SABER-MATH: Automated Benchmark for Information Retrieval Evaluation in Mathematics

Nikolay Georgiev [*1] **Maria Drencheva** [*1] **Kseniia Ibragimova** [2] **Ivo Petrov** [1] **Dimitar I. Dimitrov** [1] **Martin Vechev** [1 3]

GitHub Hugging Face

Abstract

As agentic AI systems tackle more complex mathematical tasks, they increasingly rely on information retrieval (IR) to search problem databases, theorem libraries, and educational resources. However, choosing the right retriever remains difficult, as it is infeasible to directly isolate its effect on downstream performance. On the other hand, existing retrieval-specific benchmarks often fail to capture fine-grained mathematical relevance, penalizing relevant documents. We address this gap by introducing SABER-MATH, the first fully automated benchmark for evaluating mathematical IR without expert annotation. Starting from 283K high-school-level math problems with solutions, SABER-MATH builds challenging reranking tasks in three steps: (i) first, LLMs extract concise solution summaries and mathematical topics for each problem; (ii) then, per-query relevant documents are discovered using ontology topic-based and lexical solutions-summary-based similarities, and (iii) finally, a Swiss-style LLM preference tournament produces fine-grained relevance ratings for the documents. We evaluate lexical retrievers, specialized mathematical retrieval systems, and recent embedding models. We find that while modern embedding models substantially outperform classical and math-specific baselines, even the strongest systems struggle in symbol-heavy domains like Algebra and Calculus. Importantly, we show that general-purpose IR benchmarks such as MTEB do not reliably predict mathematical performance, especially for recent embedding models, highlighting the need for math-specific retrieval benchmarks.

[*]Equal contribution [1]INSAIT, Sofia University "St. Kliment Ohridski" [2]This work was done during an internship at INSAIT, Sofia University "St. Kliment Ohridski" [3]ETH Zurich. Correspondence to: Ivo Petrov <ivo.petrov@insait.ai>.

The 3rd *AI for Math Workshop* at the 43rd International Conference on Machine Learning (ICML), Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

1. Introduction

Large language models are increasingly used in mathematical reasoning systems, serving both as natural-language assistants and components of formal reasoning pipelines (Ferrag et al., 2025; Wang et al., 2025). To improve their effectiveness and reliability, modern agentic systems often supplement proof generation with retrieval of relevant mathematical knowledge. Information retrieval (IR) (Zhu et al., 2023) helps these systems find analogous examples, relevant theorems, solution patterns, formal premises, and definitions that can guide multi-step reasoning, making retrieval a core component of mathematical AI systems.

Information retrieval for mathematics Retrieval has become important in both formal theorem proving and in informal mathematical problem solving. In formal settings, IR is used to expose relevant definitions, lemmas, declarations (Yang et al., 2023; Shen et al., 2025; Khramov et al., 2025), library code (Gao et al., 2024; Shen et al., 2025), and previously written proofs to assist a prover or autoformalizer (Liu et al., 2025). In informal settings, it is used to ground answers in trusted mathematical material and to provide analogous examples, solution schemas, or theorem-level context (Levonian et al., 2023; Yang et al., 2024). Although these systems differ in what they retrieve and how they retrieve it, they share a common requirement: retrieved documents should be useful for mathematical reasoning, rather than simply textually similar to the query. Yet, such systems rarely justify their choice of retrieval framework, making rigorous evaluation of mathematical retrieval necessary for building more effective mathematical agents.

Challenges of evaluating mathematical retrieval Evaluating mathematical retrieval is challenging because related mathematical problems are relatively infrequent, and their relevance is often not captured by surface similarity. Two problems may use different notation, wording, or terminology while relying on the same method, e.g., an extremal argument, invariant, or auxiliary construction. Conversely, problems with similar objects or formulas may require very different reasoning. Prior work (Su et al., 2025; Xiao et al., 2024; Alshammari et al., 2026) has therefore relied on ex-

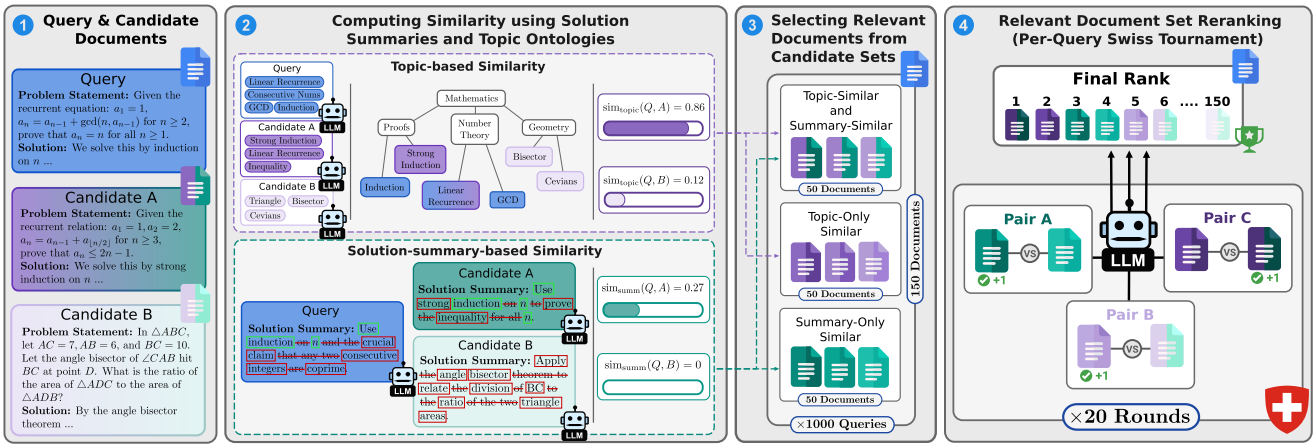

Figure 1. Overview of SABER-MATH construction. First, we source a large mathematical corpus of problems and their solutions (1). Using LLM annotation, we compute two separate relevance signals for every document pair based on topics and solution summaries (2). We then select 1000 target query documents, each with 150 relevant documents split evenly across relevance signals (3). Finally, we rank the candidates per query using an LLM-as-a-judge in a 20-round Swiss tournament based on a Bradley-Terry model (4). Throughout the figure our query documents are in blue, saturated purple denotes topic relevance, saturated teal denotes solution-summary relevance, and muted colors denote irrelevant documents.

pert annotation of semantic relevance, but such annotation is costly, hard to scale, and prone to annotator disagreement. Moreover, to keep relevance assessment tractable, many benchmarks use narrow relevance definitions and low-granularity labels (Su et al., 2025; Ju & Dong, 2025; Alshammari et al., 2026). This penalizes genuinely useful retrievals that do not match the benchmark's specific notion of relevance. All in all, existing methods thus remain hard to scale to larger corpora and diverse mathematical domains, while preserving a fine-grained notion of relevance.

SABER-Math To address this gap, we introduce **SABER-MATH**, the **S**calable **A**utomated **BE**nchmark for **R**etrieval in **Math**. To our knowledge, it is the first fully-automated benchmark designed specifically for fine-grained evaluation of mathematical information retrieval without requiring expert annotation.

Scalable relevant problem pair discovery To address the issues of rare relevant problem pairs, SABER-MATH starts from a large corpus of 283K high-school and olympiad-style problem-solution pairs and uses two scalable complementary relevance signals to find relevant problem pairs. First, we assign each problem topics in a mathematical ontology and use topic overlap to find problems involving related concepts, theorems, or domains. Second, we extract short LLM-generated summaries of each solution's core idea and use lexical overlap between summaries to find problems with similar solution methods. These signals are not used as final relevance labels. Instead, we use them only for candidate selection: we identify 1000 query problems across Algebra, Geometry, Number Theory, Combinatorics, and Calculus, such that we can pair each query with 150 candidate documents spanning different forms of mathematical relatedness.

Automated relevance estimation After constructing each query's candidate set, SABER-MATH assigns fine-grained relevance labels for each candidate using pairwise LLM judgments. To make this scalable, we avoid exhaustive all-pairs comparison and instead use a Swiss-style tournament, which prioritizes comparisons between candidates of similar estimated relevance. We aggregate the resulting preferences with a Bradley–Terry model to obtain continuous relevance scores, which serve as ground-truth ratings for our benchmark evaluation.

Experimental results We evaluate sparse lexical retrievers such as BM25 and TF-IDF, specialized mathematical retrieval frameworks such as Approach Zero (Zhong et al., 2021), and embedding-based retrievers ranging from early pretrained encoders to recent state-of-the-art models. Among all evaluated models, recent embedding models substantially outperform everything else, with the Octen open model family (Team, 2025) ranking highest for most domains, followed closely by the closed Gemini-Embedding-2 model (Lee et al., 2025). Among newer embedding models, however, we show that performance on general-purpose retrieval benchmarks such as MTEB is not a reliable predictor for mathematical performance, demonstrating the need for our work. We also provide a counterfactual analysis of models' reliance on mathematical notation, showing that weaker retrievers over-rely on symbols and formulas without capturing their meaning, which hurts their performance.

Key contributions Our contributions include:

- SABER-MATH, a scalable automated benchmark for mathematical information retrieval.
- An extensive evaluation of classical, specialized, and embedding-based retrievers, showing that general-purpose retrieval benchmarks do not fully predict mathematical IR performance.
- Domain-level and counterfactual analyses of retriever behavior, identifying weaknesses in symbolic domains.
- Publicly-available code and data to support reproducibility and further research.

2. Related Work

In this section, we discuss related work on evaluating information retrieval, its applications in mathematical reasoning, and the use of LLMs as data processors and judges.

General-purpose retrieval benchmarks General-purpose retrieval benchmarks such as BEIR (Thakur et al., 2021) and MTEB (Muennighoff et al., 2023) have become standard for comparing lexical, dense, and reranking models across domains and task types. These resources are useful for testing generalization and overall embedding quality, but they mainly measure general-purpose passage relevance: topical overlap, answer containment, or semantic rephrasing. When retrieval necessitates in-depth reasoning, it remains unclear how well general retrieval capabilities transfer to mathematical domains.

Agentic workflows with retrieval Agentic LLM systems interleave planning, tool use (Ju et al., 2026), memory (Park et al., 2023; Packer et al., 2023), and self-reflection (Madaan et al., 2023), with retrieval serving as a common tool action for grounding intermediate reasoning in external evidence or domain context (Zhu et al., 2023; Ferrag et al., 2025; Lewis et al., 2020; Nakano et al., 2021). In such workflows, retrieval quality matters, as irrelevant or incomplete context can derail subsequent reasoning, while well-targeted retrieval can support completion of harder tasks and reduce hallucinations (Wu et al., 2024; Hong et al., 2024; Yoran et al., 2024). SABER-MATH abstracts away the agentic scaffolding and directly evaluates retriever performance.

Retrieval in mathematical reasoning and theorem proving Mathematical retrieval spans a continuum from informal problem similarity to formal premise selection. In proof assistants, accurate retrieval has long been studied as premise selection for advancing the formal proof (Irving et al., 2016; Bansal et al., 2019). LLM-based provers continue this line: ReProver (Yang et al., 2023) and REAL-Prover (Shen et al., 2025) retrieve a set of premises conditioned on the formal Lean proof state code. At the same time, newer systems increasingly retrieve over natural-language mathematical statements, even when the final goal is formal reasoning. Gao et al. (2024) and Liu et al. (2025) do so by informalizing mathlib4 statements for proving and autoformalization respectively, while Archon (Ju et al., 2026) combines informal retrieval with formal search to translate informal arguments into machine-checkable developments. Beyond proof assistants, informal retrieval is also used to ground mathematical question answering, retrieve solution schemas or analogous problems, and construct mathematical knowledge graphs (Levonian et al., 2023; Dixit & Oates, 2024; Yang et al., 2024; Bian et al., 2025). However, these works typically evaluate retrieval only on downstream performance, making it difficult to isolate the quality of the retriever itself. Such measurements conflate retriever quality with the generator's reasoning ability, prior knowledge, prompt sensitivity, and robustness to irrelevant context. A model may succeed despite poor retrieval or fail despite useful retrieval, so downstream performance alone does not reliably isolate the retriever's contribution. Our work therefore focuses on informal mathematical IR as a standalone problem.

Mathematical retrieval benchmarks and relevance coverage Several recent benchmarks evaluate mathematical or reasoning-intensive retrieval more directly. BRIGHT (Su et al., 2025) evaluates mathematical relevance through a shared theorem or solving method, and MathNet (Alshammari et al., 2026) studies retrieval of mathematically equivalent or structurally similar Olympiad problems. However, these works rely on human expert annotation, which is costly and difficult to scale to larger corpora or diverse mathematical domains. Other works provide automatic pipelines, with MIRB (Ju & Dong, 2025) unifying semantic statement, premise, QA, and formula retrieval, while RAR-b (Xiao et al., 2024) converts reasoning datasets, including MATH (Hendrycks et al., 2021) and GSM8K (Cobbe et al., 2021), into retrieval over pooled answer candidates. While these benchmarks improve scalability, their construction can still lead to partial relevance coverage. Pooling-based collections leave many query-document pairs unvalidated, so relevant items surfaced by new systems may be treated as irrelevant or receive no credit, potentially introducing false negatives and affecting model comparisons (Buckley & Voorhees, 2004; Sakai & Kando, 2008). Recent work shows that missing judgments matter, but also that repairing them reliably is difficult (Upadhyay et al., 2024; Abbasiantaeb et al., 2024). This issue is especially acute in mathematics, where useful relationships are sparse and fine-grained: two problems may share a common method without being equivalent. SABER-MATH is designed with this problem

in mind, as relevance is estimated on a continuous scale, and evaluates retrievers on a task where each document has been sufficiently validated.

LLMs as data processors and judges LLMs are increasingly used to process data at scale, including annotation (Bansal & Sharma, 2023; Gilardi et al., 2023), preference estimation (Thomas et al., 2024), and evaluation (Zheng et al., 2023; Upadhyay et al., 2024; Lin et al., 2025). In particular, strong LLM judges can align well with human judgments in open-ended tasks, but also exhibit known biases and therefore require task-specific validation (Zheng et al., 2023; Ye et al., 2024; Wataoka et al., 2024; Dekoninck et al., 2024; Shi et al., 2025; Saito et al., 2023). In particular, individual pointwise scores can be noisy or non-transitive (Liu et al., 2024; Li et al., 2024), making it difficult to reliably measure subjective criteria such as relevance (Petrov et al., 2026). To address these issues, prior work has used pairwise comparisons with Bradley-Terry-style aggregation, converting many local preferences into a global ranking that can reduce noise and mitigate bias (Zheng et al., 2023; Liu et al., 2024; Dekoninck et al., 2024). In SABER-MATH, we adopt this framework to generate fine-grained relevance scores, validate it against human annotations, and show that it is preferable to standard ordinal ranking methods. We also use LLMs more broadly for data processing throughout the benchmark.

3. Methodology

In this section, we describe the construction of SABER-MATH, our benchmark for evaluating information retrieval in mathematics. SABER-MATH is designed to test whether retrieval systems can identify mathematically meaningful relevance between problems, such as shared concepts, reasoning patterns, or solution structure, rather than relying on surface-level textual similarity.

We begin by building SABER-MATH from a large corpus of high-school-level problems and solutions (§3.1). To identify related problem pairs at scale, we use two complementary automatically extracted relevance signals: topic overlap between problems (§3.2) and lexical similarity between short LLM-extracted solution summaries (§3.3). We then construct reranking tasks by selecting a set of query and candidate problems of varying types and strengths of relatedness (§3.4). Finally, we assign fine-grained relevance ratings to the candidates using an LLM-as-a-judge pairwise preference framework (§3.5), which serve as reference labels for evaluating performance in SABER-MATH. All prompts are included in §F. Additionally, we provide information about the computational resources used to construct SABER-MATH in §E.

3.1. Sourcing a Large Mathematical Corpus

Meaningful mathematical relevance is sparse: few problem pairs share non-trivial concepts, reasoning patterns, or solution structure. To identify such pairs at scale, we build a 283K-sample corpus of mostly high-school-level problems and solutions across Algebra, Geometry, Number Theory, Combinatorics, and Calculus.

The corpus contains $\sim 150K$ AoPS forum problems and community solutions, providing broad but potentially lower-quality heterogeneous coverage. We further curated $\sim 35K$ problems from national and international competitions, such as IMO, USAMO, and RMM, including higher-quality solutions with deeper structure. To reduce reliance on noisy forum data, we also include $\sim 115K$ non-synthetic problems from the olympiad split of NUMINAMATH-1.5. We deduplicate across sources using DataTrove's MinHash pipeline (Penedo et al., 2024), prioritizing official sources, NUMINAMATH-1.5, and AoPS in this order, removing $\sim 17K$ duplicates. We present further data collection and preprocessing details in §B.

3.2. Analyzing Problem Pairs using Extracted Topics

Overlapping objects, theorems, subfields and topics provide a strong signal of mathematical relevance beyond textual similarity, despite potentially different phrasing across documents. In this section, we explain how we capture this notion by constructing a MathWorld-derived ontology (Weisstein) (§3.2.1), assigning topics to problems, and computing pairwise relevance from topic sets using the hierarchy (§3.2.2).

3.2.1. Mathematical Ontology

We represent each problem by a set of mathematical topics and theorems, which provide good signals of mathematical relatedness. Because related topics may differ lexically while remaining close in the subject hierarchy, we use a fixed ontology tree rather than treating topics as a flat set.

Ontology tree construction We build our ontology from Wolfram's MathWorld (Weisstein), whose entries provide broad coverage of high-school, olympiad, and university mathematics and are linked hierarchically across mathematical areas, concepts, and theorems. We represent each MathWorld topic as a node and each hierarchical link as a directed edge from a more general topic to a more specific one, yielding a directed acyclic graph (DAG). This graph is not generally a tree: a topic can appear along multiple root-to-topic paths because mathematical concepts often belong to several areas. For example, *Pascal's Triangle* appears under both a combinatorial branch, Discrete Mathematics $\rightarrow$ Combinatorics, and a number-theoretic branch, Number Theory $\rightarrow$ Special Numbers, reflecting different uses and contexts. To preserve this infor-

mation while obtaining a tree structure, we duplicate nodes with multiple root-to-topic paths. This lets us disambiguate topic assignments by context when annotating problems from our corpus. The conversion increases the ontology size from $\sim 10.5K$ to $\sim 17.1K$ nodes, while the maximum root-to-node depth remains 5.

3.2.2. Scoring Problem Pair Relevance based on Ontologies

Automatic topic assignment To calculate problem pairs relevance signals, we need to first extract relevant ontology topics for each problem in our mathematical corpus. Due to the size of our ontology, presenting it to an LLM in full is infeasible. Instead, for each problem we do a top-down recursive traversal of the ontology. At each visited ontology node we rely on GPT-OSS-120B (Agarwal et al., 2025) to do local relevance scoring. The model is called with the problem statement, solution, and the ontology labels of the node and its children, and is asked to assign each candidate a relevance score, which we then normalize across nodes. We retain candidates whose normalized scores exceed a threshold $\tau = 0.85$ and recursively apply the same procedure to their descendants. By having the parent node as a reference, we allow the algorithm to terminate early when the children are substantially less relevant than their parent, preventing redundant computation. Finally, we keep only the most specific assigned topics by removing topic ancestors.

Topic similarity over ontologies To calculate the relevance signal for a pair of problems with annotated topics, we optimally match the two problems topic sets based on topic similarity. Our final relevance signal is given by the average similarity score within this optimal match. We measure topic similarity using Lin's information-content similarity (Lin, 1998), where two ontology nodes are more similar if their lowest common ancestor is more informative. We estimate information content from topic frequencies over our problem corpus. Concretely, let $c(t)$ be the number of annotations for node t or any of its descendants, and N be the total number of topic annotations. We define

$$p(t) = \frac{c(t)}{N}, \qquad IC(t) = -\log p(t). \tag{1}$$

For two nodes t_1 and t_2, we compute the similarity:

$$\mathrm{sim}_{\mathrm{Lin}}(t_1, t_2) = \frac{2IC(\mathrm{LCA}(t_1, t_2))}{IC(t_1) + IC(t_2) + \varepsilon}, \tag{2}$$

where $\mathrm{LCA}(t_1, t_2)$ is the nodes' lowest common ancestor in the ontology tree and $\varepsilon = 10^{-12}$ is used for numerical stability. This assigns high similarity to topics sharing a rare, specific ancestor, and low similarity to topics sharing only broad ancestors, capturing both exact and nearby ontology overlap.

Topic set matching Given the node-level similarities above, we lift them to problem-level similarity using Best-Match Average (BMA) (Wang et al., 2007). For problems x and y with topic sets $T_x = \{t_1, \ldots, t_m\}$ and $T_y = \{u_1, \ldots, u_n\}$, we define

$$\mathrm{sim}_{\mathrm{topic}}(x, y) = \frac{1}{|T_x| + |T_y|} \Bigg(\sum_{t \in T_x} \max_{u \in T_y} \mathrm{sim}_{\mathrm{Lin}}(t, u) + \sum_{u \in T_y} \max_{t \in T_x} \mathrm{sim}_{\mathrm{Lin}}(t, u) \Bigg). \tag{3}$$

BMA matches each topic to its closest counterpart in the other problem and averages these matches in both directions. This rewards broad conceptual alignment while penalizing unmatched topics, which is useful when problems combine several mathematical concepts of unequal importance. We use $\mathrm{sim}_{\mathrm{topic}}$ as the topic-based relevance signal for selecting candidate pairs in §3.4.

3.3. Scoring Problem Pair Relevance using Short Solution Summaries

While topic relevance captures an important aspect of overall mathematical relevance, problems may share underlying solution strategies despite belonging to completely different topics, or conversely, require substantially different methods even when they share topics. To capture this complementary notion of relevance, we use LLMs to extract concise natural-language summaries from problem solutions, retaining the core methodology while abstracting away formatting and presentation details. We then estimate the methodological relevance using these extracted summaries at scale via the Jaccard similarity (Jaccard, 1912).

Automatic summary extraction For each problem-solution pair in our corpus, we use GPT-OSS-120B to extract a summary of the solution's core idea. We prompt the model to produce summaries of at most 30 words in an imperative style, resulting in hint-like descriptions emphasizing the main method without reproducing the full solution, or including computational and algebraic details.

Summary-based relevance estimation For this signal, we measure the lexical overlap between the summaries of two problems using Jaccard similarity. For a problem p, let S_p denote its extracted short solution summary. Let $T(s)$ be the set of unique words appearing in a sentence s. For a pair of problems x and y, we define their summary-based relevance score as

$$\mathrm{sim}_{\mathrm{summ}}(x, y) := \frac{|T(S_x) \cap T(S_y)|}{|T(S_x) \cup T(S_y)|}. \tag{4}$$

We use lexical rather than embedding-based similarity to avoid biasing SABER-MATH toward models used in the

benchmark construction. Further, the choice of lightweight metrics, such as BMA and Jaccard also allows us to scale computation efficiently, as we require deriving the pairwise relevance signals across the entire corpus, the number of which grows quadratically ($\sim 7.8 \times 10^{10}$ total pairs). More details on the implementation can be found in §C.2.

3.4. Selecting Queries and Document Sets

We construct SABER-MATH's query and document sets from our problem corpus using our topic- and solution-summary signals. A pair is topic-relevant if $\text{sim}_{\text{topic}} > \tau_{topic}$ and summary-relevant if $\text{sim}_{\text{summ}} > \tau_{summ}$, where we choose $\tau_{topic} \approx 0.858$ and $\tau_{summ} \approx 0.211$ to obtain a representative sample spanning multiple domains. More details on how we determined these values can be found in §C.3. To ensure a balanced and challenging benchmark, we require each query to have at least $K = 50$ relevant documents in each of three categories: topic-only, summary-only, and both topic- and summary-relevant. We then sample K documents uniformly from each category, giving 150 documents per query. The final benchmark contains 1000 queries, selected to uniformly span the five top-level ontology topics: Algebra, Geometry, Number Theory, Combinatorics, and Calculus and Analysis (see Figure 5).

3.5. Tournament-based Document Relevance Ratings

While our topic- and summary-based signals provide broad evidence of mathematical relevance, they still capture only specific forms of relatedness. As a result, using them as final relevance labels in our benchmark could over-penalize other relevant connections between problems. To obtain more fine-grained benchmark ratings, we instead use a pairwise LLM-as-a-judge framework. We aggregate these judgments with a Bradley–Terry model to produce continuous relevance scores, and use a Swiss-style tournament to reduce the computation.

Pairwise preference judging with an LLM-as-a-Judge We assign each candidate its final relevance rating using an LLM-as-a-judge framework (Chan et al., 2023; Zheng et al., 2023; Liu et al., 2024; Dubois et al., 2024). Since judging mathematical relevance requires comparing solution ideas, proof structure, and techniques, we use a strong reasoning model, GPT-OSS-120B, as our judge. Rather than eliciting absolute relevance scores, we ask the judge for pairwise preferences, following prior work showing that comparative judgments are more reliable than direct scoring (Li et al., 2024). This finding is also confirmed by our human evaluation experiment in §D.1. For each query problem and its solution, we ask the judge to compare two candidate problems and their solutions and pick the one that is more mathematically relevant to the query.

Bradley–Terry estimation of latent relevance To convert judge preferences into continuous relevance scores, similarly to Dekoninck et al. (2024), we fit a Bradley-Terry (BT) model separately for each query. Under this model, the probability that candidate i is judged to be more relevant than candidate j by the LLM is assumed to be:

$$P(i > j) = \frac{\pi_i}{\pi_i + \pi_j},$$

where π_i is candidate i's latent relevance score.

Scaling pairwise comparisons with a Swiss-style tournament Exhaustive pairwise judging requires $\frac{n(n-1)}{2}$ LLM calls, which is prohibitively expensive for large reranking sets, and wastes calls on clearly unequal candidates without gaining much information. We therefore use a Swiss-style tournament to select an informative subset of comparisons:

1. **Initialization:** Candidates paired uniformly at random.
2. **Iterative matching:** In each subsequent round, each candidate is paired with an unjudged opponent whose cumulative win count is as close as possible.

The tournament concentrates judge calls on close comparisons, which are the most informative. We use $R = 20$ tournament rounds, as we find this to be sufficient in §4.3. Further details on how we run the tournament are provided in §C.4. Finally, we fit the BT model by maximizing the likelihood of the obtained pairwise judgments. These scores are linearly rescaled to the range $[0, 5]$ and used as the final relevance ratings for each query.

4. Results

In this section, we present our experimental findings. In §4.1, we describe the main results of SABER-MATH, identifying the most effective retrievers for mathematics, and analyzing different task settings across various mathematical domains. Then, in §4.2, we analyze the behaviour of different retrievers with respect to mathematical and textual content. Finally, in §4.3, we ablate the effectiveness of our Swiss-tournament, while in §4.4 we demonstrate that both of our relevance signals contribute significantly to finding relevant problems.

4.1. SABER-MATH Results and Key takeaways

We report the main SABER-MATH results in Table 1. All scores are nDCG@10 with exponential gain, using our relevance ratings as reference scores. We report both aggregate performance and performance across five mathematical domains. Details on how embedding models are run are provided in §C.1.

Unless otherwise stated, we present results in the *statement–full* setting: the query contains only a problem statement, while each document contains both a problem statement and its solution. This captures a central use case in mathematical reasoning: given a new problem, a system retrieves solved examples with relevant concepts, theorems, or solution techniques that can guide downstream reasoning (Dixit & Oates, 2024; Ju et al., 2026; Khramov et al., 2025).

In Table 1, we also evaluate two complementary settings. In *full–full* retrieval, both queries and documents include statements and solutions. This models retrieval over solved mathematical material, where a downstream system may ground answers in trusted sources, retrieve analogous examples, extract solution schemas, or construct mathematical knowledge graphs from problem-solution corpora (Levonian et al., 2023; Yang et al., 2024; Bian et al., 2025). In *statement–statement* retrieval, both queries and documents contain only problem statements. This captures search over large mathematical corpora that do not necessarily include human-readable proofs or solutions, including theorem-statement retrieval, informalized formal-library search, and retrieval over libraries of mathematical declarations (Yang et al., 2023; Shen et al., 2025; Gao et al., 2024). Comparing these settings lets us isolate how available context affects mathematical IR.

Octen-Embedding-8B is the strongest overall retriever OCTEN-EMBEDDING-8B, an open model, achieves the best aggregate performance and is also among the strongest models in all individual domains. It consistently outperforms other competitive systems, including the slightly smaller OCTEN-EMBEDDING-4B, the closed GEMINI-EMBEDDING-2 model, and the QWEN3-EMBEDDING models. We note that even the best score in the main setting remains well below perfect retrieval, leaving substantial room for progress in mathematical IR.

Modern embedding models dominate SABER-MATH Across tasks and domains, recent embedding models substantially outperform traditional lexical retrievers, such as TF-IDF, as well as older neural encoders. The specialized baseline APPROACH ZERO obtains an nDCG of 0.468 in the main setting, far below the strongest embedding models and even below most compact recent embedding models. Older contextual encoders such as BERT and ROBERTA perform particularly poorly, suggesting that mathematical retrieval systems should favor modern embedding models when available. Overall, these results indicate that dense retrieval has advanced considerably for mathematical content, while classical lexical matching and earlier math-specific retrieval pipelines are no longer competitive.

Generic retrieval benchmarks cannot solely predict performance on mathematical IR Using our main scores, we measure the Pearson and Spearman rank correlations between performance on SABER-MATH and MTEB multilingual retrieval (Muennighoff et al., 2023) as 0.77 and 0.82 respectfully. However, when restricting the analysis to recent embedding models, these correlations decrease to 0.62 and 0.73, suggesting that general benchmarks are not reliable predictors of mathematical IR performance, especially as models become more capable. For example, HARRIER-OSS-V1-27B ranks 1st among our tested models on MTEB, while being behind the Octen, Gemini, and Qwen3 models on SABER-MATH.

Retrievers struggle more on symbolic tasks For the strongest systems, Geometry and Combinatorics tend to be easier, while more symbolic-heavy domains like Algebra, Number Theory, and Calculus are more challenging. For example, OCTEN-EMBEDDING-8B achieves 0.664 on Geometry and 0.665 on Combinatorics, but only 0.594 on Algebra. This pattern is consistent with the intuition that retrievers are worse at aligning and understanding mathematical formulas vs natural language text.

Performance across domains is not uniform Although stronger retrievers generally perform well across the benchmark, domain-level rankings are not uniform. For instance, GEMINI-EMBEDDING-2 is the best-performing system on Algebra, while OCTEN-EMBEDDING-4B scores highest in Geometry and Combinatorics. GEMINI-EMBEDDING-001 performs similarly to the more recent and overall stronger GEMINI-EMBEDDING-2 on Geometry. While the top models are still close in performance, these heterogeneous differences suggest that a single aggregate score does not fully characterize a retriever's mathematical capabilities.

Stronger retrievers benefit more from additional information Comparing the three retrieval settings shows that additional context helps primarily when the retriever can represent it effectively. The top systems improve from statement-statement to statement-full, and improve further in the full-full setting. For example, OCTEN-EMBEDDING-8B increases from 0.623 to 0.636 to 0.672. In contrast, weaker systems often degrade when solution text is included. ROBERTA, for instance, scores 0.406 in statement-statement but only 0.311 in the main setting. This suggests that long text can dilute the retrieval signal unless the model is capable enough to identify and represent the relevant information. Further descriptions of the non-main settings are provided in §D.4.

Table 1. Main results of SABER-MATH. All numbers are reported as nDCG@10.

	Overall	Algebra	Geometry	Number Theory	Comb.	Calc./ Analysis	Statement-Statement	Full-Full
Octen-Embedding-8B (Team, 2025)	**0.636**	0.594	0.664	**0.629**	0.665	**0.630**	**0.623**	**0.672**
Octen-Embedding-4B (Team, 2025)	0.632	0.586	**0.673**	0.627	**0.667**	0.609	0.619	0.663
Gemini-Embedding-2 (Lee et al., 2025)	0.628	**0.599**	0.647	0.622	0.656	0.614	0.603	0.658
Qwen3-Embedding-4B (Zhang et al., 2025)	0.615	0.576	0.652	0.610	0.642	0.597	0.597	0.667
Qwen3-Embedding-8B (Zhang et al., 2025)	0.611	0.569	0.647	0.606	0.633	0.598	0.600	0.662
Harrier-OSS-v1-27b (Microsoft, 2026)	0.608	0.569	0.651	0.601	0.620	0.596	0.585	0.659
Gemini-Embedding-001 (Lee et al., 2025)	0.605	0.573	0.650	0.604	0.626	0.577	0.591	0.667
KaLM-Embedding-Gemma3-12B-2511 (Zhao et al., 2025)	0.585	0.548	0.617	0.583	0.606	0.569	0.579	0.599
LLaMa-Embed-Nemotron-8b (Babakhin et al., 2025)	0.579	0.542	0.610	0.580	0.600	0.562	0.565	0.616
Qwen3-Embedding-0.6B (Zhang et al., 2025)	0.575	0.545	0.629	0.564	0.589	0.546	0.566	0.625
Harrier-OSS-v1-0.6b (Microsoft, 2026)	0.572	0.538	0.613	0.566	0.581	0.557	0.549	0.632
Jina-Embeddings-v5-Text-Small (Akram et al., 2026)	0.570	0.525	0.620	0.561	0.593	0.549	0.554	0.610
Text-Embedding-3-Large (OpenAI, 2024)	0.558	0.535	0.571	0.560	0.574	0.552	0.539	0.593
Jina-Embeddings-v5-Text-Nano (Akram et al., 2026)	0.532	0.492	0.573	0.538	0.552	0.506	0.522	0.569
EmbeddingGemma-300m (Schechter Vera et al., 2025)	0.519	0.496	0.550	0.518	0.540	0.485	0.511	0.588
Text-Embedding-3-Small (OpenAI, 2024)	0.512	0.487	0.557	0.504	0.526	0.491	0.497	0.554
BGE-m3 (Chen et al., 2024)	0.511	0.484	0.549	0.502	0.518	0.500	0.505	0.563
Harrier-OSS-v1-270m (Microsoft, 2026)	0.498	0.470	0.545	0.491	0.512	0.469	0.497	0.557
Multilingual-E5-Large (Wang et al., 2024)	0.488	0.455	0.529	0.471	0.500	0.477	0.508	0.545
Approach Zero (Zhong et al., 2021)	0.468	0.485	0.443	0.454	0.480	0.490	0.469	0.481
TF-IDF	0.427	0.409	0.433	0.397	0.451	0.429	0.435	0.458
Jaccard (Jaccard, 1912)	0.403	0.387	0.426	0.394	0.443	0.358	0.424	0.444
BM25 (Robertson & Zaragoza, 2009)	0.398	0.394	0.420	0.381	0.422	0.368	0.409	0.410
BERT (Devlin et al., 2018)	0.357	0.369	0.342	0.344	0.389	0.335	0.417	0.429
RoBERTa (Liu et al., 2019)	0.311	0.306	0.293	0.314	0.342	0.287	0.406	0.397

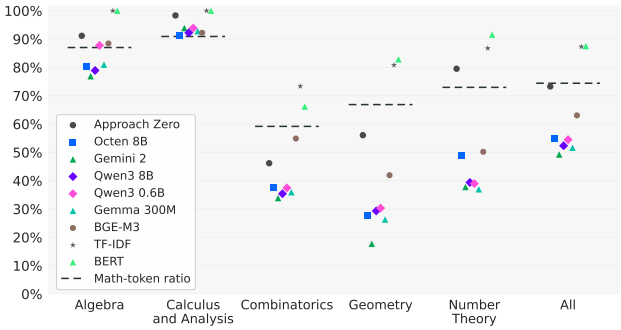

Figure 2. Distribution of retriever preferences for mathematical vs. textual content across domains, as well as the domains avg. token distribution.

4.2. Comparison of Mathematical and Textual Content in Embedding Space

We next examine what content IR methods focus on when aligning relevant pieces of mathematics. For any problem, we construct math-only and word-only variants by segmenting mathematical expressions using GPT-OSS-120B. Our prompt is presented in §F. For each retrieval system, we capture the similarity between the query's math-only and word-only representations to that of the problem-and-solution text of the five most relevant documents for each query. We then calculate the preference of a system by measuring the fraction of instances where one variant produces higher similarity than the other. Partial results are shown in Figure 2, and we include further details about the implementation in §D.2 as well as the full results in Figure 8.

Overall, we find that weaker retrievers tend to rely more on mathematical content, indicating that they do so without necessarily understanding the meaning behing it. This carries across all domains, where the systems' reliance on mathematical content is correlated to the math-token ratio, the fraction of mathematical tokens in the input. Finally, we see that models from the same family (QWEN3-8B/0.6B) show nearly identical behaviour across all domains, implying that the preferences may be inherited from the underlying training data. Results across all models are provided in §D.2.

4.3. Selecting the Number of Rounds in the Swiss Tournament

As described in §3.5, we use a Swiss-style tournament to avoid exhaustive comparisons over all $\frac{n(n-1)}{2} = 11175$ candidate pairs per query. To quantify the effect of the reduced comparisons, we compare the resulting Swiss-tournament rankings with rankings from exhaustive pairwise comparison. In Figure 3, we report the average number of preference inversions relative to the exhaustive ranking as a function of the number of rounds R.

We set $R = 20$ in the final pipeline, which is near the point where additional rounds give diminishing returns. This reduces the number of comparisons from 11175 to 1500, a factor of ~ 7.5, while keeping the ranking close to the exhaustive one, with only 700 inversions out of 11175 possible

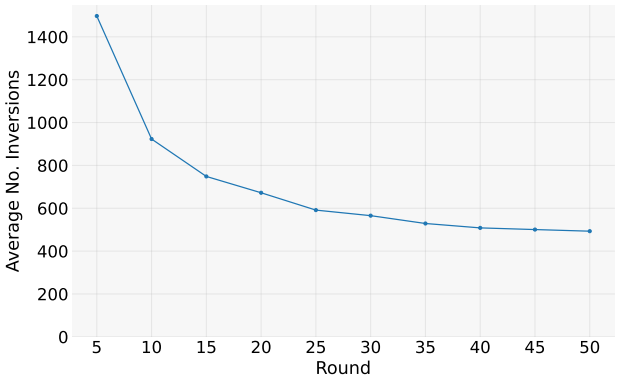

Figure 3. Average number of inversions between orderings produced by factoring all pairs compared to a Swiss tournament.

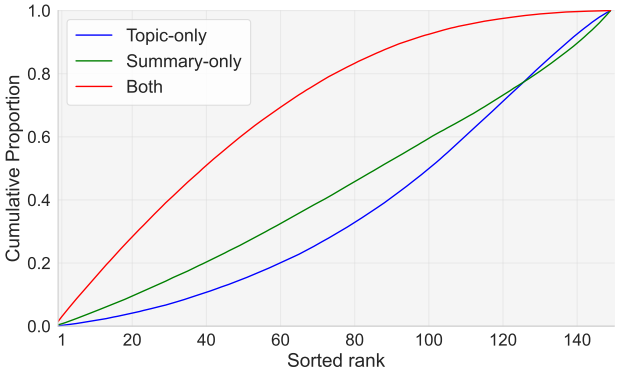

Figure 4. Cumulative rank distribution of candidates selected by different relevance signals. Higher curves indicate that candidates from that group appear earlier in the final LLM-derived ranking.

pairs. In §D.1, we further confirm through human evaluation that this setting nearly matches the exhaustive ranking in terms of human preference.

4.4. Relative Performance of Relevance Signals

In §3.4, we use two proxy signals to select candidate documents: topic-based similarity and solution-summary similarity. Since each query in SABER-MATH contains candidates sampled from three groups—topic-only, summary-only, and both—we can ask how well each group fares under the final LLM-derived relevance ranking.

Figure 4 plots, for each group, the cumulative fraction of candidates appearing within the top r ranked positions, aggregated over all 1000 queries. Thus, a faster rising curve corresponds to candidates that are placed higher by the final judge. Candidates selected by both signals rise fastest, showing that agreement between the topic and summary signals is a strong indicator of mathematical relevance. The single-signal groups are much closer to each other with summary-based similarity slightly ahead of topic similarity. Overall, the figure suggests that the two signals are complementary, each capturing separate useful relevance notion.

5. Conclusion

In this work, we introduced SABER-MATH, an automated benchmark for fine-grained evaluation of mathematical IR. We combine ontology-based topic matching, solution-summary similarity, and LLM pairwise judgments to construct realistic reranking tasks. Our results reveal that modern embedding models outperform classical and math-specific baselines, but still struggle with symbolic domains. Further, we show that general retrieval benchmarks do not reliably predict performance on mathematical retrieval tasks. Together, these results highlight the need for domain-specific evaluation and establish SABER-MATH as a valuable tool for driving progress in mathematical IR.

Acknowledgements

This research was partially funded by the Ministry of Education and Science of Bulgaria (support for INSAIT, part of the Bulgarian National Roadmap for Research Infrastructure). This project was supported with computational resources provided by Google Cloud Platform (GCP).

References

Abbasiantaeb, Z., Meng, C., Azzopardi, L., and Aliannejadi, M. Can we use large language models to fill relevance judgment holes? In Acharya, P., Clarke, C. L. A., Crestani, F., Fu, X., Jones, G. J. F., Kando, N., Kato, M. P., Lipani, A., and Liu, Y. (eds.), *Joint Proceedings of the 1st Workshop on Evaluation Methodologies,Testbeds and Community for Information Access Research (EMTCIR 2024) and the 1st Workshop on User Modelling in Conversational Information Retrieval (UMCIR 2024) co-located with the 2nd International ACM SIGIR Conference on Information Retrieval in the Asia Pacific (SIGIR-AP 2024), Tokyo, Japan, December 12, 2024*, CEUR Workshop Proceedings. CEUR-WS.org, 2024. URL https://ceur-ws.org/Vol-3854/emtcir-2.pdf.

Agarwal, S., Ahmad, L., Ai, J., Altman, S., Applebaum, A., Arbus, E., et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025. URL https://arxiv.org/abs/2508.10925.

Akram, M. K., Sturua, S., Havriushenko, N., Herreros, Q., Günther, M., Werk, M., and Xiao, H. jina-embeddings-v5-text: Task-targeted embedding distillation. *arXiv preprint arXiv:2602.15547*, 2026.

Alshammari, S. N., Wen, K., Zainal, A., Hamilton, M., Safaei, N., Albarakati, S., Freeman, W. T., and Torralba, A. Mathnet: a global multimodal benchmark for mathematical reasoning and retrieval. *CoRR*, abs/2604.18584, 2026. doi: 10.48550/ARXIV.2604.18584. URL https://doi.org/10.48550/arXiv.2604.18584.

Babakhin, Y., Osmulski, R., Ak, R., Moreira, G., Xu, M., Schifferer, B., Liu, B., and Oldridge, E. Llama-embed-nemotron-8b: A universal text embedding model for multilingual and cross-lingual tasks, 2025. URL https://arxiv.org/abs/2511.07025.

Bansal, K., Loos, S. M., Rabe, M. N., Szegedy, C., and Wilcox, S. Holist: An environment for machine learning of higher order logic theorem proving. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, Proceedings of Machine Learning Research, pp. 454–463. PMLR, 2019. URL http://proceedings.mlr.press/v97/bansal19a.html.

Bansal, P. and Sharma, A. Large language models as annotators: Enhancing generalization of NLP models at minimal cost. *CoRR*, abs/2306.15766, 2023. doi: 10.48550/ARXIV.2306.15766. URL https://doi.org/10.48550/arXiv.2306.15766.

Bian, R., Geng, Y., Yang, Z., and Cheng, B. Automathkg: The automated mathematical knowledge graph based on llm and vector database. *Computational Intelligence*, 41 (4):e70096, 2025.

Buckley, C. and Voorhees, E. M. Retrieval evaluation with incomplete information. In Sanderson, M., Järvelin, K., Allan, J., and Bruza, P. (eds.), *SIGIR 2004: Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, July 25-29, 2004*, pp. 25–32. ACM, 2004. doi: 10.1145/1008992.1009000. URL https://doi.org/10.1145/1008992.1009000.

Chan, C.-M., Chen, W., Su, Y., Yu, J., Xue, W., Zhang, S., Fu, J., and Liu, Z. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201*, 2023.

Chen, J., Xiao, S., Zhang, P., Luo, K., Lian, D., and Liu, Z. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation, 2024.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

Dekoninck, J., Baader, M., and Vechev, M. T. Polyrating: A cost-effective and bias-aware rating system for llm evaluation. *arXiv preprint arXiv:2409.00696*, 2024.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL http://arxiv.org/abs/1810.04805.

Dixit, P. and Oates, T. Sbi-rag: Enhancing math word problem solving for students through schema-based instruction and retrieval-augmented generation. *arXiv preprint arXiv:2410.13293*, 2024.

Dubois, Y., Galambosi, B., Liang, P., and Hashimoto, T. B. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

Ferrag, M. A., Tihanyi, N., and Debbah, M. From LLM reasoning to autonomous AI agents: A comprehensive review. *CoRR*, abs/2504.19678, 2025. doi: 10.48550/ARXIV.2504.19678. URL https://doi.org/10.48550/arXiv.2504.19678.

Gao, G., Ju, H., Jiang, J., Qin, Z., and Dong, B. A semantic search engine for mathlib4. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 8001–8013, 2024.

Gilardi, F., Alizadeh, M., and Kubli, M. Chatgpt outperforms crowd-workers for text-annotation tasks. *CoRR*, abs/2303.15056, 2023. doi: 10.48550/ARXIV.2303.15056. URL https://doi.org/10.48550/arXiv.2303.15056.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the MATH dataset. In Vanschoren, J. and Yeung, S. (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Hong, G., Kim, J., Kang, J., Myaeng, S., and Whang, J. J. Why so gullible? enhancing the robustness of retrieval-augmented models against counterfactual noise. In Duh, K., Gómez-Adorno, H., and Bethard, S. (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, Findings of ACL, pp. 2474–2495. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-NAACL.159. URL https://doi.org/10.18653/v1/2024.findings-naacl.159.

Irving, G., Szegedy, C., Alemi, A. A., Eén, N., Chollet, F., and Urban, J. Deepmath - deep sequence models for premise selection. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 2235–2243, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/f197002b9a0853eca5e046d9ca4663d5-Abstract.html.

Jaccard, P. The distribution of the flora in the alpine zone. *New Phytologist*, 11(2):37–50, 1912. doi: 10.1111/j.1469-8137.1912.tb05611.x.

Ju, H. and Dong, B. MIRB: mathematical information retrieval benchmark. *CoRR*, abs/2505.15585, 2025. doi: 10.48550/ARXIV.2505.15585. URL https://doi.org/10.48550/arXiv.2505.15585.

Ju, H., Gao, G., Jiang, J., Wu, B., Sun, Z., Chen, L., Wang, Y., Wang, Y., Wang, Z., He, W., Wu, P., Xiao, L., Liu, R., Dai, B., and Dong, B. Automated conjecture resolution with formal verification. *CoRR*, abs/2604.03789, 2026. doi: 10.48550/ARXIV.2604.03789. URL https://doi.org/10.48550/arXiv.2604.03789.

Khramov, N., Kozyrev, A., Solovev, G., and Podkopaev, A. Rocqstar: Leveraging similarity-driven retrieval and agentic systems for rocq generation. *CoRR*, abs/2505.22846, 2025. doi: 10.48550/ARXIV.2505.22846. URL https://doi.org/10.48550/arXiv.2505.22846.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Lee, J., Chen, F., Dua, S., Cer, D., Shanbhogue, M., Naim, I., Ábrego, G. H., Li, Z., Chen, K., Vera, H. S., Ren, X., Zhang, S., Salz, D., Boratko, M., Han, J., Chen, B., Huang, S., Rao, V., Suganthan, P., Han, F., Doumanoglou, A., Gupta, N., Moiseev, F., Yip, C., Jain, A., Baumgartner, S., Shahi, S., Gomez, F. P., Mariserla, S., Choi, M., Shah, P., Goenka, S., Chen, K., Xia, Y., Chen, K., Duddu, S. M. K., Chen, Y., Walker, T., Zhou, W., Ghiya, R., Gleicher, Z., Gill, K., Dong, Z., Seyedhosseini, M., Sung, Y., Hoffmann, R., and Duerig, T. Gemini embedding: Generalizable embeddings from gemini. *CoRR*, abs/2503.07891, 2025. doi: 10.48550/ARXIV.2503.07891. URL https://doi.org/10.48550/arXiv.2503.07891.

Levonian, Z., Li, C., Zhu, W., Gade, A., Henkel, O., Postle, M.-E., and Xing, W. Retrieval-augmented generation to improve math question-answering: Trade-offs between groundedness and human preference. *arXiv preprint arXiv:2310.03184*, 2023.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. Retrieval-augmented generation for knowledge-intensive NLP tasks. *CoRR*, abs/2005.11401, 2020. URL https://arxiv.org/abs/2005.11401.

LI, J., Beeching, E., Tunstall, L., Lipkin, B., Soletskyi, R., Huang, S. C., Rasul, K., Yu, L., Jiang, A., Shen, Z., Qin, Z., Dong, B., Zhou, L., Fleureau, Y., Lample, G., and Polu, S. Numinamath. [https://huggingface.co/datasets/AI-MO/NuminaMath-1.5](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf), 2024.

Li, T., Chiang, W.-L., Frick, E., Dunlap, L., Wu, T., Zhu, B., Gonzalez, J. E., and Stoica, I. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*, 2024. URL https://arxiv.org/abs/2406.11939.

Lin, B. Y., Deng, Y., Chandu, K. R., Ravichander, A., Pyatkin, V., Dziri, N., Bras, R. L., and Choi, Y. Wildbench:

Benchmarking llms with challenging tasks from real users in the wild. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=MKEHCx25xp.

Lin, D. An information-theoretic definition of similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pp. 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1558605568.

Liu, Q., Zheng, X., Lu, X., Cao, Q., and Yan, J. Rethinking and improving autoformalization: Towards a faithful metric and a dependency retrieval-based approach. In *The Thirteenth International Conference on Learning Representations*, 2025.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692, 2019. URL http://arxiv.org/abs/1907.11692.

Liu, Y., Zhou, H., Guo, Z., Shareghi, E., Vulic, I., Korhonen, A., and Collier, N. Aligning with human judgement: The role of pairwise preference in large language model evaluators. *arXiv preprint arXiv:2403.16950*, 2024.

Madaan, A., Tandon, N., Gupta, P., Hallinan, S., Gao, L., Wiegreffe, S., Alon, U., Dziri, N., Prabhumoye, S., Yang, Y., Gupta, S., Majumder, B. P., Hermann, K., Welleck, S., Yazdanbakhsh, A., and Clark, P. Self-refine: Iterative refinement with self-feedback. In Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/91edff07232fb1b55a505a9e9f6c0ff3-Abstract-Conference.html.

Microsoft. Harrier-oss-v1. https://huggingface.co/microsoft/harrier-oss-v1-27b, 2026.

Muennighoff, N., Tazi, N., Magne, L., and Reimers, N. MTEB: massive text embedding benchmark. In Vlachos, A. and Augenstein, I. (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2023, Dubrovnik, Croatia, May 2-6, 2023*, pp. 2006–2029. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EACL-MAIN.148. URL https://doi.org/10.18653/v1/2023.eacl-main.148.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., Jiang, X., Cobbe, K., Eloundou, T., Krueger, G., Button, K., Knight, M., Chess, B., and Schulman, J. Webgpt: Browser-assisted question-answering with human feedback. *CoRR*, abs/2112.09332, 2021. URL https://arxiv.org/abs/2112.09332.

OpenAI. text-embedding-3-large model, 2024. URL https://developers.openai.com/api/docs/models/text-embedding-3-large. Accessed: May 24, 2026.

Packer, C., Fang, V., Patil, S. G., Lin, K., Wooders, S., and Gonzalez, J. E. Memgpt: Towards llms as operating systems. *CoRR*, abs/2310.08560, 2023. doi: 10.48550/ARXIV.2310.08560. URL https://doi.org/10.48550/arXiv.2310.08560.

Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., and Bernstein, M. S. Generative agents: Interactive simulacra of human behavior. *CoRR*, abs/2304.03442, 2023. doi: 10.48550/ARXIV.2304.03442. URL https://doi.org/10.48550/arXiv.2304.03442.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., VanderPlas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.*, 12:2825–2830, 2011. doi: 10.5555/1953048.2078195. URL https://dl.acm.org/doi/10.5555/1953048.2078195.

Penedo, G., Kydlíček, H., Cappelli, A., Sasko, M., and Wolf, T. Datatrove: large scale data processing, 2024. URL https://github.com/huggingface/datatrove.

Petrov, I., Dekoninck, J., Dimitrov, D. I., and Vechev, M. Not all proofs are equal: Evaluating llm proof quality beyond correctness, 2026. URL https://arxiv.org/abs/2605.10379.

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

Robertson, S. E. and Zaragoza, H. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, 2009. doi: 10.1561/1500000019. URL https://doi.org/10.1561/1500000019.

Saito, K., Wachi, A., Wataoka, K., and Akimoto, Y. Verbosity bias in preference labeling by large language models. *CoRR*, abs/2310.10076, 2023. doi: 10.48550/ARXIV.2310.10076. URL https://doi.org/10.48550/arXiv.2310.10076.

Sakai, T. and Kando, N. On information retrieval metrics designed for evaluation with incomplete relevance assessments. *Inf. Retr.*, 11(5):447–470, 2008. doi: 10.1007/S10791-008-9059-7. URL https://doi.org/10.1007/s10791-008-9059-7.

Schechter Vera, H., Dua, S., Zhang, B., Salz, D., Mullins, R., Raghuram Panyam, S., Smoot, S., Naim, I., Zou, J., Chen, F., Cer, D., Lisak, A., Choi, M., Gonzalez, L., Sanseviero, O., Cameron, G., Ballantyne, I., Black, K., Chen, K., Wang, W., Li, Z., Martins, G., Lee, J., Sherwood, M., Ji, J., Wu, R., Zheng, J., Singh, J., Sharma, A., Sreepat, D., Jain, A., Elarabawy, A., Co, A., Doumanoglou, A., Samari, B., Hora, B., Potetz, B., Kim, D., Alfonseca, E., Moiseev, F., Han, F., Palma Gomez, F., Hernández Ábrego, G., Zhang, H., Hui, H., Han, J., Gill, K., Chen, K., Chen, K., Shanbhogue, M., Boratko, M., Suganthan, P., Duddu, S. M. K., Mariserla, S., Ariafar, S., Zhang, S., Zhang, S., Baumgartner, S., Goenka, S., Qiu, S., Dabral, T., Walker, T., Rao, V., Khawaja, W., Zhou, W., Ren, X., Xia, Y., Chen, Y., Chen, Y.-T., Dong, Z., Ding, Z., Visin, F., Liu, G., Zhang, J., Kenealy, K., Casbon, M., Kumar, R., Mesnard, T., Gleicher, Z., Brick, C., Lacombe, O., Roberts, A., Sung, Y., Hoffmann, R., Warkentin, T., Joulin, A., Duerig, T., and Seyedhosseini, M. Embeddinggemma: Powerful and lightweight text representations. 2025. URL https://arxiv.org/abs/2509.20354.

Shen, Z., Huang, N., Yang, F., Wang, Y., Gao, G., Xu, T., Jiang, J., He, W., Yang, P., Sun, M., et al. Realprover: Retrieval augmented lean prover for mathematical reasoning. *arXiv preprint arXiv:2505.20613*, 2025.

Shi, L., Ma, C., Liang, W., Diao, X., Ma, W., and Vosoughi, S. Judging the judges: A systematic study of position bias in llm-as-a-judge. In Inui, K., Sakti, S., Wang, H., Wong, D. F., Bhattacharyya, P., Banerjee, B., Ekbal, A., Chakraborty, T., and Singh, D. P. (eds.), *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, IJCNLP-AACL 2025, Mumbai, India, December 20-24, 2025*, pp. 292–314. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.ijcnlp-long.18/.

Su, H., Yen, H., Xia, M., Shi, W., Muennighoff, N., Wang, H., Liu, H., Shi, Q., Siegel, Z. S., Tang, M., Sun, R., Yoon, J., Arik, S. Ö., Chen, D., and Yu, T. BRIGHT: A realistic and challenging benchmark for reasoning-intensive retrieval. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=ykuc5q381b.

Team, O. Octen series: Optimizing embedding models to no.1 on rteb leaderboard, 2025. URL https://octen-team.github.io/octen_blog/posts/octen-rteb-first-place/.

Thakur, N., Reimers, N., Rücklé, A., Srivastava, A., and Gurevych, I. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In Vanschoren, J. and Yeung, S. (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/65b9eea6e1cc6bb9f0cd2a47751a186f-Abstract-round2.html.

Thomas, P., Spielman, S., Craswell, N., and Mitra, B. Large language models can accurately predict searcher preferences. In Yang, G. H., Wang, H., Han, S., Hauff, C., Zuccon, G., and Zhang, Y. (eds.), *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, Washington DC, USA, July 14-18, 2024*, pp. 1930–1940. ACM, 2024. doi: 10.1145/3626772.3657707. URL https://doi.org/10.1145/3626772.3657707.

Upadhyay, S., Kamalloo, E., and Lin, J. Llms can patch up missing relevance judgments in evaluation. *CoRR*, abs/2405.04727, 2024. doi: 10.48550/ARXIV.2405.04727. URL https://doi.org/10.48550/arXiv.2405.04727.

Wang, J. Z., Du, Z., Payattakool, R., Yu, P. S., and Chen, C.-F. A new method to measure the semantic similarity of go terms. *Bioinformatics*, 23(10):1274–1281, 05 2007. ISSN 1367-4803. doi: 10.1093/bioinformatics/btm087. URL https://doi.org/10.1093/bioinformatics/btm087.

Wang, L., Yang, N., Huang, X., Yang, L., Majumder, R., and Wei, F. Multilingual e5 text embeddings: A technical report. *arXiv preprint arXiv:2402.05672*, 2024.

Wang, P., Liu, T., Wang, C., Wang, Y., Yan, S., Jia, C., Liu, X., Chen, X., Xu, J., Li, Z., and Yu, Y. A survey on large language models for mathematical reasoning. *CoRR*, abs/2506.08446, 2025. doi: 10.48550/ARXIV.2506.08446. URL https://doi.org/10.48550/arXiv.2506.08446.

Wataoka, K., Takahashi, T., and Ri, R. Self-preference bias in llm-as-a-judge. *CoRR*, abs/2410.21819, 2024. doi: 10.48550/ARXIV.2410.21819. URL https://doi.org/10.48550/arXiv.2410.21819.

Weisstein, E. W. Mathworld—a wolfram web resource. https://mathworld.wolfram.com/. Accessed: 2026-05-07.

Wu, K., Wu, E., and Zou, J. Y. Clasheval: Quantifying the tug-of-war between an llm's internal prior and external evidence. In Globersons, A., Mackey, L., Belgrave, D., Fan, A., Paquet, U., Tomczak, J. M., and Zhang, C. (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/3aa291abc426d7a29fb08418c1244177-Abstract-Datasets_and_Benchmarks_Track.html.

Xiao, C., Hudson, G. T., and Moubayed, N. A. Rar-b: Reasoning as retrieval benchmark. *arXiv preprint arXiv:2404.06347*, 2024.

Yang, K., Swope, A., Gu, A., Chalamala, R., Song, P., Yu, S., Godil, S., Prenger, R. J., and Anandkumar, A. Leandojo: Theorem proving with retrieval-augmented language models. *Advances in Neural Information Processing Systems*, 36:21573–21612, 2023.

Yang, X., Lin, J., Wang, Z., and Zhai, C. Learning by analogy: Enhancing few-shot prompting for math word problem solving with computational graph-based retrieval. *CoRR*, abs/2411.16454, 2024. doi: 10.48550/ARXIV.2411.16454. URL https://doi.org/10.48550/arXiv.2411.16454.

Ye, J., Wang, Y., Huang, Y., Chen, D., Zhang, Q., Moniz, N., Gao, T., Geyer, W., Huang, C., Chen, P., Chawla, N. V., and Zhang, X. Justice or prejudice? quantifying biases in llm-as-a-judge. *CoRR*, abs/2410.02736, 2024. doi: 10.48550/ARXIV.2410.02736. URL https://doi.org/10.48550/arXiv.2410.02736.

Yoran, O., Wolfson, T., Ram, O., and Berant, J. Making retrieval-augmented language models robust to irrelevant context. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=ZS4m74kZpH.

Zhang, Y., Li, M., Long, D., Zhang, X., Lin, H., Yang, B., Xie, P., Yang, A., Liu, D., Lin, J., Huang, F., and Zhou, J. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Zhao, X., Hu, X., Shan, Z., Huang, S., Zhou, Y., Zhang, X., Sun, Z., Liu, Z., Li, D., Wei, X., Pan, Y., Xiang, Y., Zhang, M., Wang, H., Yu, J., Hu, B., and Zhang, M. Kalm-embedding-v2: Superior training techniques and data inspire a versatile embedding model, 2025. URL https://arxiv.org/abs/2506.20923.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

Zhong, W., Zhang, X., Xin, J., Zanibbi, R., and Lin, J. Approach zero and anserini at the CLEF-2021 arqmath track: Applying substructure search and BM25 on operator tree path tokens. In Faggioli, G., Ferro, N., Joly, A., Maistro, M., and Piroi, F. (eds.), *Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest, Romania, September 21st - to - 24th, 2021*, CEUR Workshop Proceedings, pp. 133–156. CEUR-WS.org, 2021. URL https://ceur-ws.org/Vol-2936/paper-09.pdf.

Zhu, Y., Yuan, H., Wang, S., Liu, J., Liu, W., Deng, C., Dou, Z., and Wen, J. Large language models for information retrieval: A survey. *CoRR*, abs/2308.07107, 2023. doi: 10.48550/ARXIV.2308.07107. URL https://doi.org/10.48550/arXiv.2308.07107.

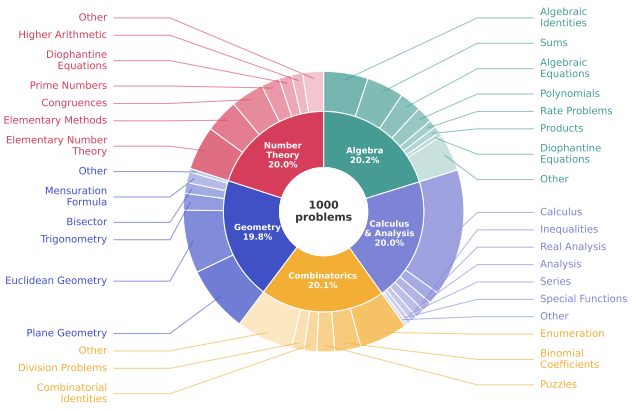

Figure 5. Domain and subdomain distribution of the query problems in SABER-MATH.

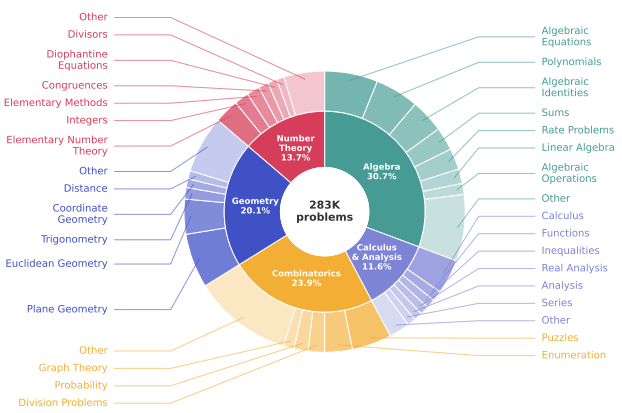

Figure 6. Domain and subdomain distribution of the sampling corpus described in §3.1.

A. Limitations

While SABER-MATH provides a scalable framework for evaluating mathematical information retrieval, several limitations remain. First, the benchmark is built primarily from high-school, olympiad, and early undergraduate-style problems, so performance may not fully transfer to research-level mathematics, formal proof libraries, textbook retrieval, or highly specialized domains. In addition, SABER-MATH evaluates reranking over preselected candidate sets rather than open-ended retrieval over the full corpus. This design makes it possible to assign fine-grained relevance judgments to every evaluated candidate, while complementary first-stage retrieval evaluation remains an important extension. Finally, the benchmark uses automated LLM-based processing to support construction at scale. Although this can introduce noise into individual relevance signals, our human validation indicates that the resulting judgments are reliable for comparing retrieval systems.

B. Data Sources

In this section we describe in details what datasets we use to compile our problem corpora from §3.1 and how we preprocess them.

B.1. Gathering Data from Public Forums

Large retrieval systems often rely on large-scale curation from public web sources. To establish a broad base for our benchmark, we first collect problems and solutions from the Art of Problem Solving (AoPS) community, which hosts a large collection of mathematical discussions. We parse nearly 500K threads and filter them to retain only those containing a problem statement and at least one relevant solution. When multiple candidate solutions are available, we use GPT-5-MINI to select the most appropriate one. Finally, we deduplicate problem statements using DataTrove's MinHash deduplication pipeline (Penedo et al., 2024), yielding 150K unique entries.

B.2. Gathering Data from High-Quality Official Sources

Although AoPS provides a large and diverse set of problems, its quality and difficulty vary substantially. Because AoPS is an open forum, extracted problems and solutions may be incomplete, noisy, or interleaved with out-of-context discussion. We therefore supplement the corpus with problems and solutions from high-quality official sources, including national and international competitions as well as preparation materials.

Starting from official PDFs manually collected from the web, we parse the documents using MathPix OCR, which is specialized for mathematical content. We then apply a multi-step LLM-based parsing pipeline using GPT-5-MINI as follows:

- (Optional) **Translation:** For documents in languages other than English, we first translate the text into English.

- **Segmentation:** We segment the parsed text into individual problems and solutions, removing extraneous information such as formatting artifacts, metadata, and irrelevant text. We do capturing problems and solutions in XML-style tags, without modifying their content. If the number of problem and solution tags do not match, we have a human manually review the document and correct the parsing.
- **Cleaning:** We apply a final cleaning step to remove any remaining formatting artifacts, problem metadata, such as the author, source, or number of the problem, and any text that is not part of the problem statement or solution.

This process yields a high-quality set of approximately 35K problems from challenging competitions, including the IMO, USAMO, and RMM. This subset is especially valuable for our benchmark because it contains a higher density of relevant pairs and more complex problems and solutions, making it more likely to challenge current embedding models.

B.3. Additional Sources

Because the AoPS subset is nearly five times larger than the official-source subset, relying too heavily on AoPS could reduce the overall quality of the corpus. We therefore further supplement the corpus with problems and solutions from another high-quality, but less challenging source, NUMINAMATH-1.5 (LI et al., 2024). To maintain a similar range of difficulty and mathematical domains, we sample approximately 115K problems from the *olympiads* split of NUMINAMATH-1.5, discarding synthetic samples, multiple-choice problems, and entries with invalid problem statements or solutions. This completes our initial corpus of approximately 300K mathematical problem-solution pairs.

B.4. Cross-Dataset Deduplication

Although the sources are collected independently, prominent competition problems may appear in multiple datasets. To avoid including duplicate problems as both queries and documents, we perform cross-source deduplication. We detect duplicates using the same MinHash-based method described in §B.1. When duplicates are found, we retain a single instance according to the following priority order: official sources, NUMINAMATH-1.5, and then AoPS. This process removes approximately 17K cross-source duplicates, leaving a final corpus of approximately 283K unique problem-solution pairs.

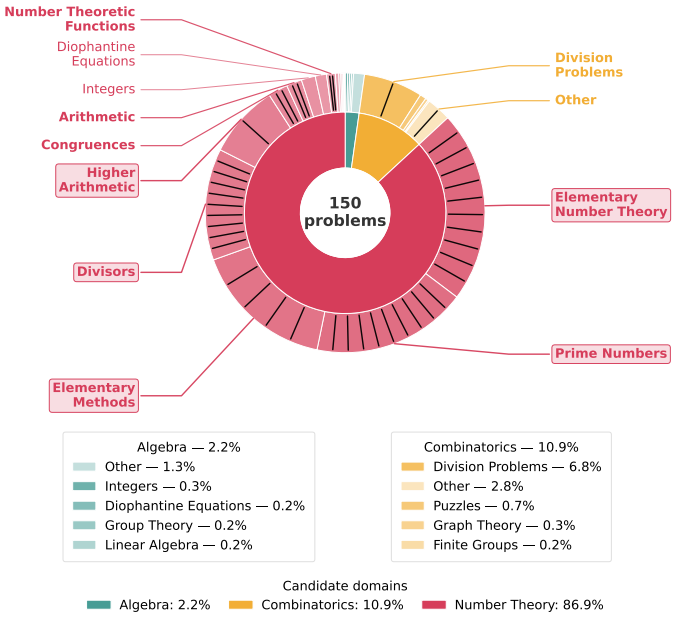

Figure 7. Domain and subdomain distribution of the documents retrieved for a representative query from SABER-MATH. Shaded subdomains indicate the subdomains associated with the query, while black lines mark documents ranked in the top 10 for the query.

B.5. Mathematical Data Distribution

In Figure 5 and Figure 6 we present visualizations of the domain and subdomain distributions of the queries in SABER-MATH and the sampling corpus respectively. The categories are determined by the mathematical ontology tagging described in §3.2.1. Since a single problem may belong to multiple domains or subdomains, each problem contributes a total weight of 1 distributed equally among its assigned categories.

B.6. Example of Query-Document Topic Distribution

To illustrate the distribution of candidate documents retrieved for a given query, Figure 7 shows the domain and subdomain distribution for a representative Number Theory query and its associated documents. The shaded subdomains correspond to those associated with the query, while the black outlines indicate the domains and subdomains of the 10 documents with the highest relevance ratings. As the figure shows, the subdomains highlighted in the query are well represented among the retrieved documents. At the same time, the pipeline also identifies relevant problems from other domains, such as

Combinatorics in this example, demonstrating that mathematical relevance is not necessarily confined to the same domain or subdomain.

B.7. Licensing

As we collate our corpus from various sources, we release our data under the Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0). This license allows others to share and adapt the material for any purpose, even commercially, as long as they provide appropriate credit, indicate if changes were made, and distribute their contributions under the same license.

All restrictions that apply to the original sources also apply to our compiled corpus. In particular, NUMINAMATH-1.5 is released under the Apache License 2.0, which allows for commercial use and modification but requires attribution and does not permit sublicensing. The AoPS content is subject to the site's terms of service, which generally allow for non-commercial use with proper attribution. The official competition problems are typically in the public domain or released under simi, with the exception of using the data for training.

C. Additional Technical Details

Here, we provide additional technical details about our implementation and experimental setup.

C.1. Running Embedding Models

Embedding models were called through their standard embedding interfaces. In particular, Gemini and OpenAI models were called through their respective embedding APIs, while local models were run with vLLM (Kwon et al., 2023), when supported, and SentenceTransformers (Reimers & Gurevych, 2019) otherwise. Output dimensionality was never specified manually, so models that support configurable dimensions used their default/full output dimension. If a piece of text exceeded the model's maximum input length, we split it into non-overlapping token chunks that fit the context window. Each chunk was embedded independently, and the final embedding was computed by elementwise averaging these embeddings.

C.2. Computing the Jaccard Similarity

We use a standard implementation of the Jaccard similarity. We first tokenize the input sentences by filtering only alphanumeric sequences, and convert them to lowercase. We filter out stop words using the Scikit-learn English stop word list (Pedregosa et al., 2011). We additionally introduce our own list of mathematics-specific stop words, that do not significantly factor into the meaning of solution summaries. The list of these words can be seen in our supplementary materials. Finally, we compute the Jaccard similarity as the size of the intersection of the two token sets divided by the size of their union.

Algorithm 1. Swiss-style greedy pairing.

Require: Query Q Documents D, rounds R
$w_d \leftarrow 0$ for all $d \in D$; $M \leftarrow \emptyset$
for $r = 1, \ldots, R$ **do**
  if $r = 1$ **then**
    $P_r \leftarrow$ random matching of D
  else
    $P_r \leftarrow \emptyset$;
      $U \leftarrow$ shuffle ties, then sort D by w_d
    while $|U| \geq 2$ **do**
      $i \leftarrow$ lowest-scoring document in U
      $F_i \leftarrow \{j \in U \setminus \{i\} : \{i, j\} \notin M\}$
      if $F_i = \emptyset$ **then**
        remove i from U
      else
        $j \leftarrow \arg\min_{k \in F_i} |w_i - w_k|$; break ties randomly
        $P_r \leftarrow P_r \cup \{\{i, j\}\}$
          $M \leftarrow M \cup \{\{i, j\}\}$
        remove i and j from U
      end if
    end while
  end if
  judge similarity of pairs in P_r to Q and increment each winner's score
end for

C.3. Selecting the Similarity Thresholds

In §3.4, we described our use of threshold-based filtering to select candidate documents for our benchmark. The thresholds were determined as follows. We first computed the distribution of similarity scores across all problem pairs in our corpus per domain. Then, for each domain, we derived a minimal threshold, for which at least 150 queries had at least 50 candidate documents above the threshold for each of the three

categories: topic-only, summary-only, and both. To determine a final single threshold for each signal, we took the minimal thresholds obtained across the five domains, resulting in $\tau_{topic} \approx 0.858$ and $\tau_{summ} \approx 0.211$.

C.4. Swiss-style Tournament Details

We used a Swiss-style pairing procedure to assign head-to-head comparisons among the 150 candidate items. In the first round, documents were paired uniformly at random. After each round, each document was assigned a running score equal to the number of times the LLM judge preferred the candidate up to that point. For all subsequent rounds, pairings were constructed greedily in ascending order of these scores.

We first randomly shuffle the documents among documents with equal scores, then sort them by their current win counts. This prevents potential biases from retaining the relative document order for documents with equal scores across the entire tournament. Starting from the lowest-scoring unpaired document, we select as its opponent an unpaired document with the closest win count among those with whom it had not previously been matched. Because this constraint can make a complete pairing infeasible in later rounds, some documents may remain unpaired in a given round. The full pairing procedure is summarized in Algorithm 1.

D. Additional Experiments and Results

Here, we display a varying range of additional results and analyses that complement our findings from the main text.

D.1. Expert Validation of Benchmark and LLM-as-a-Judge

To validate our benchmark construction, we conduct a human evaluation of both the final benchmark rankings and the reliability of our LLM-as-a-judge procedure. Two authors with experience in high-school mathematical olympiads independently annotated a random sample of 200 examples. Each example consisted of a query problem and two candidate problems, together with their corresponding solutions. Annotators were asked to decide which candidate was more mathematically relevant to the query, with the option to mark a tie when the distinction was unclear or when the two candidates appeared similarly relevant. The instructions are provided in §F.6.

Inter-annotator agreement Of the 200 examples, 20 were judged by both annotators, allowing us to estimate inter-annotator agreement, where the annotators agreed on 75.0% of the cases. This indicates that mathematical relevance judgments are somewhat subjective, especially after candidate prefiltering has removed clearly irrelevant documents. At the same time, the agreement rate suggests that the task contains substantial signal, as most disagreements arise from nuanced comparisons between plausibly related problems.

LLM-as-a-judge agreement with human annotators We next validate our pairwise LLM judge by comparing it against two alternative judging regimes: (i) an ordinal relevance judge, which assigns each query-document pair an absolute relevance score, and (ii) a five-time majority-vote variant of the pairwise judge, where ties are assigned when the vote margin is at most one. All prompts are provided in §F. To compare automatic judgments with human annotations, we assign a score of 1 for exact agreement, 0.5 when either judgment is a tie and the other expresses a preference, and 0 otherwise.

Table 2. Judge agreement rates across different document rank differences.

Method	$\Delta = 0$	$\Delta = 25$	$\Delta = 50$
Inter-annotator	75.0%	75.0%	85.7%
Pairwise Judge	**78.0%**	**82.8%**	**86.3%**
Majority Judge	77.5%	82.0%	85.6%
Ordinal Ranking	74.8%	78.9%	**86.3%**

The pairwise LLM judge achieves 78.0% agreement with human annotations, outperforming the ordinal judge, which achieves 74.8%, and slightly outperforming the majority-vote pairwise judge, which achieves 77.7%. These results provide evidence that the pairwise LLM judge is a reliable proxy for expert relevance judgments in this setting. Moreover, the stronger performance of the pairwise judge relative to the ordinal judge supports our design choice of using comparative preferences rather than absolute relevance scores.

We further analyze judge agreement as a function of the final rank difference Δ_r between the two candidates, where Δ_r is computed from the full Bradley-Terry ranking. Larger values of Δ_r correspond to pairs whose estimated relevance differs more substantially, and should therefore be easier to judge. As shown in Table 2, all judging regimes achieve higher agreement at larger Δ_r thresholds. In particular, the pairwise judge reaches 86.3% agreement for pairs with $\Delta_r \geq 50$. This trend is consistent with the intuition that close comparisons are more subjective, while large relevance gaps are easier to identify reliably. The pairwise judge remains among the strongest methods across all Δ_r thresholds, further supporting its use in our benchmark construction.

Benchmark validation Finally, we use the same human annotations to validate our Swiss-style ranking procedure. We compare the preference implied by the final Swiss-tournament rankings against the human judgments and obtain 78.0% agreement. For reference, performing full Bradley-Terry estimation over all candidate pairs yields 78.8% agreement. The small gap between these two procedures suggests that the Swiss tournament recovers most of the relevant ranking information while requiring substantially fewer LLM comparisons.

We also compare against a random tournament, where each pair being compared is selected uniformly at random at each round. Using the same number of rounds as the Swiss tournament, this random scheduling baseline achieves 75.6% agreement, below the Swiss-style procedure. This shows that Swiss-style scheduling is more effective than random comparison sampling at selecting informative pairwise judgments, and validates its use as a scalable approximation to exhaustive pairwise ranking.

D.2. Comparison of Mathematical vs. Textual Content in the Embedding Space

We now provide complemetary details and results for the experiment comparing the importance of mathematical versus textual content in the retrieval task, as described in §4.2.

Methodology and motivation To investigate whether mathematical notation or surrounding textual content contributes more to retrieving relevant documents, we conduct the following experiment using our benchmark.

For each query problem, we separately extract its mathematical expressions and its textual content. We discard queries for which either component is empty, leaving 969 of the original 1000 query problems. For each retrieval method, including both embedding-based and classical approaches, we then compute two relevance estimates for every remaining query.

First, we compare the query's mathematical content with the problem-and-solution text of each of its five highest-rated relevant candidate problems according to the benchmark. We compute the relevance score assigned by the retrieval method for each of these five candidates and take their average. We repeat the same procedure using only the query's textual content. Thus, each query receives two average relevance scores: one based on its mathematical content and one based on its textual content.

Finally, for each query and retrieval method, we record whether the mathematical or textual component receives the higher average relevance score. Since these scores are averaged real-valued quantities, ties are extremely rare and do not affect the analysis. For each retrieval method, we report the percentage of queries for which the mathematical component is judged more relevant than the textual component. The results for all evaluated embedding models and classical retrieval methods are shown in Figure 8.

For token-based methods such as Jaccard similarity and TF-IDF, appropriate tokenization is essential. We therefore tokenize mathematical expressions using the Approach Zero mathematics tokenizer.

D.3. Confidence Intervals for SABER-MATH Results

In Table 3 we include 95% confidence intervals for the results reported in §4.1, as computed by 1000-fold bootstrap resampling of the queries.

D.4. Results on SABER-MATH on Additional Tasks

We include in Table 4 the results for the non-main settings of SABER-MATH, including statement-statement and full-full retrieval. For the most part, the same trends observed in the main setting hold across these additional settings, and their respective domains. New best models emerge in some domains in the full-full setting, such as GEMINI-EMBEDDING-001

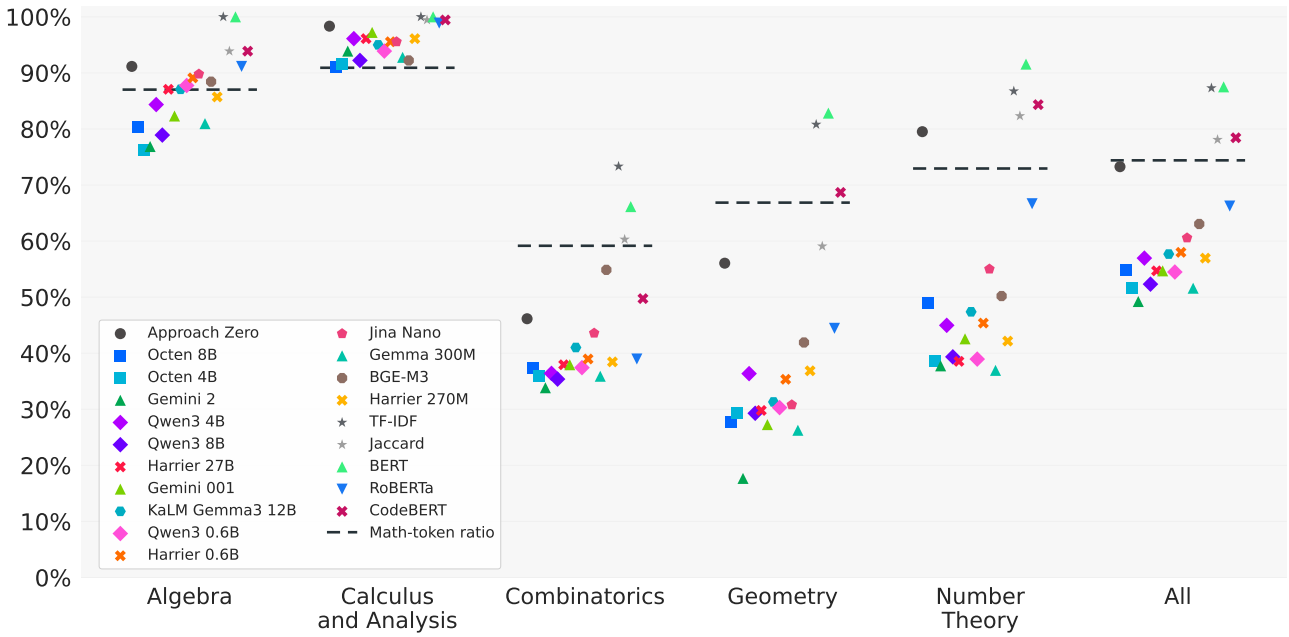

Figure 8. Percentage of query problems for which the mathematical component receives a higher average relevance score than the textual component, across retrieval methods.

Table 3. Results of SABER-MATH with 95% confidence intervals. All numbers are reported as nDCG@10.

	Overall	Algebra	Geometry	Number Theory	Comb.	Calc./ Analysis
Octen-Embedding-8B (Team, 2025)	$\mathbf{0.636^{+0.009}_{-0.010}}$	$0.594^{+0.017}_{-0.017}$	$0.664^{+0.017}_{-0.017}$	$\mathbf{0.629^{+0.017}_{-0.018}}$	$0.665^{+0.017}_{-0.018}$	$\mathbf{0.630^{+0.017}_{-0.018}}$
Octen-Embedding-4B (Team, 2025)	$0.632^{+0.009}_{-0.010}$	$0.586^{+0.017}_{-0.017}$	$\mathbf{0.673^{+0.016}_{-0.016}}$	$0.627^{+0.018}_{-0.018}$	$\mathbf{0.667^{+0.017}_{-0.017}}$	$0.609^{+0.017}_{-0.017}$
Gemini-Embedding-2 (Lee et al., 2025)	$0.628^{+0.009}_{-0.009}$	$\mathbf{0.599^{+0.016}_{-0.016}}$	$0.647^{+0.016}_{-0.016}$	$0.622^{+0.016}_{-0.016}$	$0.656^{+0.016}_{-0.017}$	$0.614^{+0.016}_{-0.016}$
Qwen3-Embedding-4B (Zhang et al., 2025)	$0.615^{+0.009}_{-0.009}$	$0.575^{+0.018}_{-0.018}$	$0.652^{+0.016}_{-0.017}$	$0.610^{+0.017}_{-0.017}$	$0.642^{+0.016}_{-0.017}$	$0.597^{+0.017}_{-0.017}$
Qwen3-Embedding-8B (Zhang et al., 2025)	$0.611^{+0.010}_{-0.010}$	$0.569^{+0.018}_{-0.017}$	$0.646^{+0.017}_{-0.017}$	$0.606^{+0.018}_{-0.017}$	$0.633^{+0.017}_{-0.017}$	$0.598^{+0.017}_{-0.017}$
Harrier-OSS-v1-27b (Microsoft, 2026)	$0.608^{+0.009}_{-0.009}$	$0.569^{+0.016}_{-0.016}$	$0.651^{+0.016}_{-0.016}$	$0.601^{+0.016}_{-0.016}$	$0.620^{+0.016}_{-0.017}$	$0.596^{+0.016}_{-0.016}$
Gemini-Embedding-001 (Lee et al., 2025)	$0.605^{+0.009}_{-0.009}$	$0.573^{+0.017}_{-0.017}$	$0.650^{+0.015}_{-0.015}$	$0.604^{+0.017}_{-0.017}$	$0.626^{+0.016}_{-0.017}$	$0.577^{+0.017}_{-0.017}$
KaLM-Embedding-Gemma3-12B-2511 (Zhao et al., 2025)	$0.585^{+0.009}_{-0.009}$	$0.548^{+0.016}_{-0.016}$	$0.617^{+0.016}_{-0.016}$	$0.583^{+0.017}_{-0.016}$	$0.606^{+0.017}_{-0.017}$	$0.569^{+0.016}_{-0.016}$
LLaMa-Embed-Nemotron-8b (Babakhin et al., 2025)	$0.579^{+0.009}_{-0.009}$	$0.542^{+0.015}_{-0.015}$	$0.610^{+0.016}_{-0.015}$	$0.580^{+0.016}_{-0.016}$	$0.600^{+0.016}_{-0.017}$	$0.562^{+0.017}_{-0.017}$
Qwen3-Embedding-0.6B (Zhang et al., 2025)	$0.575^{+0.009}_{-0.009}$	$0.545^{+0.016}_{-0.016}$	$0.629^{+0.016}_{-0.017}$	$0.564^{+0.017}_{-0.017}$	$0.589^{+0.017}_{-0.017}$	$0.546^{+0.016}_{-0.016}$
Harrier-OSS-v1-0.6b (Microsoft, 2026)	$0.572^{+0.009}_{-0.009}$	$0.538^{+0.016}_{-0.016}$	$0.613^{+0.016}_{-0.016}$	$0.566^{+0.017}_{-0.016}$	$0.581^{+0.017}_{-0.017}$	$0.557^{+0.017}_{-0.017}$
Jina-Embeddings-v5-Text-Small (Akram et al., 2026)	$0.570^{+0.009}_{-0.009}$	$0.525^{+0.016}_{-0.016}$	$0.619^{+0.017}_{-0.017}$	$0.561^{+0.017}_{-0.017}$	$0.593^{+0.017}_{-0.017}$	$0.549^{+0.015}_{-0.015}$
Text-Embedding-3-Large (OpenAI, 2024)	$0.558^{+0.009}_{-0.009}$	$0.535^{+0.016}_{-0.016}$	$0.571^{+0.015}_{-0.015}$	$0.560^{+0.016}_{-0.017}$	$0.574^{+0.016}_{-0.017}$	$0.552^{+0.016}_{-0.016}$
Jina-Embeddings-v5-Text-Nano (Akram et al., 2026)	$0.532^{+0.009}_{-0.009}$	$0.491^{+0.016}_{-0.015}$	$0.573^{+0.016}_{-0.016}$	$0.538^{+0.017}_{-0.017}$	$0.552^{+0.017}_{-0.017}$	$0.506^{+0.016}_{-0.016}$
EmbeddingGemma-300m (Schechter Vera et al., 2025)	$0.519^{+0.009}_{-0.009}$	$0.496^{+0.016}_{-0.016}$	$0.550^{+0.016}_{-0.016}$	$0.519^{+0.017}_{-0.017}$	$0.540^{+0.017}_{-0.017}$	$0.485^{+0.016}_{-0.016}$
Text-Embedding-3-Small (OpenAI, 2024)	$0.513^{+0.009}_{-0.009}$	$0.487^{+0.016}_{-0.016}$	$0.557^{+0.016}_{-0.016}$	$0.505^{+0.016}_{-0.016}$	$0.526^{+0.017}_{-0.017}$	$0.491^{+0.016}_{-0.016}$
BGE-m3 (Chen et al., 2024)	$0.512^{+0.009}_{-0.009}$	$0.484^{+0.017}_{-0.016}$	$0.549^{+0.017}_{-0.016}$	$0.503^{+0.018}_{-0.018}$	$0.518^{+0.017}_{-0.017}$	$0.500^{+0.014}_{-0.015}$
Harrier-OSS-v1-270m (Microsoft, 2026)	$0.498^{+0.009}_{-0.009}$	$0.470^{+0.016}_{-0.016}$	$0.545^{+0.016}_{-0.016}$	$0.491^{+0.017}_{-0.017}$	$0.512^{+0.016}_{-0.016}$	$0.468^{+0.015}_{-0.014}$
Multilingual-E5-Large (Wang et al., 2024)	$0.488^{+0.009}_{-0.009}$	$0.455^{+0.015}_{-0.015}$	$0.529^{+0.017}_{-0.017}$	$0.471^{+0.016}_{-0.016}$	$0.500^{+0.016}_{-0.016}$	$0.477^{+0.014}_{-0.014}$
Approach Zero (Zhong et al., 2021)	$0.446^{+0.009}_{-0.009}$	$0.452^{+0.016}_{-0.015}$	$0.430^{+0.016}_{-0.015}$	$0.436^{+0.017}_{-0.017}$	$0.461^{+0.017}_{-0.017}$	$0.450^{+0.015}_{-0.015}$
TF-IDF	$0.430^{+0.008}_{-0.008}$	$0.416^{+0.014}_{-0.014}$	$0.430^{+0.015}_{-0.015}$	$0.399^{+0.014}_{-0.014}$	$0.459^{+0.016}_{-0.016}$	$0.430^{+0.014}_{-0.014}$
Jaccard (Jaccard, 1912)	$0.404^{+0.008}_{-0.008}$	$0.387^{+0.015}_{-0.015}$	$0.426^{+0.017}_{-0.017}$	$0.394^{+0.015}_{-0.015}$	$0.443^{+0.016}_{-0.016}$	$0.357^{+0.012}_{-0.013}$
BM25 (Robertson & Zaragoza, 2009)	$0.398^{+0.008}_{-0.008}$	$0.394^{+0.015}_{-0.015}$	$0.420^{+0.016}_{-0.016}$	$0.382^{+0.014}_{-0.015}$	$0.422^{+0.016}_{-0.016}$	$0.368^{+0.013}_{-0.013}$
BERT (Devlin et al., 2018)	$0.357^{+0.008}_{-0.007}$	$0.369^{+0.014}_{-0.013}$	$0.342^{+0.014}_{-0.014}$	$0.345^{+0.013}_{-0.013}$	$0.389^{+0.016}_{-0.015}$	$0.335^{+0.012}_{-0.012}$
RoBERTa (Liu et al., 2019)	$0.311^{+0.007}_{-0.007}$	$0.306^{+0.011}_{-0.011}$	$0.293^{+0.014}_{-0.013}$	$0.314^{+0.013}_{-0.012}$	$0.342^{+0.013}_{-0.013}$	$0.287^{+0.011}_{-0.010}$

in Number Theory, and HARRIER-OSS-V1-27B in Geometry. However, the overall ranking of models remains relatively consistent across settings, with the strongest models in the main setting generally remaining strong in the additional settings.

Table 4. Results across the non-main settings of SABER-MATH. All numbers are reported as nDCG@10.

	Statement–Statement						Full–Full					
	Overall	Algebra	Geometry	Number Theory	Comb.	Calc./ Analysis	Overall	Algebra	Geometry	Number Theory	Comb.	Calc./ Analysis
Octen-Embedding-8B (Team, 2025)	**0.623**	**0.586**	0.648	**0.624**	**0.654**	**0.606**	**0.672**	**0.631**	0.685	0.670	**0.700**	**0.673**
Octen-Embedding-4B (Team, 2025)	0.619	0.581	**0.659**	0.615	0.651	0.591	0.663	0.616	0.695	0.658	0.695	0.653
Gemini-Embedding-2 (Lee et al., 2025)	0.603	0.569	0.623	0.602	0.630	0.587	0.658	0.630	0.665	0.659	0.675	0.652
Qwen3-Embedding-8B (Zhang et al., 2025)	0.600	0.562	0.643	0.600	0.622	0.569	0.662	0.621	0.693	0.660	0.679	0.657
Qwen3-Embedding-4B (Zhang et al., 2025)	0.597	0.554	0.638	0.591	0.626	0.573	0.667	0.624	0.693	0.670	0.693	0.656
Gemini-Embedding-001 (Lee et al., 2025)	0.591	0.553	0.629	0.598	0.619	0.560	0.667	0.630	0.690	**0.679**	0.685	0.651
Harrier-OSS-v1-27b (Microsoft, 2026)	0.585	0.554	0.612	0.586	0.609	0.562	0.659	0.615	**0.700**	0.664	0.678	0.640
KaLM-Embedding-Gemma3-12B-2511 (Zhao et al., 2025)	0.579	0.546	0.605	0.578	0.597	0.565	0.599	0.557	0.628	0.607	0.618	0.585
Qwen3-Embedding-0.6B (Zhang et al., 2025)	0.566	0.529	0.611	0.564	0.586	0.538	0.625	0.586	0.676	0.624	0.634	0.603
LLaMa-Embed-Nemotron-8B (Babakhin et al., 2025)	0.565	0.535	0.584	0.578	0.588	0.540	0.616	0.569	0.641	0.629	0.638	0.602
Jina-Embeddings-v5-Text-Small (Akram et al., 2026)	0.554	0.513	0.603	0.549	0.577	0.528	0.610	0.556	0.664	0.608	0.631	0.591
Harrier-OSS-v1-0.6b (Microsoft, 2026)	0.549	0.521	0.589	0.538	0.568	0.523	0.632	0.588	0.676	0.637	0.638	0.613
Text-Embedding-3-Large (OpenAI, 2024)	0.539	0.522	0.554	0.541	0.560	0.526	0.593	0.564	0.609	0.598	0.611	0.588
Jina-Embeddings-v5-Text-Nano (Akram et al., 2026)	0.522	0.484	0.563	0.529	0.547	0.487	0.569	0.522	0.604	0.576	0.585	0.560
EmbeddingGemma-300m (Schechter Vera et al., 2025)	0.511	0.488	0.547	0.502	0.545	0.472	0.588	0.548	0.629	0.594	0.604	0.559
Multilingual-E5-Large (Wang et al., 2024)	0.508	0.485	0.549	0.501	0.523	0.475	0.545	0.505	0.580	0.549	0.544	0.536
BGE-m3 (Chen et al., 2024)	0.505	0.483	0.539	0.498	0.518	0.485	0.563	0.513	0.616	0.575	0.567	0.543
Harrier-OSS-v1-270m (Microsoft, 2026)	0.497	0.466	0.537	0.489	0.517	0.473	0.557	0.525	0.590	0.565	0.565	0.539
Text-Embedding-3-Small (OpenAI, 2024)	0.497	0.470	0.540	0.483	0.519	0.473	0.554	0.520	0.597	0.552	0.565	0.536
TF-IDF	0.435	0.418	0.454	0.414	0.450	0.431	0.458	0.419	0.471	0.445	0.477	0.471
Approach Zero (Zhong et al., 2021)	0.428	0.425	0.433	0.432	0.451	0.392	0.473	0.478	0.467	0.457	0.492	0.475
Jaccard (Jaccard, 1912)	0.424	0.401	0.457	0.427	0.463	0.367	0.444	0.428	0.464	0.455	0.482	0.387
BERT (Devlin et al., 2018)	0.417	0.432	0.399	0.394	0.440	0.416	0.429	0.435	0.428	0.425	0.453	0.406
BM25 (Robertson & Zaragoza, 2009)	0.409	0.398	0.431	0.407	0.434	0.374	0.410	0.402	0.423	0.411	0.440	0.375
RoBERTa (Liu et al., 2019)	0.406	0.415	0.394	0.399	0.423	0.392	0.397	0.409	0.390	0.382	0.416	0.381

E. Compute Resources

This section details the computational and financial resources required to generate and evaluate our benchmark.

E.1. Data Collation and Processing

For the AoPS and NuminaMath datasets, the data was for the most part, readily available in a structured format, requiring only minor cleaning and deduplication. The official sources, however, required a more intensive processing pipeline, costing around $300 in compute for OpenAI Batch API usage over the course of a total of 72 hours.

E.2. Benchmark Generation

After deriving the large corpus of problems as described in §3.1, the first stage of benchmark generation consists of extracting topic-based and solution-summary-based signals for each problem in the corpus. Topic extraction for all ∼283K problems took approximately two days using 8 H200 GPUs. Extracting summaries for all solutions in the corpus took approximately 12 hours using 4 H200 GPUs. Computing the BMA and Jaccard scores was comparatively efficient, requiring 1 hour on 25 CPU cores. Storing the resulting signals for all problem pairs in the corpus required around 300GB of disk space, where we used a more efficient byte-level storage format.

After filtering for relevant problems, the next stage uses a Swiss-style tournament together with the Bradley–Terry algorithm. Pairwise judging for the 1000 filtered queries took approximately 160 hours on 4 H200 GPU nodes.

E.3. Applying the Benchmark

Computing the benchmark results in Tables 1 and 4 took approximately 1 hour using 4 H200 GPUs for the open-weight embedding models. For the OpenAI and Gemini models, our usage totalled approximately $20 in API costs.

F. Prompts

In this section, we make available the prompts used to generate our benchmark.

F.1. Ontology Topic Assignment Prompt

Ontology topic assignment

```
[SYSTEM]

You are an Expert Mathematical Annotator. You will be provided with a mathematical text(problem, solution,
theorem, lesson, etc.) and a list of possible tags. You must assign relevance score for each of the tags
with respect to the provided text.

# INPUT
You will be provided with:
1. A mathematical text
2. A comma-separated list of tags

# TASK
For each of the provided tags assign a real number from the interval [0.0, 1.0]. The most relevant tag to
the text must have a relevance score of 1.0 and the most irrelevant must have a score of 0.0. Align the
remaining tags with respect to those.

# OUTPUT SCHEMA
Constrain your output to a pure JSON with no explanations, markdown or comments. Return only pure JSON.
Follow the schema - each tag name is a key in the returned JSON object with a value the relevenace score:
```json
{
  "<tagName>": <float | tag relevance score 0.0-1.0 to the provided mathematical text>,
  ...
}

[USER]

# Mathematical Text:
{problem_and_solution}

With respect to the above text, assign relevance scores to the following tags and return the results in
JSON:

{list_of_comma_separaed_tags}
```

F.2. Extracting Short Solution Summaries Prompt

Extracting short solution summaries

```
# INSTRUCTION
You are an expert mathematical annotator tasked with identifying the *core idea*-the central mathematical
insight-from a math problem.

# GOALS
1. Identify what makes the solution work conceptually, not how to carry it out. Capture the untrial step or
 idea that is the greatest hint for the solution.
2. Never include any multi-step reasoning, equations, or numeric computations. Don't include any
annotations that are in the solution but not in the original problem statement.
3. Never try to solve the problem on your own, and don't include your reasoning or thoughts.
4. Output a single valid JSON object matching the schema below.
5. Structure the ideas imperatively so they look like you are giving a hint to someone.
6. If the problem seems too easy or straightforward, or you can't identify a core idea, store its value as
'null' and set the 'noCoreIdea' to 'true'.

# SCHEMA
```json
{
    "noCoreIdea": <true|false>,
    "coreIdea": "<string - one short sentence (up to 30 words) naming the main insight to the problem>",
    "supportingIdeas": ["<strings - 0-3 short technique phrases>"],
    "keywords": ["<strings - 1-2 word phrases summarizing the ideas, theorems, etc. in the solution>"],
    "confidence": <0.0-1.0>
}

Here are the problem statement and solution:

Statement: {problem}

Solution: {solution}
```

F.3. Pairwise Relevance Judging Prompt

Pairwise relevance judging

```
You are an expert mathematician and analytical-reasoning engine. Your job is to decide which of two sample
problems (Sample 1 or Sample 2) is *conceptually closer* to a target problem, judged by the *mathematical
ideas and solution techniques actually used* (not surface wording).

  You will receive 3 complete items, each with BOTH a problem statement and its correct solution:

  [Target]
  - Problem: {target_problem}
  - Correct Solution: {target_solution}

  [Sample 1]
  - Problem: {sample1_problem}
  - Correct Solution: {sample1_solution}

  [Sample 2]
  - Problem: {sample2_problem}
  - Correct Solution: {sample2_solution}

  Your output must follow the steps and formatting rules below exactly.

  --------
  STEP 1 - Technique Extraction (do this for Target, Sample 1, Sample 2)
  For each item, read the Problem + Correct Solution together, then produce a structured list containing:

  A. Core topic tags (2-6 tags)
    Examples: "modular arithmetic", "similar triangles", "graph invariants", "generating functions", "
optimization via calculus", "AM-GM", "inclusion-exclusion".

  B. Key tools / theorems / lemmas explicitly used
    Name the theorem/identity/strategy when possible (e.g., "Pigeonhole Principle", "Cauchy-Schwarz", "
Euclidean algorithm", "double counting", "invariant/monovariant", "casework", "recurrence + induction").

  C. Primary solution moves (3-8 bullets)
    Describe the actual moves that drive the solution, e.g.:
    - "rewrite expression into telescoping sum"
    - "introduce substitution of variable to linearize"
    - "construct auxiliary line and apply angle chase"
    - "count objects in two ways; equate counts"
    - "prove monotonicity; then bound endpoints"

  D. Structural signature (1-3 sentences)
    Summarize the solution's backbone: what is introduced, what is transformed, and what final step seals
the result.

  Label these sections clearly as:
  - Target: Technique Profile
  - Sample 1: Technique Profile
  - Sample 2: Technique Profile

  --------
  STEP 2 - Comparative Analysis
  Compare each sample to the target using *deep* similarity signals:

  1) Technique overlap:
    - Are the same kinds of tools central (not merely mentioned)?
    - Are the same transformations used (e.g., factoring vs. symmetry/invariants vs. counting)?

  2) Problem structure alignment:
    - Are the same intermediate objects introduced (auxiliary point, substitution, generating function,
invariant quantity, etc.)?
    - Do both require the same "shape" of argument (e.g., construction + chase, or setup of recurrence +
induction, or extremal argument + contradiction)? This criterion should be weighted lower than the
technique overlap but can be a tie-breaker when techniques are similar.

  3) Difficulty is NOT the criterion:
    - Prefer shared method and structure over "hard/easy".

  4) Penalize superficial similarity:
    - Do NOT reward matching variable names, story context, or domain language if the underlying method
differs. Further, algebraic computations do not imply similarity if the core method is different, or if the
 algebra is just a technical detail rather than the main driver.

  Produce two subsections:
  - Sample 1 vs Target -- Similarities / Differences / Overall Match Score (qualitative)
  - Sample 2 vs Target -- Similarities / Differences / Overall Match Score (qualitative)
```

```
  --------
  STEP 3 - Decision + Tie-break rules
  Choose the sample that is fundamentally closer in methods and structure.

  If it's close, use these tie-breakers in order:
  1) The sample whose *main* technique is the same as the target's main technique.
  2) The sample whose *sequence of moves* (setup -> transformation -> key lemma -> finish) matches more
closely.
  3) The sample that relies on the same representation (e.g., algebraic manipulation vs. geometric
configuration vs. combinatorial counting vs. graph reasoning).

  --------
  OUTPUT FORMAT (must follow)
  - You may include your extracted profiles and comparisons above.
  - Output your final decision on the last line, formatted as either:

  $\\boxed{{1}}$ -> corresponding to the first sample being more similar to the target,
  or
  $\\boxed{{2}}$ -> corresponding to the second sample being more similar to the target.
```

F.4. Ordinal Judge Prompt

Ordinal ranking prompt for additional ordering experiments §D.1

```
You are a helpful mathematical expert and technique-similarity evaluator. You will be given two
mathematical items: a Target problem with its correct solution, and a Sample problem with its correct
solution.

Your task is to determine how similar the two problems are based on the mathematical techniques, ideas, and
 solution structure actually used in their solutions --- not based on superficial wording, notation, topic
labels, or story context.

You will receive:

[Target]
Problem: {target_problem}
Correct Solution: {target_solution}

[Sample]
Problem: {sample_problem}
Correct Solution: {sample_solution}

Follow the instructions below exactly.

STEP 1 - Technique Extraction
For each item, read the Problem and Correct Solution together and identify all techniques genuinely used in
 the solution.

For both Target and Sample, produce:

A. Core topic tags (2-8 tags)
Examples: "modular arithmetic", "similar triangles", "double counting", "generating functions", "
optimization", "AM-GM", "inclusion-exclusion", "induction", "graph invariants".

B. Key tools / theorems / identities / lemmas used
Name explicit or implicit tools when possible, such as:
- Pigeonhole Principle
- Cauchy-Schwarz
- Euclidean algorithm
- Vieta's formulas
- telescoping
- contradiction
- casework
- recurrence + induction
- symmetry
- invariant / monovariant
- angle chase
- substitution
- extremal argument
- bounding

C. Primary solution moves (3-10 bullets)
Describe the actual moves that drive the solution, for example:
- rewrite the expression into a factorized form
```

```
- introduce a substitution to simplify the equation
- construct an auxiliary object
- split into cases
- count the same set in two ways
- derive a recurrence and solve it
- apply a standard inequality after normalization
- use parity or modular reduction
- reduce to a previously established identity

D. Structural signature (1-3 sentences)
Summarize the backbone of the argument: how the solution starts, what key transformation or idea unlocks it
, and what final step completes the proof.

Label these sections clearly as:

Target: Technique Profile
Sample: Technique Profile

STEP 2 - Comparative Analysis
Compare the Sample to the Target using the extracted technique profiles.

Your comparison must include:

1. Technique overlap
- Which techniques are shared?
- Which shared techniques are central rather than incidental?
- Are the same theorems, transformations, or proof strategies doing the main work?

2. Technique differences
- Which important techniques appear in one but not the other?
- Do the problems rely on different core ideas even if they are in the same broad subject?

3. Structural alignment
- Do the two solutions have a similar shape, such as:
  - setup -> substitution -> simplification -> conclusion
  - construction -> theorem application -> chase -> finish
  - recurrence -> induction
  - extremal argument -> contradiction
  - counting representation -> double count -> algebraic cleanup
- This matters, but it should be weighted less than overlap in core techniques.

Important rules:
- Focus on the methods actually used in the solutions.
- Do NOT reward superficial similarity such as matching notation, similar phrasing, same answer form, or
broad subject area alone.
- Do NOT treat routine algebraic manipulation as strong similarity unless it is a central technique in both.

- Weight main techniques much more heavily than minor technical steps.
- If one problem uses a tool only incidentally but the other depends on it centrally, count that as weak
overlap.

STEP 3 - Similarity Score
Assign a single integer score from 0 to 100 that reflects how similar the Sample is to the Target in terms
of mathematical method and solution structure.

Use this scale as guidance:

0-10: essentially no meaningful similarity in method
11-25: very weak similarity; mostly superficial or broad-topic overlap
26-40: limited similarity; some secondary techniques overlap but main approach differs
41-55: moderate similarity; at least one important method overlaps
56-70: fairly strong similarity; core ideas overlap substantially, with notable differences
71-85: strong similarity; main methods and argument structure are closely aligned
86-95: very strong similarity; nearly the same technique and solution pattern
96-100: essentially the same core method, with only minor presentation differences

Give a brief justification for the score in 2-5 sentences.

OUTPUT FORMAT
Use exactly this structure:

Target: Technique Profile
- Core topic tags: ...
- Key tools / theorems / identities / lemmas: ...
- Primary solution moves:
  - ...
  - ...
  - ...
- Structural signature: ...
```

```
Sample: Technique Profile
- Core topic tags: ...
- Key tools / theorems / identities / lemmas: ...
- Primary solution moves:
  - ...
  - ...
  - ...
- Structural signature: ...

Comparison
- Technique overlap: ...
- Technique differences: ...
- Structural alignment: ...
- Score justification: ...

Final Score:
\boxed{{N}}

where N is a single integer from 0 to 100.
```

F.5. Mathematical Equation Formatting Prompt

Formatting mathematical equations into LaTeX

```
Role: You are a LaTeX Formatting Expert specializing in mathematical notation standardization.

Task: You will be provided with mathematical text (problems or solutions). Your task is to standardize the
LaTeX formatting and enclose the entire final result within a \boxed{{...}} command.

Strict Constraints:
1. DELIMITER CONVERSION: Replace all instances of \( ... \) and \\( ... \\) with standard dollar signs.
   - Use $...$ for inline text.
   - Use $$...$$ for display equations.
2. UNIVERSAL MATH TAGGING: Apply math mode ($...$) to every single mathematical element without exception.
3. CONTENT INTEGRITY: Do not solve the problem or edit the prose.
4. FINAL WRAPPING: The entire output must be contained within \boxed{{ <your_formatted_text_here> }}.
5. NO VERBOSITY: Provide ONLY the \boxed{{...}} block.

Example Transformation:
Input: If the radius r is 5, find the area. Use \( \pi \).
Output: \boxed{{If the radius $r$ is $5$, find the area. Use $\pi$.}}
```

F.6. Human Annotator Instructions

Human Annotator Instructions

```
Goal: Compare Candidate 1 and Candidate 2 against the Target problem.

Candidate 1: Choose this if Candidate 1 is more similar to the Target.

Candidate 2: Choose this if Candidate 2 is more similar to the Target.

Tie: Choose this if both candidates are equally good / bad / equally similar.

Save: Records your judgment. Completed tasks turn green.

Remove judgment: Deletes your previous answer after confirmation. This removes the task from the list of "
Finished Tasks".
```

G. Use of LLMs in This Work

In this work, we have not used LLMs for research design, ideation, or decision-making. We have used LLMs to write code for assisting with plot creation and formatting, but all code was reviewed and edited by the authors. We have used LLMs to assist with polishing the writing, but all content was written by the authors and reviewed for accuracy. We have not relied on LLMs for generating any part of the bibliography, and all references were added manually by the authors.