

NaturalPRISM: A Natural Language Premise Retrieval Task for Research-level Intermediate Proof States

Harris Proctor¹, Li An Li¹, Austin LaHue², Michael Burr², Vinita Gangaram Jansari², Carlos Toxtli Hernandez², Luis David Garcia Puente¹, Benjamin E. Nye¹

¹Colorado College ²Clemson University

Abstract

As AI use progresses in research-level mathematics, automated information retrieval systems for constructing relevant prompts are increasingly important. Existing benchmarks and datasets for identifying which prior results are necessary to prove a statement do not align well with current workflows in which a natural language proof is in progress, but it is unclear what the next step(s) should be.

We introduce NaturalPRISM, a benchmark task and dataset containing 1.7 million pairs of intermediate proof states and the premise invoked in the next step of the proof. All pairs are extracted from natural language research-level mathematics papers published on arXiv, and span all 32 subject categories.

We evaluate the effect of fine-tuning retrieval models for domain-specific vs. domain-agnostic use cases, and find that the combination of sufficient training data with alignment between the training and testing distributions allows smaller specialized models to dramatically outperform larger general models.

The dataset and all associated code is made available online¹.

1 Introduction

Recent years have seen a dramatic increase in mathematicians using large language models (LLMs) to facilitate a wide range of tasks, including finding examples and counter-examples, proof generation and validation, conjecturing, and literature search.

The frontier of tasks that can be effectively supported with LLMs has been steadily increasing in both depth and breadth; in 2021, cutting-edge problem solving datasets (Hendrycks et al., 2021; Cobbe et al., 2021) primarily consisted of grade school-level problems. By 2025, tractable problems had progressed all the way to competition-level undergraduate difficulty (Luong et al., 2025).

By 2026, LLMs being credited with significant contributions to research-level mathematics has become a regular occurrence (Tsoukalas et al., 2026).

One of the modern paradigms for LLM use in mathematics is a human-in-the-loop approach in which mathematicians construct prompts that contain the necessary context for their task, evaluate the generated output, and revise the context as necessary. Automating the retrieval of relevant mathematical text that should be included in the context can assist this process and increase its efficacy.

There is a rich history of work that addresses information retrieval (IR) for different types of mathematical text, and more recently there has been increased effort in building Retrieval Augmented Generation (RAG) systems that leverage IR strategies to pull relevant text from corpora in order to improve performance on downstream tasks.

The success of these systems depend on the ability of models to accurately identify what text should be selected for retrieval, which can be evaluated by testing models against a variety of benchmark tasks that range from retrieving the proof for a theorem, to retrieving the answer for a question.

In this work, we introduce the new benchmark task and associated dataset **NaturalPRISM** (Natural language Premise Retrieval for Intermediate States in Mathematics) containing over 1.7 million query/answer pairs drawn from research-level proofs written in natural language, and investigate the extent to which model performance differs across distinct domains of mathematics.

2 Related Work

There exists a growing interest in using LLMs for autonomous and semi-autonomous mathematical discovery (Ahn et al., 2024). One of the primary areas of interest has been assessing and improving their ability to generate novel proofs of existing conjectures (Feng et al., 2026; Duc and Liberti,

¹<https://github.com/bnye-cc/aiming-rag>

2025; Onda et al., 2025). Considerable effort has also gone into processing the natural language used in mathematical writing (Drechsel et al., 2025), as well as proof and conjecture generation (Feng et al., 2026; Tafjord et al., 2021).

As the use cases for workflows leveraging LLMs move towards research-level tasks, it is increasingly important to establish general-purpose retrieval systems capable of supporting them. Currently, mathematical retrieval has adopted many of the practices of modern RAG and has been shown to improve conjecture and proof generation (Yang et al., 2025; Ju et al., 2026). Matlas (Ju et al., 2026) builds a semantic search engine over millions of mathematical statements extracted from peer-reviewed papers and textbooks, rendering each statement self-contained via additional context before embedding it for dense retrieval. MathNet (Alshammari et al., 2026) pairs a large multilingual problem set with a retrieval task and finds that augmentation helps only when the retriever surfaces structurally relevant neighbors, while Re²Math (Lyu et al., 2026) evaluates retrieval from partial research level proofs. The BRIGHT benchmark (Su et al., 2025) shows that establishing relevance can require reasoning rather than semantic matching, a paradigm in which conventional embedding-based retrievers perform poorly.

ReasonIR (Shao et al., 2025) responds by training a retriever on synthetic queries paired with some negative examples, achieving state of the art reasoning retrieval and downstream RAG gains. Recent best practices also focus on how queries and contexts are assembled. The O1 Embedder (Yan et al., 2025) generates intermediate “thoughts” for a query before retrieving, ReasonIR finds that longer rewritten queries act as a form of test-time scaling (Shao et al., 2025), and RAG+ (Wang et al., 2025) retrieves knowledge jointly with aligned application examples so the model is shown not only relevant facts, but also how to apply them.

In the mathematics domain specifically, identifying which documents are relevant to each other can be particularly challenging: judging semantic similarity often requires specialized domain knowledge, and the most important connections are often those that are less obvious (including cross-domain parallels). Work on mathematical information retrieval predates the popularity of modern RAG systems (Zanibbi et al., 2025), with notable early benchmarks such as MathIR (Zanibbi et al., 2016) from the NTCIR project (Aizawa and Kohlhase, 2021)

which introduced the task of matching formulas and expressions to text.

More recently, additional math information retrieval benchmark tasks have been introduced, including: matching proofs to useful definitions, lemmas, and theorems (Welleck et al., 2021); identifying missing premises for proofs; and retrieving answers for questions posted to public forums such as the Mathematics Stack Exchange (MSE) (Zanibbi et al., 2020; Mansouri et al., 2021, 2022). A collection of 12 retrieval tasks have been compiled into the Mathematical Information Retrieval Benchmark (Haocheng Ju, 2025) and grouped into four top-level categories: semantic statement retrieval, question-answer retrieval, premise retrieval, and formula retrieval.

The premise retrieval task category is focused on finding definitions, theorems, and lemmas that would be useful for augmenting a partially completed proof. This task closely aligns with how a mathematician might use an LLM to help facilitate incremental progress towards finishing a proof, without optimistically relying on the wholesale generation of an end-to-end proof.

Existing premise retrieval benchmarks are often built for formal languages - such as Lean4 (Yang et al., 2023), Light (Kaliszyk et al., 2017), and Isabelle (Mikuła et al., 2024) - due to the natural pairing between a partial proof state and the tactics or dependencies necessary to advance to the final proof state. These tasks are excellent for evaluating mathematical reasoning in models, but do little to illuminate a typical mathematicians use case since so few work in formal languages.

On the other hand, benchmarks for natural language premise retrieval include NaturalProofs (Welleck et al., 2021), but are limited either by their size (typically a couple thousand pairs), or by only using the theorem statement as the query without an accompanying partial proof. To fully evaluate the ability of a model to support a mathematician in incremental proof generation, we need a benchmark containing research-level partial proofs written in natural language, paired with the premise(s) necessary to advance to the next proof step.

In this work we generate a novel benchmark task of 1.7 million such pairs by scraping arXiv papers and identifying structures in the source L^AT_EX that allow us to automatically extract the query/answer pairs. Furthermore, the scale of this dataset allows us to investigate the effect of fine-tuning open-source models, as well as separate out performance

across different domains of mathematics.

3 Methods

3.1 Data Collection

The base dataset consists of every arXiv math paper published between April 2007 and the end of 2018, yielding a set of 235,314 papers. We identify theorem-like statements by extracting all text spans that occur inside $\LaTeX$ environments representing Lemmas (lemma, lem, lemm), Propositions (proposition, prop, propo), Corollaries (corollary, co, coro), Conjectures (conjecture, conj), Claims (claim), Definitions (definition, defn, defi, def), and Theorems (theorem, thm, mainthm, maintheorem). We then extract all proofs in $\LaTeX$ proof environment. Additionally, many papers contained proofs that occur outside of a $\LaTeX$ proof environment but inside a dedicated section for the proof; we capture these instances by also extracting any text span in a section containing the case-insensitive keyphrase “proof of.”

This process produced 3,809,157 distinct theorem-like statements, 1,845,449 proofs, and 108,362 proof sections as raw $\LaTeX$ strings before cleaning. In total, 3,102,293 of these strings (54%) contain instances of custom commands that are defined in the preamble of the source document (using `\def` or `\newcommand`); to improve their quality as stand-alone retrieval targets, we replace all occurrences of custom commands without positional arguments with the original $\LaTeX$.

3.2 Task and Query/Answer Extraction

To construct query/answer pairs, we first extract all proofs contained within each document. Then, we identify instances that could be decomposed into first a partial proof (the query), then a referenced premise (the answer), then the remainder of the proof (discarded) as shown in Figure 1. This was achieved by searching for places where the proof body contained a `\ref` tag that linked to a theorem-like statement elsewhere in the document. We truncate the proof after the sentence containing the reference in order to produce a partial proof, and the text of the referenced statement is taken to be the target premise.

It is relatively common for the body of a proof to contain a `\ref` tag linking to the text of the statement being proved. We filter these instances by establishing which statement is being proved in each proof section (by using the label associated

with the title of the section, or by proximity in the absence of a label), and ignoring all of the references to the statement itself.

The names of the labels being referenced carry clear signal about what the corresponding theorem-like text is: when you see `\ref{pythag_thm}`, it is no great mystery that the correct referent is “ $a^2 + b^2 = c^2$.” To avoid this leakage, we replace the `\ref` and label with the generic [TARGET] string. In cases where there are other non-target references in the proof body, we also avoid leaking clues by replacing them with generic [REF] strings.

Since each query is only allowed to have one correct target by construction, we also remove instances where the query contains multiple references, and the grammatical structure makes it ambiguous which one should be considered the target. For example, consider a sentence ending with “`\ref{a}`, `\ref{b}`, and `\ref{c}`.” Designating the referent `a` as the correct target would yield two false negatives, since retrieving either `b` or `c` should be considered equally valid.

Finally, we filter out instances with an insufficient amount of mathematical text: queries under 80 characters, premises under 30 characters, spans that are too dense with `\ref` tags, and spans that contain too little $\LaTeX$ math notation.

These rules are used to ensure that any queries in our dataset hold enough context clues to allude to the statement which is its answer, without explicitly revealing that answer. This process applied to all of our $\LaTeX$ papers created a query answer training dataset of 1,779,598 pairs. We create domain-specific subsets by splitting the pairs based on the primary publication tag of the paper in which they were published. The size of each domain is reported in Table 2.

3.3 Training

Many math retrieval benchmarks report the performance of a suite of frontier models on their task(s) (Haocheng Ju, 2025; Alshammari et al., 2026; Lyu et al., 2026). While this establishes a baseline for the difficulty of the proposed tasks and the differences in performance characteristics between models, there is less information available about the effect of fine-tuning frontier models on math retrieval tasks. There are many factors that contribute to this, but one prominent issue is the relatively small size of the preexisting query/answer datasets. The size of our dataset allows us to focus our evaluation on the effect of fine-tuning a single

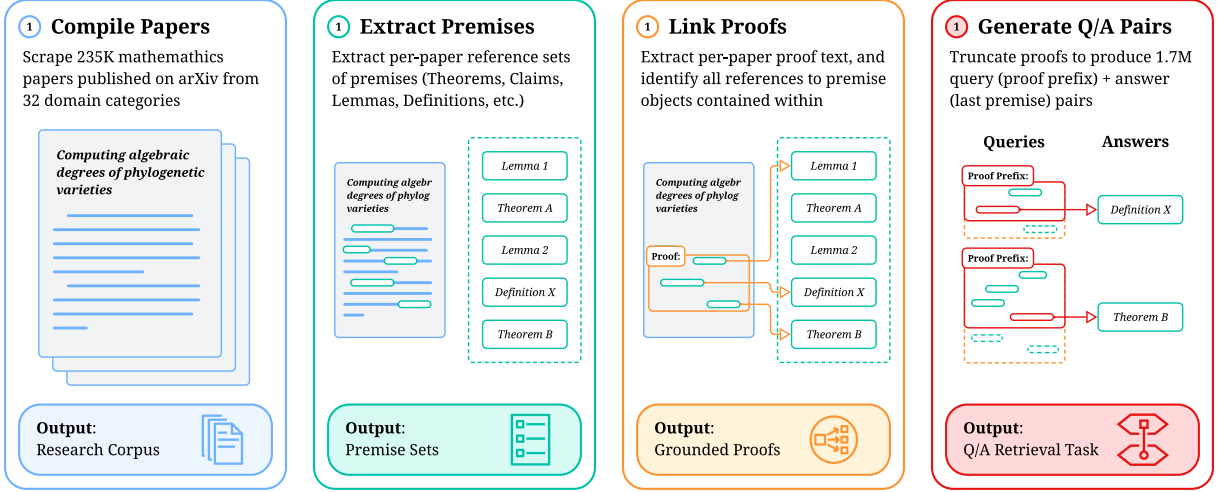


Figure 1: Overview of the Query/Answer pair generation pipeline.

Query	Answer
Suppose that $a(x)$ is a nonconstant polynomial such that every power series coefficient of $a(x)^{-1}$ is 0 or 1. Then [TARGET], $a(x) = (1 - x^m)/N(x)$ for some polynomial $N(x)$.	Let $R(x)$ be a rational function with bounded integer power series coefficients. Then $R(x) = N(x)/(1 - x^m)$ for some polynomial $N(x)$ and some positive integer m .
Apply [REF] to G with in order to get a set S with $ S \leq (k - 1)m$ such that $G \setminus S$ has a partition into components $C_1, \dots, C_t$ satisfying (i) or (ii). Without loss of generality $ C_1 \geq C_2 \geq \dots \geq C_t $. If $ C_2 + C_3 + \dots + C_t \geq m$, then we get $\text{Conn}(G \setminus S) = C_1 \leq G - m$ and so (i) holds. Therefore, we can suppose that $ C_2 + C_3 + \dots + C_t < m$ holds. In particular this means that $t \leq m - 1$, and hence part (b) of [TARGET] holds.	For every k and m , every graph G has a set $S \subseteq V(G)$ with $ S \leq (k - 1)m$ such $G \setminus S$ is partitioned into connected componets $C_1, \dots, C_t$ satisfying one of the following. 1. $t \geq m$. 2. For each $t \in \{1, \dots, t\}$, C_i is either k-connected or complete.

Table 1: Example query/answer pairs from NaturalPRISM. Queries are a partial proof state (the prefix of a complete proof up until the point of referencing another premise), and the answers are the referenced premise needed to continue the proof.

open-source base model with training sets drawn from different mathematics domains.

We select qwen3-embedding-4B as our base model. This model achieved the best performance on MathNet’s Math-Aware Retrieval Task, outperforming several proprietary models (Alshammari et al., 2026). Additionally, this model provides the ability to investigate the effect of model size by comparing against the 0.6B and 8B variants of the same model.

We fine-tune with a contrastive objective including hard negatives and batched negatives. For each query, the model embeds the query and a pool of candidate answers (the true answer, the other statements from the same paper, and the statements

from every other paper represented in the training batch) and is trained to rank the true answer above the rest. Queries are formatted with a minimal instruction prompt:

Instruct: Given a partial mathematical proof, retrieve the statement it invokes. Query: {query text}

We select six different domains to investigate; these collectively span a wide variety of mathematical topics and contain at least 80k query/answer pairs each. The chosen domains are:

For each of the selected domains, we create both a training set and an evaluation set. Additionally,

Category	Papers	QA Pairs	Category	Papers	QA Pairs
math.AC	3,629	57,918	math.IT	4,549	37,158
math.AG	12,655	212,597	math.KT	1,707	29,438
math.AP	13,929	233,850	math.LO	2,983	55,482
math.AT	3,907	60,442	math.MG	2,637	43,186
math.CA	4,970	59,619	math.MP	9,422	131,966
math.CO	14,825	206,453	math.NA	4,652	45,232
math.CT	1,963	40,203	math.NT	9,115	133,603
math.CV	3,735	50,494	math.OA	3,472	63,287
math.DG	8,882	126,917	math.OC	6,048	67,058
math.DS	7,479	114,147	math.PR	12,123	180,025
math.FA	7,775	113,992	math.QA	3,250	52,303
math.GM	246	2,528	math.RA	4,267	72,512
math.GN	1,285	19,583	math.RT	6,797	126,943
math.GR	5,675	115,260	math.SG	2,133	34,787
math.GT	5,400	81,359	math.SP	2,267	31,022
math.HO	141	1,362	math.ST	3,026	33,538

Table 2: Training pairs and papers per arXiv math category.

arXiv Tag	Category
math.AG	Algebraic Geometry
math.CO	Combinatorics
math.FA	Functional Analysis
math.GT	Geometric Topology
math.NT	Number Theory
math.PR	Probability

we consider two versions of the mixed-domain setting for baseline evaluation sets:

Random All: examples drawn randomly from all areas of math

Balanced All: examples drawn evenly from the six selected categories

Finally, we consider two additional retrieval modes that use different training data:

Random All: fine-tuned on examples drawn randomly from all areas of math

Hydra: a classifier layer first predicts which category of math the query belongs to, and then routes the retrieval task to the appropriate individually trained domain-specific fine-tuned model

4 Results

4.1 Training Dataset Size

We evaluate the retrieval performance of our models using Normalized Discounted Cumulative Gain at 10 (NDCG@10) to assess our models ranking at the top and at broader range. All evaluations

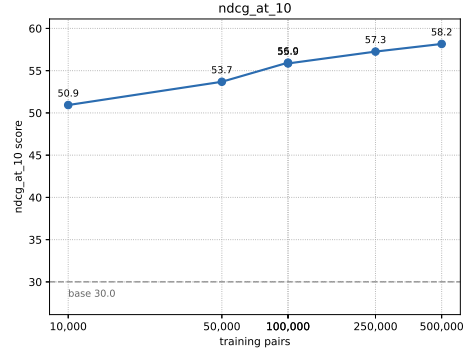


Figure 2: Model performance when trained on different numbers of example Query/Answer pairs. All experiments use the Balanced All training and evaluation sets. The dashed line indicated performance of the base model.

are done without a reranker. Fine-tuning performance of a domain-agnostic qwen3-embedding-0.6b model is shown in Figure 2. We observe that NDCG@10 performance increases steadily with training dataset sizes between 10k and 500k without any evidence of diminishing returns.

To ensure that comparisons of performance across models trained on different categories of math are fair, we must first establish a uniform size for training and evaluation datasets. The Geometric Topology (GT) category contains only 81,359 examples, and as a result we choose to limit all training datasets to 50k examples and all evaluation datasets to 20k examples. We expect that overall performance of the fine-tuned models for both evaluation metrics would increase if we collected additional data or limited ourselves to only the most prolific math categories, but since our primary focus is relative performance between different training strategies we omit these optimizations as future work.

4.2 Domain-Specific Retrieval

Performance of the different training and evaluation settings for a fine-tuned qwen3-embedding-4b model are reported in Table 3. We see that there is an enormous improvement in retrieval performance after fine-tuning across the board: NDCG@10 for the domain-agnostic evaluation (**Random All**) improves from 36.0 to 57.6, and the domain-specific versions show a similar pattern.

Both the training and the evaluation domains have a visible effect on retrieval performance. The mixed domain evaluation settings (**Balanced All** and **Random All**) appear to be easier than the

Model	ARQMath-Task-1	ProofWiki	Stacks	NaturalProofs	NaturalPRISM
<i>Open-source models ($\leq 1.5B$)</i>					
gte-large-en-v1.5	37.78	66.49	32.26	28.42	25.34
gte-Qwen2-1.5B-instruct	41.97	77.83	52.56	27.46	-
<i>Qwen3-Embedding (ours)</i>					
qwen3-embedding-4b (base)	<u>47.94</u>	73.41	50.34	33.34	35.95
ALL-4b	44.67	73.15	50.36	34.09	57.62
NT-4b	43.82	70.46	48.10	34.01	58.48
AG-4b	44.45	<u>73.83</u>	51.02	34.48	58.84
CO-4b	47.59	73.06	<u>51.72</u>	33.85	58.10
FA-4b	45.49	72.26	49.04	34.68	59.03
GT-4b	43.89	73.08	47.71	34.21	58.04
PR-4b	46.62	72.72	48.52	<u>35.73</u>	58.24
Hydra-4b	44.78	72.36	50.86	34.50	<u>59.38</u>
<i>Open-source models ($\geq 7B$)</i>					
e5-mistral-7b-instruct	46.46	77.29	39.85	32.14	33.26
NV-Embed-v2	47.34	83.08	58.56	37.21	38.23
gte-Qwen2-7B-instruct	44.74	77.02	49.35	30.08	-
SFR-Embedding-2_R	51.15	85.07	54.94	34.67	39.64
GritLM-7B	46.35	81.59	55.89	32.92	-

Table 3: Comparison with MIRB benchmarks from (Haocheng Ju, 2025). Lower performing models are omitted from the source table for space. Overall best models are in **bold**, and our best performing models are underlined.

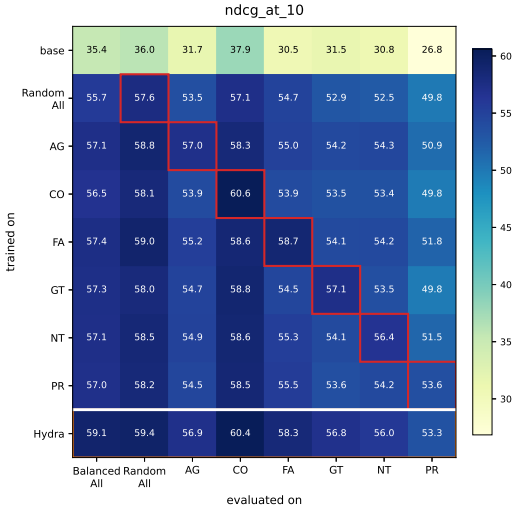


Figure 3: Multiple Fine-tunes of qwen-4B-embedder each fine-tuned on a different area of math, all evaluated on evaluation sets from each domain of math.

domain-specific settings, although this is likely due to the fact that the difficulty of retrieving the correct answer for a given query decreases when many of the candidate answers can be dismissed simply due to a category mismatch.

On the other hand, all fine-tuned models perform significantly better on retrieving Combinatorics answers than on Probability answers - including the model fine-tuned specifically on Probability queries. This demonstrates that there are domain differences in the difficulty of the retrieval task.

Performance on domain-specific evaluation sets is always maximized when using the matching training set, offering up to 8% relative improvement over the domain-agnostic training set.

True	Predicted						F1
	CO	AG	NT	GT	FA	PR	
math.CO	2996	85	142	73	49	81	0.848
math.AG	101	3154	154	102	24	8	0.852
math.NT	196	379	3013	64	46	68	0.822
math.GT	161	156	108	2805	66	25	0.869
math.FA	111	66	107	65	3322	194	0.887
math.PR	74	17	42	29	115	3189	0.907

Table 4: Confusion matrix and per-class F1 for the Hydra routing classifier (rows = true category, columns = predicted). Overall accuracy 86.4%, macro-F1 0.864.

Furthermore, the model that uses a classification layer to route queries to the appropriate domain expert (**Hydra**) achieves performance that is only marginally lower than the perfectly domain-aligned settings. As shown in Table 4, this approach achieves 0.864 F1-score on the 6-way classification task; as the size of the training set increases we expect this number to continue to improve, shrinking the performance difference between this model and the domain-aligned settings even further.

Overall fine-tuning significantly improves performance on the retrieval task, and using domain-specific models offers consistent improvement over

the domain-agnostic baseline. We compare results on our benchmark version of the task (**Random All**) against other top-performing retrieval models in Table 3; our **Hydra** model outperforms models almost twice its size by a margin of 50%.

This suggests that when sufficient training data is available for the target information retrieval task, using smaller models than can be fine-tuned with available computational resources is preferable to running inference with generalist frontier models. As reported in Section 4.1, as few as 10k training samples are sufficient to realize a large boost to model performance.

4.3 Benchmark Performance

Task	Input	Target	Domain
ARQMath	MSE Question	MSE Answer	General
ProofWiki	Theorem	Proof	General
Stacks	Theorem	Proof	Algebra, AG
NaturalProofs	Theorem	Theorem-like	Various
NaturalPRISM	Partial proof	Theorem-like	Various

Table 5: Comparison benchmark evaluation tasks from MIRB. *General* indicates that there is no domain structure; *various* indicates that there are separate distinct domains aggregated together.

To evaluate the performance of our math retrieval system on external benchmarks, we choose 4 of the 12 MIRB (Haocheng Ju, 2025) tasks that align most closely with ours: all three Question-Answer Retrieval tasks (ARQMath-Task-1, ProofWiki, and Stacks), and the one natural language Premise Retrieval task (NaturalProofs). We omit the three formal language Premise Retrieval tasks due to the mismatch between languages. A summary of the benchmark tasks is presented in Table 5.

We compute the NDCG@10 for all of our models across the four selected benchmark tasks, and report the results in Table 3. The base qwen3-embedder-4B is already a strong model, outperforming several open source models nearly twice its size. The impacts of fine-tuning are not uniform across benchmarks, and depend on how closely each task resembles our training objective and whether or not there is any alignment between the training and evaluation domains.

For the most dissimilar task, ARQMath, fine-tuning adversely affects performance across the board. For ProofWiki, which involves similar types of text objects but does not have any domain-specific structure, there are no notable improvements from fine-tuning. The Stacks task involves

exactly the same text objects as ProofWiki, but is focused specifically on Algebra and Algebraic Geometry problems. We see that our Algebraic Geometry model (and, interestingly, our Combinatorics model) outperform the baseline model.

On NaturalProofs, whose reference retrieval format is closest to our proof and statement training pairs, every fine-tuned model improves over the base model, and our PR fine-tuned model reaches 35.73, second in the table behind only the much larger NV-Embed-v2.

Model	ProofWiki	Stacks	Real Analysis	Number Theory
base	21.27	34.71	35.66	41.70
ALL	18.62	37.28	36.08	44.38
AG	18.53	37.41	34.54	47.45
CO	18.93	37.03	33.88	45.53
FA	18.72	36.87	37.08	46.07
GT	19.96	37.25	35.11	44.50
NT	18.61	37.24	35.13	45.07
PR	19.64	37.02	37.30	48.97
Hydra	18.66	37.39	36.89	45.07

Table 6: nDCG@10 on the four NaturalProofs premise-retrieval subsets of MIRB. The MIRB NaturalProofs score is the average of the four. Best single-model score per column in bold.

To further probe the effect of domain-alignment, in Table 6 we break down performance on NaturalProofs across the four different source datasets that it aggregates. Fine-tuning still offers no advantage for the general-domain ProofWiki subtask, but domain-aligned models offer improvement for the Stacks (Algebraic Geometry) and Real Analysis (Functional Analysis) subtasks. For the Number Theory subtask, the domain-aligned model improves over the base model and domain-agnostic fine-tune, but is significantly outperformed by the Probability and Algebraic Geometry models.

Overall, we find that the benefits of fine-tuning drop off as the alignment between tasks decreases. However, when the tasks are similar enough (NaturalProofs) or there is enough domain-specificity to take advantage of (Stacks), fine-tuning on domain-aligned training sets outperforms the base model.

4.4 Theorem Embeddings

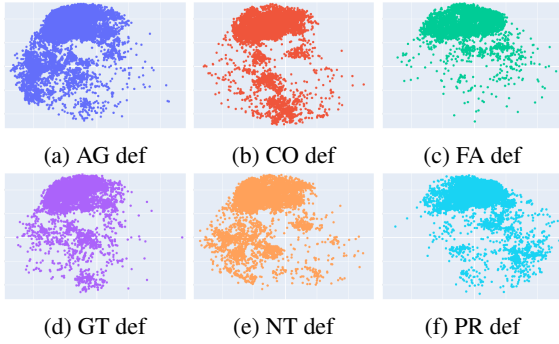


Figure 4: t-SNE projection of the embeddings of extracted definitions, separated by the domain of the source paper.

As a byproduct of this work, we also visualize the two-dimensional projections of the embedded queries and answers. This allows us to visualize some of the latent structure captured by the embedding model; in Figure 4, we see that there is a common cluster of definitions shared between domains, but smaller domain-specific clusters appear with varying proximity to each other.

We present an interactive browser that allows the user to explore and inspect the embedded objects to investigate the spatial relations between objects. The tool is available via the repository at <https://github.com/bnye-cc/aiming-rag>.

5 Conclusion

We presented a pipeline for constructing large-scale mathematical retrieval training data directly from the internal reference structure of arXiv papers. By pairing partial proofs with the statements they cite, we assembled a dataset of query answer pairs spanning 235,314 papers, requiring no synthetic query generation or manual annotation. This dataset provides a new benchmark task for evaluating the retrieval quality of models in a setting that directly aligns with an emerging human-in-the-loop workflow paradigm for how mathematicians can use LLMs to accelerate mathematical discovery.

Using this dataset, we fine-tuned Qwen3 embedding models at the 0.6B and 4B scales, both across mathematics generally and within six individual arXiv subject areas. Our experiments yield several useful findings. First, fine-tuning provides a dramatic boost in performance over the base model, but this improvement does not transfer to related Information Retrieval tasks. Second, domain specialized fine-tuning consistently improves

retrieval within the target domain without degrading cross domain performance. Our specialized models matched or exceeded the generalist fine-tune on mixed domain evaluations. Third, routing queries to domain specialists via a lightweight classifier achieved the strongest overall results, ranking first on both mixed domain evaluations and second only to the in domain specialist on every category specific evaluation.

Taken together, these results suggest that the AI mathematics community should treat different text types, and even mathematical subfields, as meaningfully distinct retrieval domains that could benefit from model specialization. Although open-source embedding models offer relatively stable performance across a wider variety of retrieval tasks, they can be outperformed by much smaller models that are fine-tuned on a specific task - and even on specific domains of math within that task.

6 Future Work

One direction that follows naturally would be to push performance of the fine-tuned model(s) on this IR task. Due to the limitations of our available resources, we restricted our experimental training datasets to 50k examples and our base model to the 4B parameter version of the qwen3-embedding model. There is clear evidence in Figure 2 that scaling the size of the training dataset will continue to enhance performance, and moving from the 0.6B model to the 4B model yielded an average relative improvement of 5.7%. Experimenting with a wider variety of higher parameter open-source models would likely provide much better absolute scores.

Another direction would be to explore the relationships between text in different domains of mathematics. In this work we simplify the data to six categories, but our method could easily be extended to cover a broader set of topics with varying amounts of training data support. Furthermore, many exciting mathematical discoveries are made when ideas are connected across different domains; we could filter for cases where the answer to a given query appear to come from another domain, and use these examples to construct explicitly cross-domain retrieval training datasets.

Finally, evaluating the impact of our retrieval models on tasks such as proof generation or conjecturing would better contextualize the value of RAG systems for natural language mathematics.

7 Limitations

The reported findings are only evaluated with respect to a single base model, qwen3-embedding-4b. Robustness of these effects with respect to different model architectures or configurations is undetermined.

The dataset itself is constructed via pattern-matching in the source $\text{\LaTeX}$, with manually designed filters to improve the quality of the samples. However, there are instances where unusual or malformed text is extracted and not removed by the filtering steps. The exact percentage is unknown, but clustering the embedded text and manually inspecting outliers has confirmed their presence.

Our answers are extracted as they appear in their source paper without additional context, and they often depend on definitions and notation defined elsewhere in the document. We take surface-level steps to mitigate this issue by unwinding document-specific $\text{\LaTeX}$ commands, but additional preprocessing strategies that bundle relevant context with theorem-like statements, such as explored by Matlas (Ju et al., 2026), could improve retrieval, particularly across papers with differing notation.

The comparison of external models on our NaturalPRISM task in Table 3 is restricted to models evaluated in the original MIRB paper that were also compatible with the FlagEmbedding evaluation framework. More recent embedding models and alternative retrieval architectures are not represented.

References

- Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. [Large language models for mathematical reasoning: Progresses and challenges](#). *Preprint*, arXiv:2402.00157.
- Akiko Aizawa and Michael Kohlhase. 2021. *Mathematical Information Retrieval*, pages 169–185. Springer Singapore, Singapore.
- Shaden Alshammari, Kevin Wen, Abrar Zainal, Mark Hamilton, Navid Safaei, Sultan Albarakati, William T. Freeman, and Antonio Torralba. 2026. [Mathnet: a global multimodal benchmark for mathematical reasoning and retrieval](#). *Preprint*, arXiv:2604.18584.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.
- Jonathan Drechsel, Anja Reusch, and Steffen Herbold. 2025. [Mamut: A novel framework for modifying mathematical formulas for the generation of specialized datasets for language model training](#). *Preprint*, arXiv:2502.20855.
- Hieu Le Duc and Leo Liberti. 2025. [Mathematics with large language models as provers and verifiers](#). *Preprint*, arXiv:2510.12829.
- Tony Feng, Trieu H. Trinh, Garrett Bingham, Dawsen Hwang, Yuri Chervonyi, Junehyuk Jung, Joonkyung Lee, Carlo Pagano, Sang hyun Kim, Federico Pasqualotto, Sergei Gukov, Jonathan N. Lee, Junsu Kim, Kaiying Hou, Golnaz Ghiasi, Yi Tay, YaGuang Li, Chenkai Kuang, Yuan Liu, and 9 others. 2026. [Towards autonomous mathematics research](#). *Preprint*, arXiv:2602.10177.
- Bin Dong Haocheng Ju. 2025. [Mirb: Mathematical information retrieval benchmark](#). *Preprint*, arXiv:2505.15585.
- 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](#). *Preprint*, arXiv:2103.03874.
- Haocheng Ju, Leheng Chen, Peihao Wu, Bryan Dai, and Bin Dong. 2026. [Matlas: A semantic search engine for mathematics](#). *Preprint*, arXiv:2604.17484.
- Cezary Kaliszyk, François Chollet, and Christian Szegedy. 2017. [Holstep: A machine learning dataset for higher-order logic theorem proving](#). *Preprint*, arXiv:1703.00426.
- Thang Luong, Dawsen Hwang, Hoang H. Nguyen, Golnaz Ghiasi, Yuri Chervonyi, Insuk Seo, Junsu Kim, Garrett Bingham, Jonathan Lee, Swaroop Mishra, Alex Zhai, Clara Huiyi Hu, Henryk Michalewski, Jimin Kim, Jeonghyun Ahn, Junhwi Bae, Xingyou Song, Trieu H. Trinh, Quoc V. Le, and Junehyuk Jung. 2025. [Towards robust mathematical reasoning](#). *Preprint*, arXiv:2511.01846.
- Zicheng Lyu, Wenjie Yang, Shengzhong Zhang, and Zengfeng Huang. 2026. [Re²math: Benchmarking theorem retrieval in research-level mathematics](#). *Preprint*, arXiv:2605.09012.
- Behrooz Mansouri, Anurag Agarwal, Douglas W. Oard, and Richard Zanibbi. 2022. Advancing Math-Aware Search: The ARQMath-3 Lab at CLEF 2022. In *Advances in Information Retrieval*, pages 408–415, Cham. Springer International Publishing.
- Behrooz Mansouri, Richard Zanibbi, Douglas W. Oard, and Anurag Agarwal. 2021. [Overview of ARQMath-2 \(2021\): Second CLEF Lab on Answer Retrieval for Questions on Math](#). In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 215–238, Cham. Springer International Publishing.

- Maciej Miłucha, Szymon Tworkowski, Szymon Antoniak, Bartosz Piotrowski, Albert Qiaochu Jiang, Jin Peng Zhou, Christian Szegedy, Łukasz Kuciński, Piotr Miłoś, and Yuhuai Wu. 2024. [Magnushammer: A transformer-based approach to premise selection](#). *Preprint*, arXiv:2303.04488.
- Naoto Onda, Kazumi Kasaura, Yuta Oriike, Masaya Taniguchi, Akiyoshi Sannai, and Sho Sonoda. 2025. [Leanconjecturer: Automatic generation of mathematical conjectures for theorem proving](#). *Preprint*, arXiv:2506.22005.
- Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muennighoff, Xi Victoria Lin, Daniela Rus, Bryan Kian Hsiang Low, Sewon Min, Wen tau Yih, Pang Wei Koh, and Luke Zettlemoyer. 2025. [Reasonir: Training retrievers for reasoning tasks](#). *Preprint*, arXiv:2504.20595.
- Hongjin Su, Howard Yen, Mengzhou Xia, Weijia Shi, Niklas Muennighoff, Han yu Wang, Haisu Liu, Quan Shi, Zachary S. Siegel, Michael Tang, Ruoxi Sun, Jinsung Yoon, Sercan O. Arik, Danqi Chen, and Tao Yu. 2025. [Bright: A realistic and challenging benchmark for reasoning-intensive retrieval](#). *Preprint*, arXiv:2407.12883.
- Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. [ProofWriter: Generating implications, proofs, and abductive statements over natural language](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3621–3634, Online. Association for Computational Linguistics.
- George Tsoukalas, Anton Kovsharov, Sergey Shirobokov, Anja Surina, Moritz Firsching, Gergely Berczi, Francisco J. R. Ruiz, Arun Suggala, Adam Zsolt Wagner, Eric Wieser, Lei Yu, Aja Huang, Miklós Z. Horváth, Andrew Ferraiuolo, Henryk Michalewski, Codrut Grosu, Thomas Hubert, Matej Balog, Pushmeet Kohli, and Swarat Chaudhuri. 2026. [Advancing mathematics research with ai-driven formal proof search](#). *Preprint*, arXiv:2605.22763.
- Yu Wang, Shiwan Zhao, Zhihu Wang, Ming Fan, Xicheng Zhang, Yubo Zhang, Zhengfan Wang, Heyuan Huang, and Ting Liu. 2025. [Rag+: Enhancing retrieval-augmented generation with application-aware reasoning](#). *Preprint*, arXiv:2506.11555.
- Sean Welleck, Jiacheng Liu, Ronan Le Bras, Hannaneh Hajishirzi, Yejin Choi, and Kyunghyun Cho. 2021. [Naturalproofs: Mathematical theorem proving in natural language](#). *Preprint*, arXiv:2104.01112.
- Ruiran Yan, Zheng Liu, and Defu Lian. 2025. [O1 embedder: Let retrievers think before action](#). *Preprint*, arXiv:2502.07555.
- Kaiyu Yang, Aidan M. Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Prenger, and Anima Anandkumar. 2023. [Leandojo: Theorem proving with retrieval-augmented language models](#). *Preprint*, arXiv:2306.15626.
- Tianbo Yang, Mingqi Yan, Hongyi Zhao, and Tianshuo Yang. 2025. [Lemmahead: Rag assisted proof generation using large language models](#). *Preprint*, arXiv:2501.15797.
- Richard Zanibbi, Akiko Aizawa, Michael Kohlhase, Iadh Ounis, Goran Topic, and Kenny Davila. 2016. [Ntcir-12 mathir task overview](#). In *NTCIR Conference on Evaluation of Information Access Technologies*.
- Richard Zanibbi, Behrooz Mansouri, and Anurag Agarwal. 2025. [Mathematical information retrieval: Search and question answering](#). *Preprint*, arXiv:2408.11646.
- Richard Zanibbi, Douglas W. Oard, Anurag Agarwal, and Behrooz Mansouri. 2020. [Overview of AR-QMath 2020: CLEF Lab on Answer Retrieval for Questions on Math](#). In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 169–193, Cham. Springer International Publishing.