Robotics	1972/7091	432/1192				
q-fin.PM	Portfolio Management	3474/6878	296/626				
cond-mat.dis-nn	Disordered Systems and Neural Networks	565/6180	67/504				
physics.comp-ph	Computational Physics	1202/6038	313/1149				
econ.GN	General Economics	3402/5953	446/695				
nlin.PS	Pattern Formation and Solitons	1042/5864	121/518				
physics.soc-ph	Physics and Society	1052/5639	211/819				
q-bio.QM	Quantitative Methods	1417/5506	263/775				
cs.CL	Computation and Language	1381/5300	277/767				
cs.NI	Networking and Internet Architecture	1565/5000	222/664				
q-bio.MN	Molecular Networks	1339/4549	134/408				

F MORE EXPERIMENTS

We ran our experiments with a larger set of embedders, including Multilingual-E5-Large-Instruct Wang et al. (2024), zbMath-Bert Steinfeldt & Mihaljević (2024), and KaLM-Embedding-V2.5 Zhao et al. (2025); Hu et al. (2025). The results of these experiments are listed in Table 8 and Table 9. We also tested alternative methods to vector search, including BM25 and hybrid-retrieval methods which are listed in Table 10.

Additionally, we ran an experiment with synthetic queries generated by an LLM. We took 300 theorems at random, covering 30 of the 32 math arXiv categories and had an LLM generate queries using the PDF source of the paper. The theorems were also manually verified by a co-author who is a research mathematician. The results of this experiment are listed in Table 11.

Table 8: Extended results comparing context window size in retrieval performance. Embedded without task instructions.

Model	P@1	Hit@10	Hit@20	MRR@20
Body Only				
zbMath-Bert 0.3B	0.053	0.158	0.237	0.086
Gemma 0.3B	0.184	0.474	0.539	0.290
Qwen3 0.6B	0.224	0.461	0.513	0.299
Qwen3 8B	0.224	0.553	0.632	0.313
KaLM-Embedding-V2.5	0.131	0.355	0.434	0.203
Multilingual-E5-Large-Instruct	0.171	0.395	0.500	0.237
w/ Abstract				
zbMath-Bert 0.3B	0.092	0.184	0.237	0.113
Gemma 0.3B	0.303	0.539	0.566	0.376
Qwen3 0.6B	0.224	0.487	0.526	0.297
Qwen3 8B	0.250	0.553	0.618	0.332
KaLM-Embedding-V2.5	0.197	0.474	0.513	0.272
Multilingual-E5-Large-Instruct	0.197	0.421	0.553	0.283
w/ First Section				
zbMath-Bert 0.3B	0.092	0.184	0.237	0.122
Gemma 0.3B	0.237	0.500	0.566	0.321
Qwen3 0.6B	0.263	0.526	0.645	0.352
Qwen3 8B	0.211	0.605	0.645	0.343
KaLM-Embedding-V2.5	0.184	0.447	0.474	0.272
Multilingual-E5-Large-Instruct	0.224	0.474	0.539	0.304

Table 9: Extended results comparing LLM slogans in retrieval performance. Embedded without task instructions. Body+Abstract.

Model	P@1	Hit@10	Hit@20	MRR@20
DeepSeek V3.1				
zbMath-Bert 0.3B	0.092	0.184	0.237	0.113
Gemma 0.3B	0.303	0.539	0.566	0.376
Qwen3 0.6B	0.224	0.487	0.526	0.297
Qwen3 8B	0.250	0.553	0.618	0.332
KaLM-Embedding-V2.5	0.197	0.474	0.513	0.272
Multilingual-E5-Large-Instruct	0.197	0.421	0.553	0.283
DeepSeek R1				
zbMath-Bert 0.3B	0.000	0.079	0.105	0.014
Gemma 0.3B	0.158	0.421	0.474	0.228
Qwen3 0.6B	0.132	0.316	0.408	0.199
Qwen3 8B	0.118	0.355	0.461	0.195
KaLM-Embedding-V2.5	0.066	0.289	0.461	0.140
Multilingual-E5-Large-Instruct	0.066	0.250	0.355	0.111
Gemini 3 Pro				
zbMath-Bert 0.3B	0.053	0.171	0.237	0.092
Gemma 0.3B	0.250	0.553	0.632	0.348
Qwen3 0.6B	0.211	0.487	0.539	0.290
Qwen3 8B	0.237	0.566	0.684	0.348
KaLM-Embedding-V2.5	0.197	0.434	0.553	0.265
Multilingual-E5-Large-Instruct	0.197	0.513	0.579	0.297
Claude Opus 4.5				
zbMath-Bert 0.3B	0.053	0.184	0.211	0.087
Gemma 0.3B	0.303	0.553	0.579	0.387
Qwen3 0.6B	0.224	0.539	0.605	0.325
Qwen3 8B	0.263	0.632	0.737	0.394
KaLM-Embedding-V2.5	0.197	0.421	0.526	0.271
Multilingual-E5-Large-Instruct	0.224	0.526	0.592	0.322

Table 10: Extended results comparing different retrieval approaches, including sparse, dense, and hybrid methods.

Model	P@1	Hit@10	Hit@20	MRR@20
BM25 (slogan)	0.099	0.252	0.297	0.143
BM25 (L ^A T _E X)	0.045	0.108	0.180	0.071
RRF (slogan)	0.162	0.387	0.486	0.231
RRF (L ^A T _E X)	0.018	0.117	0.126	0.045

Table 11: Extended results comparing retrieval performance on a synthetically generated query set.

Model	P@1	Hit@10	Hit@20	MRR@20
Baselines				
GPT 5.2	0.094	0.124	0.127	0.106
Gemini 3 Pro	0.187	0.261	0.298	0.216
Our Methods				
BM25	0.291	0.478	0.512	0.354
RRF	0.351	0.682	0.736	0.461
Qwen 8B	0.481	0.706	0.722	0.558

G PROMPTS

In this section we list the prompts used during slogan generation and retrieval, shown in Table 12, Table 13, and Table 14.

Table 12: Task instructions for embedders

Side	Prompt
Theorem	Instruct: Represent the given math statement for retrieving related statement by natural language query.\nQuery:
User Query	Instruct: Given a math search query, retrieve theorems mathematically equivalent to the query.\nQuery:

Table 13: Slogan prompts

Context	Prompt
Body Only	You generate summaries of math theorems based on theorem.body. Summaries are accurate and at most four sentences. Summaries are plain ASCII sentences with no Unicode. Describe the result without referencing it as 'this theorem' or similar. Avoid LaTeX and mathematical symbols; use words instead. Output only the final summary sentences, with no reasoning, explanations, or commentary. Do not restate the prompt, input fields, or instructions. Do not include proof steps, motivation, or background discussion.
Body+Abstract	You generate summaries of math theorems based on theorem.body. You also consider paper.summary in your summaries. Summaries are accurate and at most four sentences. Summaries are plain ASCII sentences with no Unicode. Describe the result without referencing it as 'this theorem' or similar. Avoid LaTeX and mathematical symbols; use words instead. Output only the final summary sentences, with no reasoning, explanations, or commentary. Do not restate the prompt, input fields, or instructions. Do not include proof steps, motivation, or background discussion.
Body+Introduction	You generate summaries of math theorems based on theorem.body. You also consider paper.summary and the first section of the paper in your summaries. Summaries are accurate and at most four sentences. Summaries are plain ASCII sentences with no Unicode. Describe the result without referencing it as 'this theorem' or similar. Avoid LaTeX and mathematical symbols; use words instead. Output only the final summary sentences, with no reasoning, explanations, or commentary. Do not restate the prompt, input fields, or instructions. Do not include proof steps, motivation, or background discussion.

Table 14: Retrieval prompts for baselines.

Context	Prompt
GPT-5.2	<pre> { ``role``: ``system``, ``content``: (``You are an assistant that MUST answer immediately.\n`` ``Do NOT ask the user questions or request permission.\n`` ``Use web search as needed, including opening the latest arXiv PDF(s) to verify statement numbering.\n`` ``If exact numbering cannot be verified for some item, still include it but mark number as 'UNVERIFIED'.\n`` ``Return ONLY the final list no preamble.''), }, { ``role``: ``user``, ``content``: (``Return a list of the top 20 most relevant math statements to the query below.\n`` ``Constraints:\n`` ``- Statements must be from arXiv papers.\n`` ``- For each item include: (1) arXiv id, (2) version used (e.g. v3), `` ``(3) statement type+number exactly as in that arXiv version (e.g. Theorem 1.2 / Lemma 3.4), `` ``(4) section name/number, (5) statement title/short descriptor, (6) 1{2 sentence relevance note.\n`` ``- Use the most recent arXiv version available.\n`` ``Query:\n`` f``{row.query}``) } </pre>
Gemini 3 Pro	<pre> SYSTEM.INSTRUCTION = (``You are an assistant that MUST answer immediately.\n`` ``Do NOT ask the user questions or request permission.\n`` ``Use Google Search grounding as needed, including opening the latest arXiv PDF(s) to verify statement numbering.\n`` ``If exact numbering cannot be verified for some item, still include it but mark number as 'UNVERIFIED'.\n`` ``Return ONLY the final list no preamble.'') user_prompt = (``Return a list of the top 20 most relevant math statements to the query below.\n`` ``Constraints:\n`` ``- Statements must be from arXiv papers.\n`` ``- For each item include: (1) arXiv id, (2) version used (e.g. v3), `` ``(3) statement type+number exactly as in that arXiv version (e.g. Theorem 1.2 / Lemma 3.4), `` ``(4) section name/number, (5) statement title/short descriptor, (6) 1{2 sentence relevance note.\n`` ``- Use the most recent arXiv version available.\n`` ``Query:\n`` f``{row.query}``) </pre>