RESEARCHMATH-14K: Scaling Research-Level Mathematics via Agents

Anonymous ACL submission

Abstract

The frontier of mathematics is defined by problems whose solutions are not yet known, yet it remains unclear whether language models can meaningfully engage with such problems without human intervention. A major obstacle is the lack of large-scale research-level math datasets. To this end, we introduce RESEARCHMATH-14K, a set of 14,056 problems curated from academic sources via a multi-agent pipeline, making it the largest collection of research-level mathematical problems to date. We further generate RESEARCHMATH-REASONING, 220K teacher trajectories from two open models, where we observe recurring avoidance behaviors such as non-attempts and fabricated references. Interestingly, across eight open-weight models, newer generations produce 5.6× more references and 5.0× more fake references per trace. After agentic filtering of RESEARCHMATH-REASONING, fine-tuning Qwen3 models from 4B to 30B parameters improves over base models by 9.2 points on average. This shows that filtered open-problem attempts can provide useful supervision even without fully correct reasoning traces.

1 Introduction

Mathematicians are trained over years, escalating from undergraduate textbooks and exercises to seminar problems, qualifying-style questions, and short-term research. Over time, they learn practices that are central to becoming mathematicians: decomposing problems into lemmas, testing examples, isolating tractable subproblems, distinguishing a plausible route from a proof, and reasoning under genuine uncertainty. Frontier proprietary models increasingly appear to internalize parts of this curriculum (Alexeev et al., 2026a,b; Zheng et al., 2026). However, the open-source landscape has not kept pace. Nearly all publicly available math training data targets contest-style problems at the olympiad level or below (Li et al., 2024; Fan et al., 2025b), and the few datasets that do reach the research frontier are positioned as held-out evaluation benchmarks, often gate-kept to prevent contamination (Glazer et al., 2024; Phan et al., 2025).

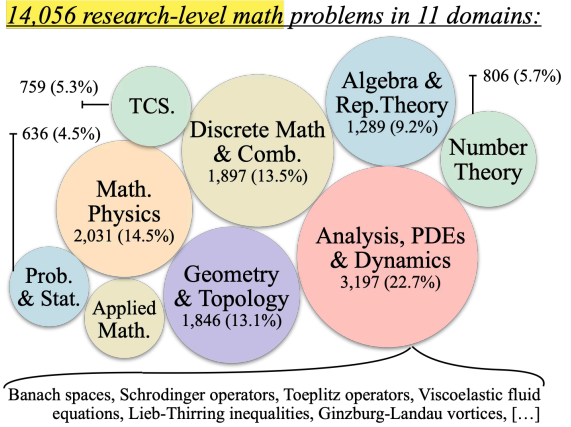

Figure 1: **Domain distribution of the RESEARCHMATH-14K.** Bubble size is proportional to the number of problems in each mathematical area. Logic and Foundations (455 problems, 3.2%) and Other/Cross-disciplinary (311 problems, 2.2%) are omitted from the visualization for readability.

Where, then, can research-level mathematical questions be obtained at scale? Recent work has largely relied on two expensive sources: multi-LLM pipelines that synthesize difficult problems (Zhang et al., 2026; Dekoninck et al., 2026), or expert mathematicians who write and curate them by hand (Son et al., 2026b; Garre et al., 2026). Both approaches are valuable, but neither provides an easy path to a broad, open training corpus. We take a different route. The mathematical literature already contains thousands of open problems, conjectures, seminar questions, and research directions. The bottleneck is extracting them from their local context and rewriting them into self-contained form. We collect 1,233 open-problem lists and research papers from zbMATH, arXiv, and academic repositories, then leverage agents to identify candidate questions, recover missing definitions and assumptions, and normalize them into

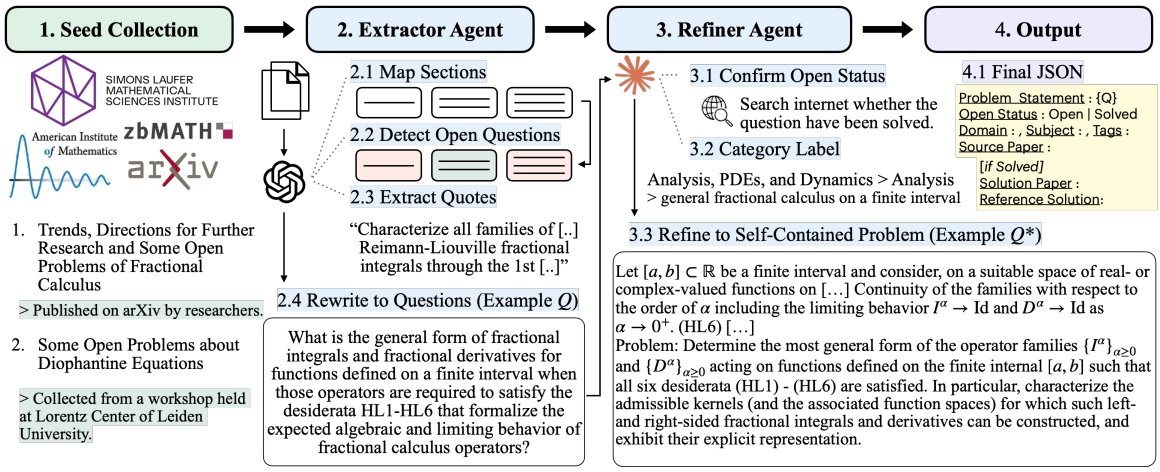

Figure 2: **Agentic construction pipeline for RESEARCHMATH-14K.** Starting from curated open-problem lists and research papers, an extractor agent maps document sections, detects candidate open questions, preserves supporting quotes, and rewrites them as standalone questions. A refiner agent then verifies open status, assigns taxonomy labels, and rewrites each candidate into a self-contained research problem before producing the final JSON record with statement, status, domain metadata, source, and solution fields when applicable.

standalone research-level problems. This process yields RESEARCHMATH-14K, a corpus of 14,056 research-level mathematical questions along with RESEARCHMATH-REASONING, 220K reasoning trajectories generated from two open models.

In a manual review of 100 sampled trajectories, roughly 30% are visibly problematic, including non-attempts, substitutions to narrower problems, and fabricated arXiv or PDF URLs (Section 2). These failures recur in a larger trace-level analysis of eight open-weight models, including `DeepSeek V4-Pro` (DeepSeek-AI, 2026) and `Kimi K2.6` (Team et al., 2026) (Section 3.2). Interestingly, newer models become more citation-heavy but less factual, with 54.0% of 720 RESEARCHMATH-14K traces containing at least one fake reference (Section 4). We use the same behavioral and factuality filters to clean RESEARCHMATH-REASONING into RESEARCHMATH-REASONING-FILTERED, a 5,000-trace training-ready subset (Section 5). Fine-tuning three Qwen3 base models (4B, 8B, and 30B-A3B) on RESEARCHMATH-REASONING-FILTERED improves them by 9.2 percentage points on average, showing that the RESEARCHMATH family is a valuable training resource for research-level reasoning even without ground-truth solutions. We openly release the RESEARCHMATH family, comprising RESEARCHMATH-14K (14,056 research-level problems), and RESEARCHMATH-REASONING (220K teacher trajectories), under the MIT license to support future work on research-level mathematical reasoning.

2 RESEARCHMATH-14K

2.1 Collecting Existing Open Questions

We build RESEARCHMATH-14K with a two-stage agentic pipeline: an Extractor agent pulls candidate problem statements from each source document, and a Refiner agent rewrites each statement into a self-contained problem, consulting online references. The pipeline produces 20,835 problems from 1,233 source documents, see Figure 2.

Sources. Mathematicians have long published unresolved questions through workshops, surveys, and curated lists (Guy, 2004), both to attract collaborators and to record which questions a field regards as important enough to foreground for the broader community. Our pipeline captures both classical entries, such as Hilbert- or Erdős-style problem lists, and *contemporary* problem statements about modern mathematical objects and local technical settings. The latter are closer to the day-to-day research questions a working mathematician might pose at a workshop or in a recent survey, and are therefore the kind of supervision signal we target. See Appendix A for examples.

Specifically, source documents are drawn from three streams. **arXiv open-problem papers** (e.g., Diethelm et al. (2022)) (524 documents, 8,182 problems) are surveyed by searching arXiv for titles and abstracts mentioning "open problems," or "unsolved." **Open-problem web pages** (161 documents, 5,331 problems) are discovered by Google search and cover hosts such as `academia.edu`, MathOverflow, and Wikipedia. **Problem-session**

sheets and curated lists (548 documents, 7,322 problems) are the third stream and include two sub-types: AIM-style *workshop problem sessions* where participants pose questions at the end of a meeting,[1] and *conference/proceedings open-problem rounds* compiled by an editor at the close of a special session.

Extractor Agent. The Extractor, driven by Codex with GPT-5.5 at *xhigh* reasoning effort, processes one source per run. It first follows the source URL down to the PDF or HTML page that holds the full text, discarding any document hidden behind a paywall. Before extraction, it also screens the document to confirm that it actually contains a problem list, skipping papers that do not in fact pose open problems (e.g., regular research papers that merely mention "open problem"). It then reads the paper end-to-end and extracts each open problem as a verbatim quote together with a first-level rewrite. While rewriting, the model is instructed to jump back and forth through the paper to pull in every definition and statement needed to understand that problem. Across the 1,233 documents the Extractor yields a mean of 16.9 questions per source (median 10, maximum 358).

Refiner Agent. Reading through the extracted questions, the authors noticed that some of the snippets still miss the definitions, notation, and hypotheses the original paper treats as already given. The Refiner, driven by Claude Code with Opus 4.7 at *medium* reasoning effort, fills that gap. It performs two tasks. First, it re-reads the original paper to inline every definition and hypothesis needed to state the problem in isolation. Second, it searches up to ten later papers that cite or extend the source, both to pull in the background of the source treated as implicit and to determine whether the problem has since been resolved. Each problem is tagged as *open*, *partially solved*, *solved*, or *unknown*. We audit 500 random records labels using GPT-5.5 as LLM-Judge. It labels 94.2% of refined statements as self-contained, compared with 67.2% of original extractions, a 27.0 percentage-point improvement (Appendix B). Refined statements also average 1,192 characters, up from 290 at the Extractor stage, a 4.1× expansion.

[1] https://aimath.org/pastworkshops/nonselfadjointproblems.pdf

Dataset	#Problems	Source	Diff.
GSM8K (Cobbe et al., 2021)	8.5k	textbook	
MATH (Hendrycks et al., 2021)	12.5k	competitions	
LeanDojo (Yang et al., 2023)	122k	research lit.	
MathInstruct (Yue et al., 2024)	262k	synthetic	
MetaMathQA (Yu et al., 2024)	395k	synthetic	
PRM800K (Lightman et al., 2024)	800k	competitions	
NuminaMath (Li et al., 2024)	860k	synthetic	
AceMath-Instruct (Liu et al., 2025b)	1.66M	synthetic	
OpenMathInstruct (Toshniwal et al., 2024)	1.8M	synthetic	
Riemann-Bench (Garre et al., 2026)	25	expert	
FrontierMath (Glazer et al., 2024)	300	expert	
Soohak (Son et al., 2026a)	439	expert	
GHOSTS (Frieder et al., 2023)	709	textbook	
HARDMath (Fan et al., 2025a)	1,426	textbook	
HLE (Phan et al., 2025)	2,500	expert	
RESEARCHMATH-14K	**14,056**	**research lit.**	

Table 1: **Representative public math datasets.** Top section: training datasets are large, but do not expand to the research level. Lower section: existing research-grade resources are small (< 3k items) and evaluation-only. RESEARCHMATH-14K fills both gaps, large-scale and research-grade. Difficulty key: grade-school, olympiad, undergrad, research.

2.2 Filtering Near Duplicates

The collection pipeline produces a seed set of 20,835 problems, but multiple sources often state the same open problem in slightly different forms, making duplicate filtering necessary. We embed all problems with `Qwen3-Embedding-8B` (Zhang et al., 2025) and compute pairwise similarities over both the original statements and the self-contained rewrites. Questions extracted from the same paper often share extensive background text and can look similar even when they are distinct, so a low similarity threshold would introduce many false positives. After manually inspecting borderline pairs at several cutoffs, we set the threshold to 0.9. A pair is marked as a duplicate if either similarity score exceeds this value. This threshold separates most true duplicates from same-paper false positives. For each duplicate pair, we keep the version from arXiv or another paper source and discard the version discovered through Google search; when both sources have the same priority, we choose one at random. Although this filtering is conservative, some distinct but closely related questions may still be removed, so we also release the raw seed set. This leaves a final collection of 14,056 problems. See Appendix D for further details on the similarity distribution and examples of non-duplicate question pairs near the threshold.

2.3 Dataset Statistics

Composition. Each problem is assigned a three-level taxonomy: one of 11 domain groups,

$$\mathcal{G} = \begin{cases} \text{Analysis, PDEs, and Dynamics;} & \text{Mathematical Physics;} \\ \text{Discrete Mathematics and Combinatorics;} & \text{Number Theory;} \\ \text{Geometry and Topology;} & \text{Theoretical Computer Science;} \\ \text{Algebra and Representation Theory;} & \text{Probability, Statistics, and ML;} \\ \text{Applied and Computational Mathematics;} & \text{Logic and Foundations;} \\ \text{Other / Cross-disciplinary.} \end{cases}$$

one of 28 macro-subjects, and a research-level category tag (11,611 unique tags). The hierarchy runs from broad area to research field to local topic. For example, one branch is:

$$\begin{aligned} \text{domain} &= \text{Geometry and Topology} \\ \text{macro} &= \text{Algebraic Geometry} \\ \text{tags} &= \begin{cases} \text{Hilbert schemes;} & \text{Brill–Noether theory;} \\ \text{curves over finite fields;} & \text{Kuznetsov components;} \\ \text{stability conditions;} & \text{line arrangements;} \\ \text{affine spaces;} & \text{Hurwitz spaces;} \\ \text{automorphism groups of curves.} \end{cases} \end{aligned}$$

Figure 1 shows the level-one distribution. The corpus is broad but skewed toward four large areas: Analysis/PDEs/Dynamics, Mathematical Physics, Discrete Mathematics/Combinatorics, and Geometry/Topology together account for 8,971 problems (63.82%). A small fraction, 311 problems (2.2%), falls into the *Other/Cross-disciplinary* group and covers science-adjacent open questions (e.g., on supernova progenitors, origin of language, computational theory of mind). Open problems form the majority (8,313, 59.14%), followed by *unknown* (2,489, 17.71%), *partially solved* (2,083, 14.82%), and *solved* (1,171, 8.33%). The set is source-diverse, spanning 1,138 unique documents, with the top 10 contributing 1,431 problems (10.18%) and the top 50 contributing 3,623 (25.78%).

Difficulty. Difficulty is multidimensional, and a problem can be hard because it requires obscure background knowledge (*Knowledge*), demands novel thinking that deviates from existing approaches (*Novelty*), or involves compute-heavy multi-step reasoning (*Procedural*). We compare RESEARCHMATH-14K against `AceMath` (Liu et al., 2025b), `AIME(2024-2026)` (Dekoninck et al., 2026), `HLE-Verified` (Phan et al., 2025), and `NuminaMath` (Li et al., 2024). From each of the five datasets we sample 90 problems and consider all $\binom{5}{2}$ dataset pairs. For each pair we randomly draw 100 cross-dataset problem pairs and randomize their order, giving 1,000 total comparisons. Each comparison is judged by `GPT-5-mini` along the three axes, producing win/loss/draw labels from which we compute Elo ratings. On all three axes, RESEARCHMATH-14K ranks above these existing math datasets by roughly 400 Elo points (Figure 3), implying that it is a qualitatively harder problem class rather than an incremental step above existing math datasets. This highlights our contribution as the hardest open-source math problem set to date.

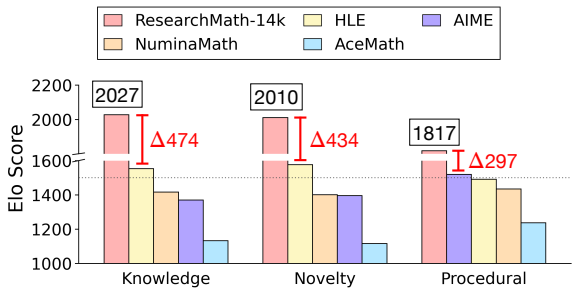

Figure 3: **Elo Ratings for Difficulty Comparison.** Ratings are computed from pairwise LLM difficulty judgments. All sources start at 1500 with $k = 32$; wins score 1, losses score 0, and draws score 0.5. Higher Elo means the source is judged more difficult more often.

2.4 Generating Responses

We use two teacher models, `GPT-OSS-120B` (Agarwal et al., 2025) and `Qwen3-30B-A3B` (Yang et al., 2025), to generate reasoning trajectories for RESEARCHMATH-14K. Note that the goal is not to produce *correct* solutions. Most solutions are not yet known, and we do not expect sub-trillion-parameter models to solve open research questions. We initially fine-tune `Qwen3-4B` on these trajectories without any filtering. This leads to substantial degeneration of the student model, including repetitive outputs and frequent non-attempts.[2] To understand why, we conduct a human review of 100 randomly sampled trajectories. We find that in 25 cases the teacher does not attempt the problem at all. Instead, the model appears to recognize the question as an open problem and outputs a non-attempt in one of the following forms:

- 21/100: lists known related references, and outputs "open" as the answer.
- 4/100: after concluding the problem is open, narrows the conditions and either solves the narrowed version or simply lists related references.

These observations motivate the larger-scale behavioral and factuality analysis in Section 3.2. Nonetheless, the resulting set pairs 14K prompts with 220K responses (approximately 16 per prompt) from two teacher models, and we release it as RESEARCHMATH-REASONING, which is, to our knowledge, the largest publicly available collection of model attempts on research-level math.

[2]We do not report specific scores for this unfiltered fine-tune because the resulting model degenerated on nearly every evaluation, scoring close to zero. The point of the anecdote is the failure mode, which motivates the larger-scale analysis.

3 Experiment Setup

The cause of such fabricated reasoning trajectories (Section 2.4) is subject to several possible explanations. The behavior may reflect problem difficulty, stylistic mismatch between paper-derived prompts and benchmark-style questions, or the limited capacity of GPT-OSS-120B. We therefore set up experiments across models and benchmarks (Section 3.1) and evaluate them with complementary behavioral metrics (Section 3.2).

3.1 Baselines

Models. We evaluate a broad set of models, including several substantially larger systems and both older and newer generations from each model family: `DeepSeek R1` (Guo et al., 2025), `DeepSeek V4-Pro` (DeepSeek-AI, 2026), `Kimi K2` (Team et al., 2025), `Kimi K2.6` (Team et al., 2026), `Qwen3` (30B-A3B, 235B-A22B) (Yang et al., 2025), and `Qwen3.5` (35B-A3B, 397B-A17B) (Qwen Team, 2026). Throughout the analysis we group these models into four older→newer matched pairs (`R1`→`V4-Pro`, `K2`→`K2.6`, `Qwen3 30B`→`Qwen3.5 35B`, and `Qwen3 235B`→`Qwen3.5 397B`).

Benchmarks. RESEARCHMATH-14K has two defining properties: problems are *research-level*, and their surface form is *AI-refined* from a source paper. We choose four control benchmarks to isolate each property. To control for any artifact of the AI-refining step, we use `SOOHAK` (Son et al., 2026a) and `Leipzig Tier-4` (ScienceBench, 2026), both research-level but human-authored. To study the effect of difficulty, we use the math subset of `HLE-Verified` (a version of Humanity's Last Exam (Phan et al., 2025) verified by Zhai et al. (2026)) and `AIME` (Zhang and Math-AI, 2024, 2025, 2026). Both are easier than the research-level sets, with `AIME` being easiest. `AIME` combines questions from 2024, 2025, and 2026 for 90 problems in total. We sample 90 items from each of the other four benchmarks, with `SOOHAK` restricted to items labeled *graduate* or beyond from the challenge subset, and all benchmarks further filtered to short-form-answer questions; this leaves `SOOHAK` with 86 items, for 446 prompts overall.

3.2 Behavior and Factuality Metrics

Analyzing trace-level behavior is not trivial. We use two complementary methods that together cover two aspects of a reasoning trace, the model's *behavior* (how it reasons) and the *factuality* of what it cites. Each method covers both axes.

Rule-Based Counting. We use three curated phrase lists, each targeting a distinct phrasing pattern. The lists were assembled by the authors after reviewing dozens of model reasoning traces and collecting recurrent phrases that fit each pattern, and matching is performed against the lower-cased trace (full lists in Appendix C). `cite` matches citation-like nouns (e.g. "paper"). `abandon` catches abandonment (e.g. "cannot solve", "educated guess"). `assume` catches claims made without justification (e.g. "known result", "i remember"). Two of these (`abandon`, `assume`) measure behavior, while `cite` measures factuality and bridges into the agent-judge below. Each counter increments by one per match, and per benchmark we report the *row-hit rate* $\frac{1}{|T|}\sum_{i\in T}\mathbf{1}[n_{i,c}>0]$, the fraction of traces in which counter c matches at least once ($n_{i,c}$ is the match count in trace i and T is the set of traces). These rules are transparent, cheap, and chosen to broadly cover recurring failure patterns. Counting alone, however, cannot judge whether a given match is a real failure in context.

Agent-Judge. For an additional behavior check, we use `GPT-5.5` as a judge (Zheng et al., 2023) to detect *lemma decomposition*. The judge is prompted to generate a binary label on whether the solver model breaks the problem into provable subgoals, inspected over the first 30% of the trace, where subgoal-setting tends to happen. We highlight lemma decomposition as it is one of the most critical behaviors for LLMs to tackle open questions across long reasoning time. The factuality check inspects whether reference-like spans in the trace correspond to real sources. Because running an agent over a full reasoning trace is expensive, we use a two-stage pipeline. We slice each trace into newline-delimited blocks and use `GPT-5.4-nano` as to audit each block and extract reference-like spans (books, papers, website URLs). A search-enabled `Codex` agent then iterates over each span to confirm whether the span is genuine reference text (filtering out e.g. named mathematical theorems) and whether the referenced source exists on the web. We provide the surrounding block for reference, and require multiple web searches before every judgment. Prompts for both checks are in Appendix H.4. Both judge outputs measure properties of the reasoning trace, not correctness.

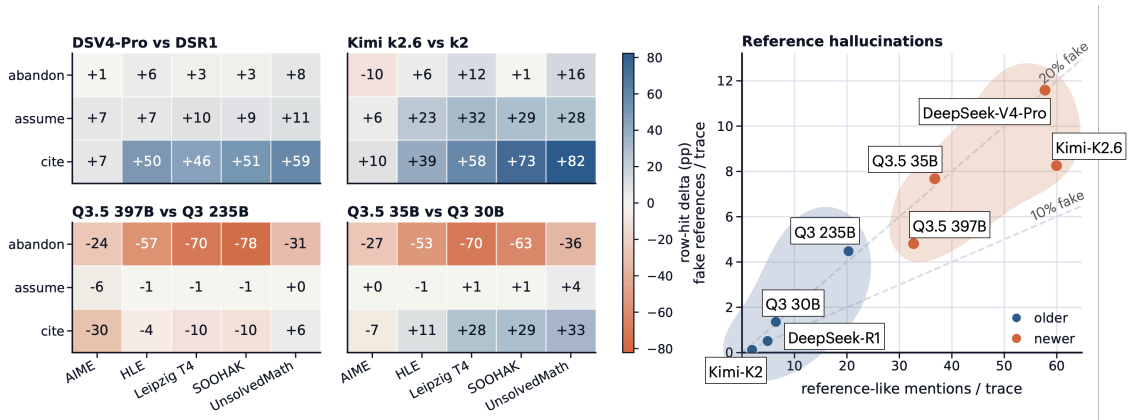

Figure 4: **Citation behavior across four matched older→newer model pairs. Left**: Row-hit-rate deltas (newer minus older, in percentage points) for the three rule-based counters (abandon, `assume`, `cite`) across the five benchmarks. **Right**: Agent-Judge reference verification on 720 RESEARCHMATH-14K traces, one point per model. Newer models (orange) sit upper-right of their predecessors (blue), with more reference-like mentions per trace (x-axis) and more references judged fake per trace (y-axis); dashed guides mark 10% and 20% fake-reference shares. Full per-model counts in Appendix E.

4 Analyzing Reasoning Behavior on RESEARCHMATH-14K

The manual review in Section 2 flagged roughly 30% of teacher trajectories as visibly problematic. We now measure the same failure modes at corpus scale using the eight models and five benchmarks from Section 3, and report two findings.

> **Finding**: Across model families, the newer generation cites external sources more often than the older one, and produces more fake references per trace.

Citation-like reasoning rises sharply in newer model generations (Figure 4, left, `cite` row), with row-hit rates increasing by 30-80 percentage points on RESEARCHMATH-14K, Leipzig Tier-4, and SOOHAK across the DeepSeek, Kimi, and Qwen3 matched pairs. The effect weakens as benchmarks get easier (modest on HLE, near zero on AIME), suggesting that newer models' tendency to cite is an artifact of the academic level of the questions.

To supplement the keyword counter, we use the Agent-Judge (Section 3.2) on 90 traces from RESEARCHMATH-14K for each of our 8 models. Across all 720 traces, 629 (87.4%) cite at least one reference-like object and 389 (54.0%) contain at least one fake reference. At the reference level, we inspect 19,864 extracted mentions and label 3,492 fake (17.6%) after consulting internet search (Figure 4, right). Per-trace mention counts grow dramatically across the matched comparisons. DeepSeek R1 → V4-Pro rises from 4.9 to 57.8 mentions per trace (0.5 → 11.6 fakes), Kimi K2 → K2.6 from 1.9 to 60.0 (0.1 → 8.3 fakes), Qwen3 30B → Qwen3.5 35B from 6.5 to 36.7 (1.4 → 7.7 fakes), and Qwen3 235B → Qwen3.5 397B from 20.3 to 32.7 (4.5 → 4.8 fakes). In aggregate, newer models produce 5.6× more reference-like mentions per trace and 5.0× more fakes. The fake mentions are mostly hallucinated paper titles and author attributions. Models try to ground their arguments on wrong statements by fabricating that a supporting reference exists, making the result sound correct. Representative fakes:

- "Neeman's paper: *A remark on the unique factorization theorem*"
- "J. Winkelmann, *On the holomorphic equivalence of the Koras–Russell cubic*"
- "a specific paper: *On the probability that a random polynomial is stable* by J. M. Anderson"

Why do newer models fabricate more often? Interestingly, we observe that models released in 2025 (`DeepSeek R1`, `Kimi K2`, `Qwen3`) cite less, while models released in 2026 (`DeepSeek V4-Pro`, `Kimi K2.6`, `Qwen3.5`) cite far more, with more fake citations. In other words, factuality on research-level prompts is moving backward. Because this pattern holds across DeepSeek, Kimi, and Qwen, three different model families, it is unlikely to be a quirk of any single training set. One plausible explanation is internet-search RL, or more broadly agentic RL (Dong et al., 2025; Liu et al., 2025a; Li et al., 2026). Recent post-training pipelines often place the model inside an agentic harness at train time, equipped with explicit search and citation tools, and reward it for grounding claims in retrieved sources. Over training, the model learns to invoke papers, books, and URLs as a routine part

Model	RESEARCHMATH-14K	Leipzig T4	SOOHAK
DeepSeek R1	1/90	0/90	0/86
DeepSeek V4-Pro	2/90 (+1)	2/90 (+2)	0/86 (0)
Kimi K2	2/90	0/90	0/86
Kimi K2.6	0/90 (−2)	1/90 (+1)	1/86 (+1)
Qwen3 30B	0/90	0/90	0/86
Qwen3.5 35B	3/90 (+3)	0/90 (0)	0/86 (0)
Qwen3 235B	0/90	0/90	0/86
Qwen3.5 397B	3/90 (+3)	1/90 (+1)	2/86 (+2)

Table 2: LLM-judge lemma-decomposition positives by model and benchmark. Each cell reports positive traces over judged traces; the colored parenthetical on the newer model gives the delta (newer minus older) within each matched pair (green = newer fires more, red = newer fires less).

of producing an authoritative-looking answer. In our setting, however, models are evaluated without internet access. A plausible explanation is that rather than abandoning the citation behavior when the search tool is unavailable, models keep invoking the learned pattern and simply fabricate the references they would normally retrieve.

Finding: Models parrot the style of research mathematics without engaging its underlying reasoning.

It should be noted, however, that citations and compression are not themselves failures. Mathematicians cite, reduce, and skip routine details too, and if models could ground their citations correctly, this would be less of a concern. But citations are not the only place where models try to look the part. On RESEARCHMATH-14K the abandon counter (Section 3.2) matches only 125/720 traces (17.4%), while `assume` matches 677/720 (94.0%; Figure 4, left). Models rarely give up outright, and the attempt almost always leans on compressed claims rather than from-scratch derivation. These surface signs resemble mathematician practice, but we cannot tell whether models employ the underlying reasoning or simply parrot the form.

Using the Agent-Judge lemma-decomposition metric from Section 3.2, we find that the behavior is almost absent (Table 2). Across RESEARCHMATH-14K, Leipzig Tier-4, and SOOHAK, only 18/2,128 judged traces are marked positive, and on RESEARCHMATH-14K only 11/720. This matters not just for RESEARCHMATH-14K but for any research-level mathematics models will face. Such problems are hard enough that they cannot be solved in a single pass and must be broken down into checkable subproblems.

5 Learning from RESEARCHMATH-14K

Prior work shows that supervised fine-tuning on mathematical reasoning can tolerate a moderate fraction of incorrect solutions (Toshniwal et al., 2025; Muennighoff et al., 2025; Son et al., 2025). We push this idea to a setting where correctness is largely unavailable. For RESEARCHMATH-14K, most problems are open or beyond the reach of current sub-trillion-parameter LLMs, so the trajectories in RESEARCHMATH-REASONING are unlikely to be complete nor correct. However, we hypothesize that, after filtering RESEARCHMATH-REASONING to exclude trajectories flagged by either rule-based counters or agent judges (Section 3.2), the remaining traces are qualitatively different from merely wrong work. Watching a trained researcher attempt an open problem and fall short can be instructive in a way that watching a kindergarten student make an arithmetic mistake is not. The former may introduce relevant objects, explore plausible reductions, test examples, or develop partial arguments, while the latter usually carries little transferable structure. Whether these *wrong-but-reasonable* traces are genuinely useful is therefore an empirical question with practical consequences. Requiring verified-correct reasoning at the research level would mean expert-annotating every trajectory, a cost that does not scale. If such traces are sufficient to teach useful behavior, they provide a cheaper path for future frontier-level data curation. In the following section, we investigate whether training on these attempts provides a useful signal.

5.1 Training Setup

We filter RESEARCHMATH-REASONING using the Agent-Judge pipeline from Section 3.2. This verifies every reference-like span against web search, and traces containing any reference judged fake are removed. Because the agent step calls multiple agents and paid web-search APIs, our budget allows producing only 5,000 filtered traces, which form RESEARCHMATH-REASONING-FILTERED. For comparison, we randomly sample 5,000 traces from `DASD-Thinking` (Yan et al., 2026) to test the alternative explanation that any gain comes from learning the output format rather than from research-level content. We fine-tune `Qwen3-4B/8B/30B-A3B-base` with LoRA on each training set. See Appendix G for training configurations. We evaluate on AIME 2024–2026 ($n = 90$), HLE ($n = 315$), and SOOHAK Challenge and

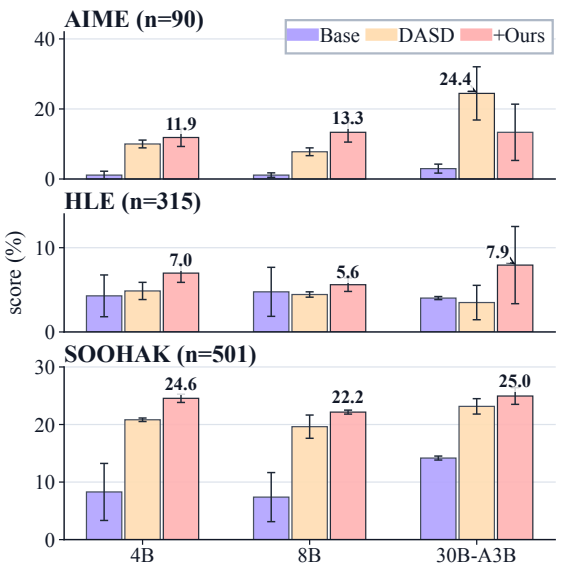

Figure 5: **Fine-tuning results by benchmark.** Bars show mean score for each model averaged across three runs; whiskers show standard deviation over three runs.

Mini combined ($n = 501$). We filter HLE and SOOHAK to include questions with integers only, and use `math-verify`[3] for scoring.

5.2 Training Results

Finding: Open-problem trajectories teach models more about research-level reasoning than olympiad data does, even without ever solving the problem.

Training on RESEARCHMATH-REASONING-FILTERED improves over the base models in all 9 model×benchmark cells, with a mean gain of $+9.2$ percentage points, while DASD improves in $7/9$. RESEARCHMATH-REASONING-FILTERED also outperforms DASD in $8/9$ cells, with the clearest gains on the research level evaluations. Averaged over HLE and SOOHAK, it is $+2.6$ points above DASD, with the largest gaps at HLE for the 30B model ($+4.4$) and SOOHAK for the 4B model ($+3.8$). The only exception is AIME for the 30B model, where DASD wins by 11.1 points.

Two implications follow. First, the gains are not explained by generic math reasoning exposure alone. DASD improves the base models, but RESEARCHMATH-REASONING-FILTERED does better in nearly all settings. Second, useful research-level supervision need not be verified-correct. Once non-attempts, unsupported claims, and fake citations are removed, wrong-but-reasonable attempts still improve student models. We leave further tests of this signal at larger scale to future work.

[3] `https://github.com/huggingface/Math-Verify`

6 Related Works

Research-Level Mathematics with LLMs. Inducing mathematical reasoning in LLMs has been driven mainly by resources with known answers (Toshniwal et al., 2024; Li et al., 2024; Yuan et al., 2026). However, most remain below the research frontier. These problems are typically solved, verifiable (Albalak et al., 2025), synthetic (Toshniwal et al., 2025), textbook-derived (Fan et al., 2025b), olympiad-derived (Mahdavi et al., 2025; Ko et al., 2025), or tied to formal proof environments (Yang et al., 2023). Research-level mathematical data remains expensive and non-trivial to scale: existing resources are often expert-authored (Son et al., 2026a), private or gated (Garre et al., 2026), small, continuously maintained for evaluation (Dekoninck et al., 2026), or difficult to convert into training material because usable prompts require local definitions, notation, hypotheses, status checks, and deduplication (Zhang et al., 2026). We address this gap by collecting research-level questions already present in the mathematical literature and rewriting them. The result is RESEARCHMATH-14K, a 14,056-problem corpus that, to the best of our knowledge, is the largest collection of research-level mathematical questions available for training.

7 Conclusion and Future Work

This work uses RESEARCHMATH-14K to study how open reasoning models behave on research-level mathematical problems whose complete solutions are often unavailable. Our trace-level analysis shows a concerning shift. Newer model generations produce more citation-heavy responses, but also more fake references. At the same time, these imperfect attempts still contain useful supervision. Fine-tuning on the filtered trajectories improves models by an average 9.2 percentage points over their base versions. These results suggest that research-level training need not rely only on verified complete solutions: wrong-but-reasonable attempts can be useful when their most harmful failure modes are removed. We encourage future works to test this signal at a larger scale, while clarifying when correct traces remain necessary for reliable proof behavior. We publicly release RESEARCHMATH-14K and RESEARCHMATH-REASONING to support future works on research-level mathematics.

Limitations

Despite our efforts to make RESEARCHMATH-14K broad, reproducible, and useful as a training resource, several limitations remain. First, our trace-level analysis is restricted to open-weight models. We evaluate eight models from three families, but exclude frontier proprietary systems such as GPT, Claude, and Gemini because they do not expose reasoning traces. Closed models may exhibit different reasoning patterns, especially with some deliberately targeting research level math as key goals. Second, both filtering and training are limited in scale. We cap the filtered training set at 5,000 traces, train a single LoRA configuration at fixed rank, and do not run full SFT. This is much smaller than recent open-source math SFT efforts that use hundreds of thousands of examples (Li et al., 2024; Toshniwal et al., 2024). We note that this experiment is meant to confirm that our dataset works as a training signal, not to aim for state-of-the-art performance. Nevertheless, our experiments should be viewed as evidence that the signal exists, not as an exhaustive study of how far it can scale. Finally, our evaluation necessarily simplifies research-level mathematical reasoning. For HLE and SOOHAK, we restrict evaluation to integer-answer questions because reliable automatic grading of free-form research-level proofs remains difficult at scale. This gives us reproducible quantitative results, but it likely underrepresents the proof-style reasoning that motivates RESEARCHMATH-14K. Thus, our experiments measure whether filtered open-problem attempts transfer to hard mathematical benchmarks, not whether the resulting models can produce complete research-grade proofs.

References

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, and 1 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*.

Alon Albalak, Duy Phung, Nathan Lile, Rafael Rafailov, Kanishk Gandhi, Louis Castricato, Anikait Singh, Chase Blagden, Violet Xiang, Dakota Mahan, and 1 others. 2025. Big-math: A large-scale, high-quality math dataset for reinforcement learning in language models. *arXiv preprint arXiv:2502.17387*.

Boris Alexeev, Moe Putterman, Mehtaab Sawhney, Mark Sellke, and Gregory Valiant. 2026a. Short proofs in combinatorics and number theory. *arXiv preprint arXiv:2603.29961*.

Boris Alexeev, Moe Putterman, Mehtaab Sawhney, Mark Sellke, and Gregory Valiant. 2026b. Short proofs in combinatorics, probability and number theory ii. *arXiv preprint arXiv:2604.06609*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI. 2026. Deepseek-v4: Towards highly efficient million-token context intelligence.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. 2026. Beyond benchmarks: Matharena as an evaluation platform for mathematics with llms.

Kai Diethelm, Virginia Kiryakova, Yuri Luchko, JA Tenreiro Machado, and Vasily E Tarasov. 2022. Trends, directions for further research, and some open problems of fractional calculus. *Nonlinear Dynamics*, 107(4):3245–3270.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, and 1 others. 2025. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849*.

Fan Fan, Sarah Martinson, Erik Wang, Kaylie Hausknecht, Jonah Brenner, Danxian Liu, Nianli Peng, Corey Wang, and Michael Brenner. 2025a. Hardmath: A benchmark dataset for challenging problems in applied mathematics. In *International Conference on Learning Representations*, volume 2025, pages 13523–13556.

Run-Ze Fan, Zengzhi Wang, and Pengfei Liu. 2025b. Megascience: Pushing the frontiers of post-training datasets for science reasoning. *arXiv preprint arXiv:2507.16812*.

Simon Frieder, Luca Pinchetti, Chevalier Chevalier, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Petersen, and Julius Berner. 2023. Mathematical capabilities of chatgpt. *Advances in neural information processing systems*, 36:27699–27744.

Suhaas Garre, Erik Knutsen, Sushant Mehta, and Edwin Chen. 2026. Riemann-bench: A benchmark for moonshot mathematics. *arXiv preprint arXiv:2604.06802*.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, and 1 others. 2024. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai. *arXiv preprint arXiv:2411.04872*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Richard K Guy. 2004. *Unsolved problems in number theory*, volume 21. Springer.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Hyunwoo Ko, Guijin Son, and Dasol Choi. 2025. Understand, solve and translate: Bridging the multilingual mathematical reasoning gap. In *Proceedings of the 5th Workshop on Multilingual Representation Learning (MRL 2025)*, pages 78–95.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, and 1 others. 2024. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9.

Wanli Li, Bince Qu, Bo Pan, Jianyu Zhang, Zheng Liu, Pan Zhang, Wei Chen, and Bo Zhang. 2026. Literesearcher: A scalable agentic rl training framework for deep research agent. *arXiv preprint arXiv:2604.17931*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601.

Junteng Liu, Yunji Li, Chi Zhang, Jingyang Li, Aili Chen, Ke Ji, Weiyu Cheng, Zijia Wu, Chengyu Du, Qidi Xu, and 1 others. 2025a. Webexplorer: Explore and evolve for training long-horizon web agents. *arXiv preprint arXiv:2509.06501*.

Zihan Liu, Yang Chen, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2025b. Acemath: Advancing frontier math reasoning with post-training and reward modeling. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 3993–4015.

Sadegh Mahdavi, Muchen Li, Kaiwen Liu, Christos Thrampoulidis, Leonid Sigal, and Renjie Liao. 2025. Leveraging online olympiad-level math problems for llms training and contamination-resistant evaluation. *arXiv preprint arXiv:2501.14275*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. 2025. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20286–20332.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, and 1 others. 2025. Humanity's last exam. *arXiv preprint arXiv:2501.14249*.

Qwen Team. 2026. Qwen3.5: Towards native multimodal agents.

ScienceBench. 2026. Benchmarks in leipzig. Accessed: 2026-05-22.

Guijin Son, Seungone Kim, Catherine Arnett, Hyunwoo Ko, Hyein Lee, Hyeonah Kang, Jiang Longxi, Jin Yun, JungYup Lee, Kyungmin Lee, and 1 others. 2026a. Soohak: A mathematician-curated benchmark for evaluating research-level math capabilities of llms. *arXiv preprint arXiv:2605.09063*.

Guijin Son, Donghun Yang, Hitesh Laxmichand Patel, Amit Agarwal, Hyunwoo Ko, Chanuk Lim, Srikant Panda, Minhyuk Kim, Nikunj Drolia, Dasol Choi, and 1 others. 2025. Pushing on multilingual reasoning models with language-mixed chain-of-thought. *arXiv preprint arXiv:2510.04230*.

Guijin Son, Donghun Yang, Hitesh Laxmichand Patel, Hyunwoo Ko, Amit Agarwal, Sunghee Ahn, Kyong-Ha Lee, and Youngjae Yu. 2026b. Judging what we cannot solve: A consequence-based approach for oracle-free evaluation of research-level math. *arXiv preprint arXiv:2602.06291*.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, and 1 others. 2026. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*.

Kimi Team, Yifan Bai, Yiping Bao, Y Charles, Cheng Chen, Guanduo Chen, Haiting Chen, Huarong Chen, Jiahao Chen, Ningxin Chen, and 1 others. 2025. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*.

Shubham Toshniwal, Wei Du, Ivan Moshkov, Branislav Kisacanin, Alexan Ayrapetyan, and Igor Gitman. 2025. Openmathinstruct-2: Accelerating ai for math with massive open-source instruction data. In *International Conference on Learning Representations*, volume 2025, pages 19243–19275.

Shubham Toshniwal, Ivan Moshkov, Sean Narenthiran, Daria Gitman, Fei Jia, and Igor Gitman. 2024. Openmathinstruct-1: A 1.8 million math instruction tuning dataset. *Advances in Neural Information Processing Systems*, 37:34737–34774.

Shaotian Yan, Kaiyuan Liu, Chen Shen, Bing Wang, Sinan Fan, Jun Zhang, Yue Wu, Zheng Wang, and Jieping Ye. 2026. Distribution-aligned sequence

distillation for superior long-cot reasoning. *arXiv preprint arXiv:2601.09088*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Kaiyu Yang, Aidan Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan J Prenger, and Animashree Anandkumar. 2023. Leandojo: Theorem proving with retrieval-augmented language models. *Advances in Neural Information Processing Systems*, 36:21573–21612.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. Metamath: Bootstrap your own mathematical questions for large language models. In *International Conference on Learning Representations*, volume 2024, pages 45040–45061.

Weizhe Yuan, Jane Yu, Song Jiang, Karthik Padthe, Yang Li, Dong Wang, Ilia Kulikov, Kyunghyun Cho, Yuandong Tian, Jason Weston, and 1 others. 2026. Naturalreasoning: Reasoning in the wild with 2.8 m challenging questions. *Advances in Neural Information Processing Systems*, 38.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2024. Mammoth: Building math generalist models through hybrid instruction tuning. In *International Conference on Learning Representations*, volume 2024, pages 40320–40341.

Weiqi Zhai, Zhihai Wang, Jinghang Wang, Boyu Yang, Xiaogang Li, Xander Xu, Bohan Wang, Peng Wang, Xingzhe Wu, Anfeng Li, and 1 others. 2026. Hle-verified: A systematic verification and structured revision of humanity's last exam. *arXiv preprint arXiv:2602.13964*.

Jie Zhang, Cezara Petrui, Kristina Nikolić, and Florian Tramèr. 2026. Realmath: A continuous benchmark for evaluating language models on research-level mathematics. *Advances in Neural Information Processing Systems*, 38.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, and 1 others. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

Yifan Zhang and Team Math-AI. 2024. American invitational mathematics examination (aime) 2024.

Yifan Zhang and Team Math-AI. 2025. American invitational mathematics examination (aime) 2025.

Yifan Zhang and Team Math-AI. 2026. American invitational mathematics examination (aime) 2026.

Daniel Zheng, Ingrid von Glehn, Yori Zwols, Iuliya Beloshapka, Lars Buesing, Daniel M Roy, Martin Wattenberg, Bogdan Georgiev, Tatiana Schmidt, Andrew Cowie, and 1 others. 2026. Ai co-mathematician: Accelerating mathematicians with agentic ai. *arXiv preprint arXiv:2605.06651*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Appendix Contents

A Example Source Comparisons . . . 13
B Self-Containment Audit . . . 13
C Keyword and Judge Metric Details . . . 13
C.1 Keyword Groups . . . 13
C.2 LLM-Judge and Agent-Judge Annotations . . . 13
C.3 Aggregation . . . 14
D Near-Duplicates Filtering Details . . . 16
D.1 Pairwise Similarity Distribution . . . 16
D.2 `GPT-5.5` Judgments Near Decision Boundary . . . 16
E Reasoning Behavior Details . . . 19
E.1 Behavior-Counter Rates . . . 19
F License and Release . . . 20
G Training Details . . . 21
H Prompts . . . 21
H.1 Dataset Generation Agents . . . 21
H.2 Difficulty Comparison . . . 21
H.3 Response Generation . . . 21
H.4 Factuality Metrics . . . 21

A Example Source Comparisons

Tables 3 and 4 illustrate the source-level contrast discussed in Section 2.1. The first pair stays within number theory/arithmetic geometry; the second pairs a broad algebraic-geometry grand challenge with a narrower modern question about hyperkähler Chow rings.

B Self-Containment Audit

To quantify whether refinement makes questions usable without the source document, we run a first-pass automatic audit on 500 randomly sampled released records. For each record, Codex labels both the original extracted question and the refined standalone question as self-contained or not. A statement is counted as self-contained only if a mathematically trained reader can understand the task from the text alone, without source-local notation, missing definitions, or references to external sections, figures, or problem numbers. Table 5 shows that original extracted questions are self-contained in 67.2% of sampled cases, while refined standalone questions are self-contained in 94.2%. The refiner turns $142/500$ initially non-self-contained snippets into self-contained questions, leaving $29/500$ refined questions flagged for remaining context gaps.

C Keyword and Judge Metric Details

This appendix specifies the surface-form counters used in Section 3.2. All keyword matching is performed after lowercasing the analyzed text. A keyword group count is the sum of exact substring occurrences of all phrases in that group. The counters are descriptive trace features; they are not used as a standalone hallucination classifier.

C.1 Keyword Groups

abandon. This counter marks claims of being stuck, time-limited, or unable to complete the solution. The keyword list is: `lack of progress`, `given the time`, `time constraints`, `too complex`, `not practical manually`, `i'm stuck`, `i am stuck`, `dead end`, `can't solve`, `cannot solve`, `without progress`, `hazard a guess`, `educated guess`.

cite. This counter marks references to source objects, external databases, or citation-like artifacts. The keyword list is: `paper`, `book`, `article`, `textbook`, `monograph`, `survey`, `journal`, `proceedings`, `publication`, `arxiv`, `doi`, `wikipedia`, `mathworld`, `oeis`, `stackexchange`, `aops`, `art of problem solving`, `website`, `webpage`, `online source`.

assume. This counter combines assertive shortcuts and remembered-result language, both of which substitute confident assertion for derivation in the trace. The keyword list is: `it can be shown`, `one can show`, `it is easy to see`, `clearly`, `obviously`, `intuitively`, `by symmetry`, `must be`, `should be`, `known result`, `standard result`, `i remember`, `similar problem online`, `look it up mentally`, `the problem implies`, `strongly suggests`, `well-known`, `it is known`, `i recall`.

C.2 LLM-Judge and Agent-Judge Annotations

Agent-Judge reference verification. This annotation is applied to traces that mention citation-like references, including papers, books, articles, arXiv identifiers, DOI-like strings, named sources, or database references. A Codex-based search agent queries the internet for the mentioned reference and records whether the referenced source appears to exist. The purpose is to separate genuine provenance signals from hallucinated bibliographic support. This annotation does not judge whether the source proves the model's claim; it only checks whether the cited object itself can be found.

LLM-Judge lemma decomposition. GPT-5.5 inspects the trace and marks whether the model decomposes the problem into explicit intermediate lemmas, claims, subgoals, or cases that structure the solution. A positive annotation requires more than generic planning language: the trace should state a reusable intermediate fact or subproblem and then use it in the subsequent reasoning. The purpose is to measure constructive proof organization rather than surface verbosity.

LLM-Judge counterexample search. GPT-5.5 inspects the trace and marks whether the model actively tests a conjecture, proposed formula, candidate solution, or simplifying assumption against counterexamples, edge cases, small instances, or adversarial constructions. A positive annotation requires an explicit attempt to falsify or stress-test an idea, not merely checking arithmetic. The purpose is to measure whether the model uses skeptical reasoning before committing to a claim.

	Grand-challenge entry	Contemporary workshop-style entry
Dataset record	`google__mppc/q_001`	`07-workshop-problems/q_007`
Source	*The Millennium Prize Problems*	*Some Open Problems About Diophantine Equations*
Area	Number theory; elliptic curves and L-functions	Number theory; rational points on hyperelliptic curves
Problem form	Prove the Birch–Swinnerton-Dyer conjecture: for an elliptic curve $C/\mathbb{Q}$, the order of vanishing of $L(C, s)$ at $s = 1$ equals the Mordell–Weil rank of $C(\mathbb{Q})$.	Prove unconditionally that the genus-2 curve $$y^2 = -3x^6 - x^5 + 2x^4 + 2x^2 - 3x - 3$$ has no rational points over $\mathbb{Q}$, given that this is known assuming BSD for its Jacobian.
Status	Open	Unknown
Comment	This is the kind of classical, high-visibility challenge problem that appears in many public lists of unsolved mathematics.	This is narrower and more local: it sits near BSD and rational-points methods, but asks for a concrete unconditional proof for one curve. It better represents the workshop and survey questions that form much of RESEARCHMATH-14K.
Source	Clay monograph PDF	Leiden workshop PDF

Table 3: Side-by-side example of a classical arithmetic-geometry grand-challenge source and a narrower contemporary open-problem source represented in RESEARCHMATH-14K.

	Grand-challenge entry	Contemporary survey-style entry
Dataset record	`google__mppc/q_004`	`1002.4321/q_010`
Source	*The Millennium Prize Problems*	arXiv problem list on compact hyperkähler manifolds
Area	Algebraic geometry; Hodge theory and algebraic cycles	Algebraic geometry; hyperkähler manifolds and Chow rings
Problem form	Prove the Hodge conjecture: every rational Hodge class on a smooth projective complex variety is a $\mathbb{Q}$-linear combination of cohomology classes of algebraic subvarieties.	For every projective hyperkähler manifold X, prove that the subring of $\mathrm{CH}(X)$ generated by $\mathrm{Pic}(X)$ and the Chern classes of X injects into cohomology under the cycle-class map.
Status	Open	Partially solved
Comment	This is a universal conjecture about the relation between Hodge-theoretic and algebraic cycles across all smooth projective complex varieties.	This is much more local: it asks for a specific Chow-ring injectivity statement in the modern setting of hyperkähler geometry. It uses current objects and tools while still sitting near the Hodge/cycle-theoretic grand challenge.
Source	Clay monograph PDF	arXiv:1002.4321

Table 4: Second side-by-side example: a broad algebraic-geometry grand challenge compared with a narrower contemporary hyperkähler/Chow-ring problem represented in RESEARCHMATH-14K.

Audit item	Count	Rate
Original extracted question	$336/500$	67.2%
Refined standalone question	$471/500$	94.2%
Original no, refined yes	$142/500$	28.4%
Original yes, refined no	$7/500$	1.4%

Table 5: **Automatic self-containment audit.** We sample 500 released records and ask Codex to label both the original extracted question and the refined standalone question as self-contained or not.

C.3 Aggregation

For each keyword group, LLM-Judge annotation, and Agent-Judge verification result, we aggregate by model, model family, and benchmark. The row-hit rate treats each trace as a binary hit for a counter; for judged annotations, this is the fraction of traces marked positive. The benchmark trend view reports the average newer-minus-older delta across the DeepSeek, Kimi, and Qwen comparison pairs.

Case	Original extraction judged not self-contained	Refined standalone problem
Geodesics	Suppose $\mathrm{dist}(\alpha(t), \beta(t)) \to 0$ as $t \to -\infty$.	Let (M, g) be a complete, simply connected Riemannian manifold without conjugate points. Let $\alpha, \beta : \mathbb{R} \to M$ be two distinct unit-speed geodesics, and denote the Riemannian distance by dist. Suppose that $\mathrm{dist}(\alpha(t), \beta(t)) \to 0$ as $t \to -\infty$. Does it follow that $\mathrm{dist}(\alpha(t), \beta(t)) \to \infty$ as $t \to +\infty$?
Words	Let Σ be an alphabet and let $w \in \Sigma^*$ be a p-free word. Then w is a factor of a maximal p-free word over Σ.	Let Σ be a finite alphabet and let Σ^* denote the set of finite words over Σ. A pattern is a non-empty word p over an alphabet of variables. An occurrence of p in a word $w \in \Sigma^*$ is a factor of w of the form $h(p)$, where h is a non-erasing morphism. A word w is p-free if no factor of w is an occurrence of p. A p-free word is maximal if it cannot be extended on the left or right while remaining p-free. Prove or disprove: for every alphabet Σ, every pattern p, and every p-free word $w \in \Sigma^*$, there exists a maximal p-free word over Σ that contains w as a factor.
Remaining gap	Fix $p = c/n$ with $c > 0$. Is it true that, with high probability for $G_{n,p}$, the width $s(G_{n,p})$ of the independence complex satisfies $s(G_{n,p}) = (1 + o(1)) \max_r \lvert Q(G_{n,p})^{(r)} \rvert$?	Let G be a finite simple graph with independence complex $I(G)$, whose faces are the independent sets of G. Following Meshulam's recursive framework, one associates to G a nonnegative integer parameter $s(G)$ and a finite sequence $Q(G) = (Q(G)^{(0)}, Q(G)^{(1)}, \ldots)$ produced by the standard recursive process used to bound the topological connectivity of $I(G)$. For fixed $c > 0$ and $G_{n,p}$ with $p = c/n$, is it true with high probability that $s(G_{n,p}) = (1+o(1)) \max_r \lvert Q(G_{n,p})^{(r)} \rvert$? The refined version still depends on the source for the exact recursive definition of $Q(G)$ and $s(G)$.

Table 6: Examples from the self-containment audit. The first two rows show originally non-self-contained extractions that become self-contained after refinement. The final row shows a remaining failure case in which both the original extraction and refined statement still depend on source-local definitions.

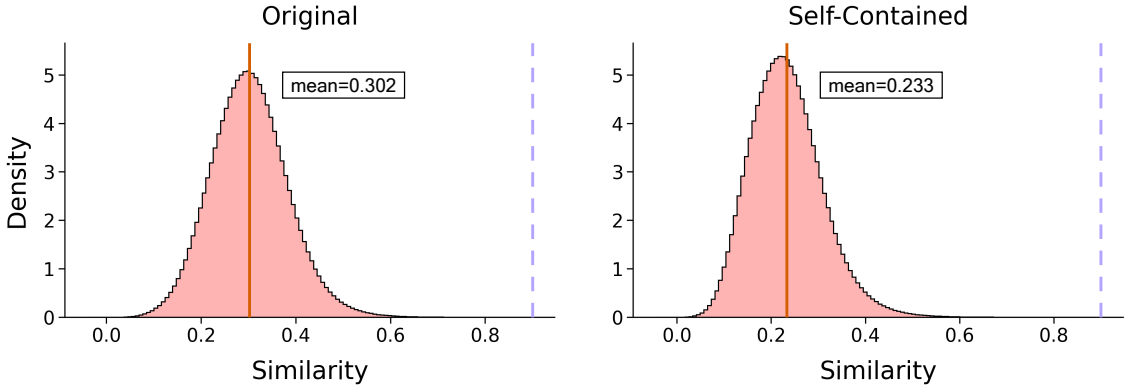

Figure 6: **Pairwise Similarity Distribution.** Empirical distributions of pairwise embedding similarities across all 98,778,540 problem pairs. Dashed vertical lines indicate the maximum similarity values, both below the 0.90 duplicate threshold. *Left*: similarities between original statements. *Right*: similarities between self-contained rewrites.

D Near-Duplicates Filtering Details

D.1 Pairwise Similarity Distribution

Figure 6 reports the pairwise embedding similarity distributions for the original statements and the self-contained rewrites in RESEARCHMATH-14K.

D.2 `GPT-5.5` Judgments Near Decision Boundary

Tables 7, 8, and 9 present problem pairs with similarity scores close to the 0.9 threshold that `GPT-5.5` nevertheless judged to be distinct. These cases support our conservative threshold choice.

GPT-5.5 Judgment Example 1 ($similarity = 0.89967543$)

[Problem 1]
Let S be a convex body in R^3 (a compact convex set with non-empty interior). A convex body K ⊂ R^3 is called S-Kakeya if, for every rotation ρ ∈ SO(3), K contains a congruent copy ρ(S) + v of S (for some translation vector v depending on ρ).

Say that S has the Kakeya rotation property if, for every convex S-Kakeya body K ⊂ R^3 and every two congruent copies S_1, S_2 of S contained in K, there exists a continuous path of congruent copies of S, all contained in K, starting at S_1 and ending at S_2 (so that any two copies of S inside K can be continuously rotated and translated into one another while remaining inside K).

Does there exist a convex body S in R^3 that fails the Kakeya rotation property? Equivalently, can one find S ⊂ R^3 and a convex S-Kakeya body K ⊂ R^3 containing two congruent copies of S that cannot be joined by a continuous family of congruent copies of S lying inside K?

[Problem 2]
Let d ≥ 2 be an integer. For a convex body S ⊂ ℝ^d (a compact convex set with non-empty interior, or, more generally, including line segments), say that a convex body K ⊂ ℝ^d is S-Kakeya if, for every rotation ρ ∈ SO(d), K contains a translate of ρ(S). Say that S has the Kakeya rotation property if, for every S-Kakeya convex body K ⊂ ℝ^d and every two congruent copies S_1, S_2 of S contained in K, there exists a continuous family (S_t)_{t ∈ [0,1]} of congruent copies of S, all contained in K, with S_0 = S_1 and S_1 = S_2.

It is known that, in every dimension d ≥ 2, line segments have the Kakeya rotation property, and that d-dimensional Euclidean balls trivially have this property.

Question: For some dimension d ≥ 3, does there exist a convex body S ⊂ ℝ^d, other than a line segment or a d-dimensional ball, that has the Kakeya rotation property?

[GPT-5.5]
They are **different problems**.

Problem 1 asks whether there exists a convex body (S \subset \mathbb{R}^3) that **fails** the Kakeya rotation property.

Problem 2 asks whether, in some dimension (d \ge 3), there exists a nontrivial convex body (S) that **has** the Kakeya rotation property.

So they ask about opposite kinds of examples: failure vs existence of nontrivial success.

Table 7: GPT-5.5 Judgment Example 1

GPT-5.5 Judgment Example 2 ($similarity = 0.8989585$)

[Problem 1]
Consider the quadratic polynomial family
$$P_\alpha(z) = e^{2\pi i \alpha} z + z^2, \qquad z \in \mathbb{C},$$
where $\alpha \in \mathbb{R}\setminus\mathbb{Q}$. Recall that an irrational number α with continued fraction convergents p_n/q_n is called a Brjuno (Bruno) number if
$$\sum_{n=0}^{\infty} \frac{\log q_{n+1}}{q_n} < \infty.$$
By a theorem of Brjuno–Yoccoz, for every Brjuno number α the polynomial P_α is locally linearizable at the fixed point 0; that is, there is a maximal open neighborhood S_α of 0 (the Siegel disk of P_α) on which P_α is conformally conjugate to the rigid rotation $z \mapsto e^{2\pi i \alpha} z$.

Question: Does there exist a Brjuno number α such that the boundary ∂S_α of the Siegel disk of P_α is not a Jordan curve (i.e., is not homeomorphic to a circle)?

[Problem 2]
For an irrational real number α, consider the quadratic polynomial
$$P_\alpha(z) = e^{2\pi i \alpha} z + z^2.$$
The Bruno series of α is defined by $B(\alpha) = \sum_{n \ge 0} \frac{\log q_{n+1}}{q_n}$, where p_n/q_n are the convergents of the continued fraction expansion of α; the number α is called a Bruno number if $B(\alpha) < \infty$. By a theorem of Bruno and Yoccoz, α is a Bruno number if and only if P_α is analytically linearizable near 0, i.e., there is a holomorphic injection φ from a neighborhood of 0 in $\mathbb{C}$ to $\mathbb{C}$ with $\varphi(0)=0$, $\varphi'(0)=1$, and $\varphi^{-1} \circ P_\alpha \circ \varphi(z) = e^{2\pi i \alpha} z$. Let S_α denote the maximal such linearization domain (the Siegel disk of P_α around 0).

Question: Does there exist a Bruno number α such that the boundary $\partial S_\alpha \subset \mathbb{C}$ is a C^∞ embedded circle (i.e., a C^∞ submanifold of $\mathbb{C}$ diffeomorphic to S^1)?

[GPT-5.5]
They are **different problems**.

Both use the same quadratic family and the same Bruno/Brjuno condition, but the questions ask different things:

* **Problem 1:** asks whether some Siegel disk boundary is **not a Jordan curve**.
* **Problem 2:** asks whether some Siegel disk boundary is a **(C^\infty) embedded circle**, which is much stronger than being

```
      Jordan.

So they are not the same; they ask essentially opposite types of boundary regularity.
```

Table 8: GPT-5.5 Judgment Example 2

GPT-5.5 Judgment Example 3 ($similarity = 0.89961576$)

```
[Problem 1]
Let $D=\{z\in\mathbb{C}:|z|<1\}$ be the open unit disc, and let $n\ge 3$ be an integer. Consider the
family
$$\mathcal{F} = \{\, f : f \text{ is meromorphic on } D \text{ and } f'(z)\,f(z)^n \neq 1 \text{ for
   every } z\in D\,\}.$$
Prove or disprove: the family $\mathcal{F}$ is normal on $D$ (in the sense of Montel, with respect
to the spherical metric); that is, every sequence in $\mathcal{F}$ has a subsequence which converges
locally uniformly on $D$ to a meromorphic function or to the constant $\infty$.

[Problem 2]
Let $\mathbb{D}=\{z\in\mathbb{C}:|z|<1\}$ denote the open unit disc, and fix an integer $n\ge 1$.
Let $\mathcal{F}_n$ denote the family of all functions $f$ that are holomorphic in $\mathbb{D}$ and
satisfy
\[ f'(z)\,f(z)^{n}\neq 1 \qquad\text{for every } z\in\mathbb{D}. \]
A family of holomorphic functions on $\mathbb{D}$ is called normal if every sequence in the family
has a subsequence that converges locally uniformly on $\mathbb{D}$ either to a holomorphic function
or to $\infty$.

Prove or disprove: for every integer $n\ge 1$, the family $\mathcal{F}_n$ is normal in $\mathbb{D}$.

[GPT-5.5]
They are **different problems**.

Key differences:

1. **Function class:**
   Problem 1 allows **meromorphic** functions; Problem 2 only allows **holomorphic** functions.

2. **Range of (n):**
   Problem 1 assumes (n \ge 3); Problem 2 asks for **all (n \ge 1)**.

3. **Normality target:**
   Problem 1 allows subsequential limits to be **meromorphic or (\infty)** in the spherical metric.
   Problem 2 asks for holomorphic normality, with limits **holomorphic or (\infty)**.

So they are related, but **not the same problem**.
```

Table 9: GPT-5.5 Judgment Example 3

Benchmark	assume rows	cite rows	abandon rows	Lemma-decomposition rows
RESEARCHMATH-14K	677/720 (94.0%)	482/720 (66.9%)	125/720 (17.4%)	11/720 (1.5%)
Leipzig Tier-4	666/720 (92.5%)	461/720 (64.0%)	200/720 (27.8%)	3/720 (0.4%)
SOOHAK	651/688 (94.6%)	443/688 (64.4%)	161/688 (23.4%)	4/688 (0.6%)
HLE-Verified	673/720 (93.5%)	272/720 (37.8%)	141/720 (19.6%)	–
AIME	660/720 (91.7%)	72/720 (10.0%)	70/720 (9.7%)	–
Research-level total	1,994/2,128 (93.7%)	1,386/2,128 (65.1%)	486/2,128 (22.8%)	18/2,128 (0.85%)

Table 10: Absolute behavior-counter row-hit rates across the eight paper models. The assume, cite, and abandon counters correspond to the three rule-based keyword groups defined in Appendix C.1. Lemma-decomposition rows are judged separately by the Agent-Judge and are available for the three research-level benchmarks.

E Reasoning Behavior Details

E.1 Behavior-Counter Rates

Table 10 reports the fraction of traces in each benchmark that trigger each behavior counter across the eight evaluated models.

F License and Release

We release the RESEARCHMATH family under the MIT License. The release covers two artifacts: RESEARCHMATH-14K, the corpus of 14,056 research-level mathematical problems described in Section 2, and RESEARCHMATH-REASONING, the 220K reasoning trajectories described in Section 2.4. The MIT License permits unrestricted use, modification, and redistribution for both research and commercial purposes, with the only requirement being preservation of the copyright notice. We chose this license to maximize the resource's utility for follow-up work on research-level mathematical reasoning, including training, evaluation, and behavioral analysis.

Both artifacts derive from publicly available academic sources (arXiv preprints, open-problem web pages, and workshop or conference problem sheets). The Extractor agent discards any document hidden behind a paywall before extraction (Section 2.1); paywalled or restricted sources are not represented in the released data. The released model trajectories may contain fabricated references, unsupported claims, and other artifacts documented in Section 4; we release them as-is rather than filter them at the source, so that future work can apply alternative filtering or analysis pipelines. Each problem record carries provenance metadata (source URL or identifier, document title where available) sufficient to trace it back to the original document.

We do not release the filtered 5,000-trace training subset (RESEARCHMATH-REASONING-FILTERED) as a separate named artifact, since it is a function of RESEARCHMATH-REASONING and the publicly described filtering pipeline. Researchers can reproduce it by running the rule-based and agent-based filters from Section 3.2 on RESEARCHMATH-REASONING subject to their own compute budget.

G Training Details

All fine-tuning runs use LoRA on top of three Qwen3 base models (`Qwen3-4B-base`, `Qwen3-8B-base`, `Qwen3-30B-A3B-base`) on $5{,}000$ randomly sampled traces from either filtered RESEARCHMATH-14K or the DASD-Thinking control. Each setting is run with three seeds and the reported numbers are averages over the runs.

LoRA configuration. Rank $r = 64$, alpha $\alpha = 128$, dropout 0.05, no bias, applied to the attention and MLP projections of each transformer block (`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`). The language-model head is intentionally left out of the adapter set, since training it together with chunked-NLL loss is incompatible.

Optimization. Cross-entropy loss is computed with the chunked-NLL kernel from `trl`, which avoids materializing the full V-dimensional logit tensor and is what makes long-context training feasible on 24 GB GPUs. We use `adamw_torch_fused` with learning rate 1×10^{-4}, cosine schedule, warmup ratio 0.03, 2 epochs, and `bfloat16` mixed precision. Gradient checkpointing is enabled when FSDP activation checkpointing is not already on.

Batching. Per-device batch size is 1; global batch size is 16 for the 30B run and 32 for the smaller models. Gradient accumulation is computed as $\lceil \text{global}/(\text{per-device} \times \text{GPUs}) \rceil$. Sequence packing is enabled.

Sequence length. Examples are truncated at $24{,}512$ tokens for the 4B/8B runs and $32{,}768$ tokens for the 30B run. Longer traces are truncated rather than dropped.

Infrastructure. All runs use $4\times$ RTX 3090 (24 GB) via FSDP with SDPA attention. Wandb logs the full hyperparameter set per run for reproducibility.

H Prompts

H.1 Dataset Generation Agents

Table 11 and Table 12 present the prompts used for the Extractor and Refiner agents in Section 2.1, respectively.

H.2 Difficulty Comparison

Table 13 shows the prompt for difficulty comparison in Section 2.3.

H.3 Response Generation

Table 14 shows the prompt used to generate model responses in Section 2.4.

H.4 Factuality Metrics

Tables 15, 16, and 17 present the prompts used for the factuality metric in Section 3.2. Table 17 gives the short-block variant of Table 16, which is used for detected blocks shorter than 200 characters with additional surrounding reasoning context.

Prompt for the Extractor Agent

```
---
name: paper-question-parser
description: "Parse one paper or scholarly open-problem source (local PDF path, arXiv URL, direct PDF URL, DOI URL, HAL/DBLP/
      zbMATH/OpenAlex/Crossref landing page, workshop page, MathOverflow thread, or domain-specific Q&A forum), resolve it to
      readable full text, extract open/unsolved questions, rewrite them into self-contained form with evidence, dedupe,
      quality-gate, and auto-save structured JSON. Triggers: 'parse paper questions', 'extract open problems', 'unsolved
      questions from paper', 'parse open-problem page'."
---

# Paper Question Parser

You run a single-source workflow to extract open/unsolved research questions from a paper, workshop problem sheet,
      bibliographic landing page that resolves to full text, or scholarly problem page, and rewrite each into a self-contained
      , evidence-grounded question.

## Path Configuration
- **RUN_DIR**: `outputs/parse-paper/`
- **LATEST_FILE**: `outputs/latest.json`
- **TEMP_DIR**: `.sisyphus/paper-question-parser/tmp/`

## Operating Contract

- Process exactly one source per run.
- Accept either a local PDF path or one URL pointing to a scholarly source. Supported URL families include arXiv `abs`/`pdf`,
      direct PDFs, DOI URLs, HAL pages, DBLP/Crossref/OpenAlex/zbMATH landing pages, workshop/problem-session pages, and
      MathOverflow threads.
- Resolve URL inputs to a readable full-text artifact before extraction.
- Do not ask the user to run scripts.
- Return STRICT JSON only (no prose) with keys: `source`, `accepted`, `needs_review`, `trace`.
- Auto-save the final JSON result to disk every run.

## Input Normalization

1. Read `$ARGUMENTS` and extract a single source.
2. If input is a local PDF path:
   - Verify the file exists.
   - Use that absolute path as `paper_path`.
   - Set:
     - `source_kind = "pdf"`
     - `source_locator = paper_path`
     - `source_name = <local_pdf_stem>`
     - `citation_mode = "page_number"`
3. If input is a URL:
   - Normalize obvious wrappers while preserving the identity of the same source.
   - Resolve the URL in this order:
     1. If the URL is `https://arxiv.org/abs/<id>`, convert it to `https://arxiv.org/pdf/<id>.pdf`.
     2. If the URL is already a direct PDF URL, or the resolved response is a PDF, download it internally with Bash:
        - `mkdir -p TEMP_DIR`
        - `curl -L --fail "<pdf_url>" -o "TEMP_DIR/<paper_slug>.pdf"`
        - set `paper_path` to the downloaded file
        - set:
          - `source_kind = "pdf"`
          - `source_locator = paper_path`
          - `source_name = <arxiv_id_or_pdf_slug>`
          - `citation_mode = "page_number"`
     3. If the URL is a DOI URL or a bibliographic/landing page (for example Crossref, OpenAlex, DBLP, zbMATH, HAL, or a
      publisher landing page), inspect the page and follow the most specific full-text link available:
        - preferred targets: `PDF`, `Download article`, `Download extract`, `document`, `unpaywalled`, `external edition`, or
      an explicit article page whose visible body contains the problems
        - treat DBLP, Crossref, OpenAlex, and zbMATH primarily as resolvers, not as the technical source
        - for HAL, prefer an explicit `/document` or file-download link when available
        - if the landing page or download endpoint returns a missing-resource page, paywall page, or metadata-only page, query
       source-specific metadata APIs when available to recover the exact title and candidate readable copies of that same work
      :
          - HAL: `https://api.archives-ouvertes.fr/search/?q=halId_s:<id>&rows=1&fl=title_s,uri_s,openAccess_bool,submitType_s
      ,label_xml&wt=json`
          - Crossref: `https://api.crossref.org/works/<doi>`
          - OpenAlex: `https://api.openalex.org/works?filter=doi:https://doi.org/<doi>&select=display_name,doi,open_access,
      best_oa_location,primary_location`
        - use API metadata only to identify the exact work, discover candidate full-text URLs, and detect access restrictions;
       do not extract open problems from API metadata, abstracts, or resolver records
        - only follow candidate URLs from APIs when the destination still matches the same DOI or the same exact title
     4. If the URL is a workshop/problem-session/forum page whose own body contains the relevant problem statements, use the
      page itself as the primary source.
        - examples: AIM pages, workshop problem-session pages, MathOverflow threads, Stack Exchange sites (e.g., Physics SE,
      Biology SE, Cross Validated), domain-specific Q&A forums
        - for Q&A forum sources, prefer the question body as the primary source; use answers/comments only for explicit
      resolution claims or clarifications that are needed and clearly attributable
     5. If direct fetching fails because of redirects, robots/SSL issues, transient HTML fetch failures, or citation-only
      landing pages, search for the same source by exact title, DOI, repository identifier, or trailing URL slug and recover a
       readable artifact for that same work/source. Do not switch to a different work.
   - If the resolved primary source is HTML rather than PDF, set:
```

```
    - ‘source_kind = "html"‘
    - ‘source_locator = <canonical URL of the chosen page>‘
    - ‘source_name = <arxiv_id_or_doi_slug_or_url_slug>‘
    - ‘citation_mode = "locator_string"‘

## Source Metadata

- Every final JSON object must include a top-level ‘source‘ object.
- ‘source.url‘ must preserve the original user-supplied source locator:
  - if input was a URL, store that original URL
  - if input was a local PDF path, store the absolute local path
- ‘source.resolved_locator‘ must record the canonical URL or local file path actually used for extraction.
- ‘source.title‘ must be a non-empty source title.
- ‘source.title_origin‘ must be one of:
  - ‘source_text‘
  - ‘resolver_metadata‘
  - ‘ai_extracted‘
  - ‘filename‘
- Determine ‘source.title‘ in this order:
  1. Prefer the visible title from the chosen primary source itself:
     - PDF title page, first page heading, or explicit article/workshop title
     - HTML ‘<title>‘, ‘<h1>‘, main heading, or obvious visible page title
  2. If the chosen primary source lacks a reliable visible title, use trusted resolver metadata for the same exact work/source
     .
  3. If no reliable title is available, infer a concise descriptive title from the readable source body and set ‘title_origin
     = "ai_extracted"‘.
  4. If the source is unreadable and no better title is available, fall back to the local filename stem or URL slug and set ‘
     title_origin = "filename"‘.
- Never leave ‘source.title‘ blank.

## Source Resolution Rules

- Prefer the direct PDF of the exact source document when available.
- Otherwise use full HTML text whose visible body actually contains the problem statements.
- Otherwise use the closest source-authorized readable copy.
- Treat direct-fetch failures such as robots-blocked pages, broken TLS, redirect loops, and citation-only HTML as resolver
    failures, not as proof that the source is unusable.
- Metadata APIs may be used for discovery, title recovery, DOI verification, and access-status checks, but never as the
    technical source for extraction.
- If a resolver or API confirms the record is metadata-only or restricted with no readable full text (for example a HAL ‘
    notice‘ record with ‘openAccess_bool=false‘), stop retrying equivalent resolver URLs and move the item to ‘needs_review‘
     unless you can recover another exact-work readable copy.
- Candidate links discovered in resolver metadata, API fields, or publisher abstracts must be treated as hints only; follow
    them only if the resulting page or file still matches the same DOI or the same exact title.
- Do not rely on abstract-only pages, citation-only pages, search snippets, or index metadata as the technical source when
    they do not contain the problem statements themselves.
- Do not combine multiple separate works into one run. If a resolver page exposes multiple editions, choose one canonical
    readable source and stay with it.

## Access-Failure Policy

For batch runs, resolve access problems with a bounded recovery pass. Prefer a clear ‘error_input‘ checkpoint over a long
    speculative search.

When a page has access issues, classify the failure mode first:

- ‘network_or_dns_failure‘: shell/browser fetch cannot resolve or connect
- ‘blocked_or_challenged‘: Cloudflare, JavaScript/cookie challenge, 403/401, robot block, or login wall
- ‘paywalled_or_restricted‘: publisher landing page or resolver confirms no readable full text
- ‘metadata_only‘: resolver, abstract, citation page, table of contents, index entry, or review page with no problem
    statements
- ‘missing_or_invalid_artifact‘: download endpoint returns missing-resource HTML, a corrupt file, or a non-readable PDF
- ‘ambiguous_resolution‘: multiple plausible works are found and none can be confirmed as the same exact source

Bounded recovery steps:

1. If the source has a stable identifier, try identifier-specific recovery first: arXiv PDF, DOI landing page, Crossref,
    OpenAlex, HAL API, zbMATH metadata, or explicit resolver links.
2. If metadata provides a DOI, exact title, year, and authors, use them only to find a readable artifact for that same work.
3. Run at most three exact-work search attempts after API/resolver recovery. Use precise queries such as exact DOI, exact
    title plus ‘PDF‘, exact title plus first author, or repository identifier. Avoid broad topical searches.
4. Inspect at most five candidate pages or files from those searches. Accept a candidate only when the visible title or DOI
    confirms the same exact work/source.
5. Stop immediately once remaining candidates are only snippets, abstracts, citation metadata, paywalls, blocked pages, or
    different works.

Failure output rules:

- Do not extract questions from metadata, abstracts, search snippets, title text, or related papers.
- Do not keep retrying equivalent blocked resolver URLs once the same failure mode is established.
- If no readable exact-work PDF or full HTML source is recovered within the bounded pass, return the standard ‘error_input‘
    JSON with ‘accepted: []‘.
- Set ‘source.title‘ from source text when available, otherwise trusted resolver/API metadata, otherwise the URL slug or
    filename.
- Leave ‘source.resolved_locator‘ empty unless a readable primary source artifact was actually used for extraction.
- In ‘trace‘, include the failure mode, identifiers attempted, and why the recovered candidates were rejected.
```

1132

Source-Family Hints

- **AIM / AIMPL pages**: if the exact URL does not fetch cleanly, search the AIM Problem Lists site by the trailing slug and exact visible title, then use the matching list page that contains the problem statements. If the root page is only an index, follow the numbered internal subpage that actually lists the problems.
- **Legacy HAL hosts**: normalize old ‘hal-*.ccsd.cnrs.fr‘ pages to their canonical ‘hal.science‘ record when possible; if needed, search by the HAL identifier and prefer a ‘document‘ or direct file URL. If ‘/document‘ returns a missing-resource page, query the HAL API by identifier. Treat records such as ‘submitType_s="notice"‘ with ‘openAccess_bool=false‘ as metadata-only dead ends unless another exact-work readable copy is found.
- **DBLP / zbMATH / OpenAlex / Crossref**: treat these as bibliographic resolvers. Prefer explicit links labeled ‘unpaywalled version‘, ‘electronic edition‘, ‘external edition‘, ‘PDF‘, ‘document‘, ‘arXiv‘, or DOI targets that lead to readable full text. If the resolver page still exposes only metadata, use the exact title, DOI, year, and author metadata to recover an accessible copy of that same work.
- **OpenAlex**: prefer ‘best_oa_location.pdf_url‘, then ‘best_oa_location.landing_page_url‘, then ‘primary_location‘ when they correspond to the same exact work. If ‘open_access.is_oa=false‘ and there is no repository or PDF location, treat OpenAlex as identity metadata only.
- **Crossref**: use Crossref to recover the exact title, DOI, and candidate resource URLs. Treat ‘resource.primary.URL‘, ‘link []‘, and any full-text URL embedded in the abstract only as candidate leads, and only follow them when the resolved destination still matches the same DOI or exact title.
- **DOI URLs**: if the DOI landing page is paywalled or metadata-only, query Crossref and OpenAlex by DOI before doing a generic exact-title search. Prefer repository copies, accepted manuscripts, or author-posted PDFs of the same work.
- **zbMATH**: if the zbMATH page itself is access-denied or metadata-only, use the zbMATH identifier plus the exact title/year/author metadata to search for the same work elsewhere. If no readable exact-work copy exists, move the item to ‘needs_review‘ rather than extracting from bibliographic metadata.

If input is invalid or cannot be downloaded, return:

‘‘‘json
```
{
  "source": {
    "url": "<original_input_url_or_path>",
    "resolved_locator": "",
    "title": "<best_effort_slug_or_filename>",
    "title_origin": "filename"
  },
  "accepted": [],
  "needs_review": [
    {
      "id": "error_input",
      "reason": "invalid_or_unreadable_input",
      "question_text_raw": "",
      "evidence": []
    }
  ],
  "trace": [
    {
      "id": "error_input",
      "stage": "input",
      "notes": "Input could not be normalized to a readable PDF or HTML full-text source.",
      "evidence_refs": []
    }
  ]
}
```
‘‘‘

Workflow Stages

Stage 1: Map

- If ‘source_kind = "pdf"‘, use ‘look_at(file_path=paper_path, goal=...)‘ to extract:

  - section hierarchy (major sections and subsections)
  - notation/terminology/definitions index
  - key result/claim/finding reference index (theorems, propositions, lemmas, experiments, algorithms, models, case studies, and similar formal structures, with labels and where they appear)
  - likely "open problems/questions" regions

- If ‘source_kind = "html"‘, use the available web/page-reading tools on ‘source_locator‘ to extract:
  - section hierarchy or page structure (headings, numbered problem blocks, post/answer/comment blocks, list items)
  - notation/terminology/definitions index when present
  - likely "open problems/questions" regions
  - stable locators for later citation, such as section titles, numbered items, and HTML line spans

Record this as ‘doc_map‘ in working memory and add a ‘trace‘ entry with stage ‘map‘.

Stage 2: Candidate Extraction (Recall-First)

Iterate through all the sections in ‘doc_map‘ and extract ALL explicit and implicit open problems/questions.

Extraction target:

- include markers like "Question", "Problem", "Open", "Unknown whether", "Is it true that"
- include bullet lists, numbered problem sessions, named open-problem blocks, and forum-style top-level question statements when they are clearly research questions in the field
- include entries labeled as solved (e.g., ‘(Solved) problem ...‘) and keep them in output

```
- include implicit question statements that represent unresolved research problems
- for forum-style sources, extract the main post and any explicitly separated subquestions; do not mine side comments or
      answers as separate research problems unless they clearly formulate one
- for each candidate, capture:
  - raw question text (as close to source as possible)
  - page number(s) or stable HTML locator(s)
  - verbatim supporting quote(s)
  - solved-status metadata derived from source labels (for example ‘(Solved)‘, ‘Solved‘, or equivalent)

Add one ‘trace‘ entry per section with stage ‘extract_candidates‘.

### Stage 3: Self-Contained Rewrite (Precision-First)

For each candidate:

- rewrite into a standalone question understandable without the source
- ‘question_text‘ is the primary payload and must by itself be fully self-contained for a reader who has not seen the source
      or any other extracted question
- ‘question_text‘ should be as long as needed to inline all notation, terminology, definitions, assumptions, experimental
      conditions, datasets, models, evaluation metrics, domain context, and other problem data required for a fully usable
      standalone statement
- for extracted questions, prefer a clearly longer rewrite than the original source wording whenever that wording is too
      compressed to stand alone; the goal is faithful expansion, not brevity
- prefer over-explaining rather than under-explaining when deciding whether to inline definitions; if a careful reader could
      not start solving without source-specific terminology being unpacked, unpack it in ‘question_text‘
- ‘question_text‘ may be a multi-sentence or multi-paragraph JSON string, but it must still read as one self-contained
      question rather than disconnected notes
- avoid source-pointing text in ‘question_text‘ such as "in the paper", "in this section", "as defined above", "the authors
      define", or similar phrasing that gestures at the source instead of stating the needed content directly
- ‘context_brief‘ is only a short label and must never carry definitions that are required to understand the problem
- resolve references like "Proposition 2.1", "Experiment 3", "Table 2", "Algorithm 1", "Figure 5", "Section 3", "Problem 4",
      or "the accepted answer" only using evidence in the chosen primary source artifact
- if required context is missing in evidence, do NOT guess; move item to ‘needs_review‘

Add one ‘trace‘ entry per candidate with stage ‘rewrite‘.

### Stage 4: Dedupe

- merge near-duplicate rewritten questions
- preserve all evidence references from merged members in ‘trace‘
- keep one canonical ‘id‘ per merged cluster

Add one ‘trace‘ entry per candidate with stage ‘dedupe‘.

### Stage 5: Quality Gates

Run deterministic gates:

1. Self-containedness gate:
   - no dangling references like "this", "above", "as discussed", "see Section X" unless expanded
   - if a reader would need source-specific notation, terminology, definitions, experimental conditions, datasets, evaluation
      criteria, or domain-specific context to understand the question, include them directly in ‘question_text‘ when supported
       by evidence
   - ‘question_text‘ must not rely on source-pointing language such as references to the paper, section, source text, or prior
       definitions instead of restating the content directly
2. Evidence coverage gate:
   - accepted items must include at least one evidence quote with a page number or stable HTML locator
3. Evidence completeness gate
   - accepted items must include complete sentences or a contiguous self-contained problem statement as evidence; otherwise,
      move them to ‘needs_review‘
4. No-new-facts gate:
   - rewritten content, including any added definitions or settings in ‘question_text‘, must not introduce unsupported claims
5. Source adequacy gate:
   - do not accept items extracted only from abstracts, citation metadata, link labels, thread titles, or search snippets when
       the technical problem statement is not visible in the chosen source text
6. Schema conformance gate:
   - accepted items must match the JSON output schema exactly and contain all required fields

Any failed item goes to ‘needs_review‘ with specific ‘reason‘.
Add one ‘trace‘ entry per candidate with stage ‘quality_gates‘.

## Required Prompt Clauses

## EVIDENCE_CITATION

- Every extracted or rewritten question must be backed by verbatim quote evidence and a source locator.
- Evidence format: ‘{ "page": <number_or_string>, "quote": "<verbatim text>" }‘
- Use numeric page numbers for PDFs.
- Use stable locator strings for HTML/page/thread sources, such as ‘"HTML lines 57-58"‘, ‘"Problem 3, HTML lines 23-28"‘, or
      ‘"Accepted answer lines 167-169"‘.
- If evidence cannot be located, do not accept the item.

## JSON_SERIALIZATION

- The final payload must be valid JSON that parses without repair.
```

- Escape every backslash inside JSON strings. This matters especially for sources with domain-specific markup such as TeX/LaTeX (mathematics), chemical formulas, code snippets, pseudocode, or structured notation (e.g., MathOverflow, arXiv HTML, bioRxiv).
- In ‘question_text‘, prefer plain-text or Unicode notation over raw domain-specific markup (e.g., TeX/LaTeX, chemical notation, pseudocode) when either would be equally faithful.
- In ‘evidence.quote‘, keep the source text verbatim, but serialize it as a valid JSON string with required escaping.
- Before returning and before saving, check that the exact payload would parse as JSON; if not, repair the escaping first.

NO_NEW_FACTS

- Do not invent facts, assumptions, constraints, or definitions.
- Do not infer missing technical statements beyond provided evidence.
- When expanding a question to make it self-contained, only inline definitions, notation, assumptions, or ambient-setting details that are directly supported by the chosen source.
- Do not treat citation metadata, abstracts, or search snippets as sufficient technical evidence unless the chosen source page itself states the full problem there.
- If missing context is required to make the question self-contained, move to ‘needs_review‘.

NEEDS_REVIEW_FLAG

Move candidate to ‘needs_review‘ when any of these hold:

- unresolved cross-reference (proposition, theorem, experiment, figure, table, algorithm, or section not recoverable from evidence)
- ambiguous pronoun/deixis with unclear antecedent
- insufficient quote evidence for a rewritten claim
- undefined source-specific terminology remains in ‘question_text‘
- ‘question_text‘ points back to the source instead of stating the needed content directly
- potential merge conflict between similar but distinct problems
- source resolves only to abstract/citation metadata without the full problem text
- resolver/API metadata confirms the record is restricted, notice-only, or lacks any readable full-text copy of the same work
- forum/workshop page is not self-contained enough and missing context cannot be recovered from the same source
- resolver chain is ambiguous between multiple possible full-text sources

SOLVED_STATUS_METADATA

- Do not drop candidates just because they are labeled solved.
- Every accepted question row must include solved-status metadata.
- Metadata format in each accepted row:
  - ‘”meta”: { ”is_solved”: <boolean> }‘
- Set ‘is_solved=true‘ only when the chosen primary source explicitly marks the item as solved or explicitly resolves it in the same source artifact.
- Otherwise set ‘is_solved=false‘.

Output Schema (STRICT JSON)

Return exactly one JSON object:

‘‘‘json
```
{
  "source": {
    "url": "https://example.org/problem-page",
    "resolved_locator": "https://example.org/problem-page",
    "title": "Example Problems Page",
    "title_origin": "source_text"
  },
  "accepted": [
    {
      "id": "q_001",
      "question_text": "Fully self-contained open question text. It may be substantially longer than the original source wording when needed to include notation, terminology, definitions, assumptions, experimental conditions, datasets, evaluation criteria, and domain context required to understand the problem without the source.",
      "context_brief": "Short topic label only; not a place to hide required definitions.",
      "meta": {
        "is_solved": false
      },
      "evidence": [
        {
          "page": 12,
          "quote": "Verbatim supporting quote from the source."
        }
      ]
    }
  ],
  "needs_review": [
    {
      "id": "q_017",
      "reason": "unresolved_cross_reference",
      "question_text_raw": "Original extracted statement with unresolved reference.",
      "evidence": [
        {
          "page": "HTML lines 57-58",
          "quote": "Verbatim source quote for the unresolved item."
        }
      ]
    }
```

```
  ],
  "trace": [
    {
      "id": "q_001",
      "stage": "rewrite",
      "notes": "Expanded Proposition 2.1 reference using evidence from page 10.",
      "evidence_refs": [
        "p10",
        "p12"
      ]
    }
  ]
}
```

Do not include markdown fences in final answer. Output raw JSON only.

‘page‘ may be either a numeric page number or a stable string locator, depending on ‘citation_mode‘.

Auto-Save (MANDATORY)

Before returning the final JSON to user:

1. Create output directory:
   - ‘RUN_DIR‘
2. Generate output file name from the normalized source name:
   - ‘<source_name_sanitized>.json‘ (local PDF stem, arXiv id, DOI slug, or URL slug)
   - Sanitization rule: replace every ‘/‘ with ‘_‘ and every remaining non-alphanumeric character except ‘-‘ and ‘.‘ with ‘_‘; do not use ‘-‘ as a slash replacement
3. Save the exact JSON output to:
   - ‘RUN_DIR/<source_name_sanitized>.json‘
4. Also write/update:
   - ‘LATEST_FILE‘
5. Append a ‘trace‘ entry noting save paths under stage ‘persist‘.

If save fails, still return JSON but add a ‘needs_review‘ item with reason ‘persist_failed‘ and include failure details in ‘trace‘.

Table 11: Prompt for the Extractor Agent

Prompt for the Refiner Agent

I will give you an arXiv paper ID and a draft question.

Your job is to read the paper itself and turn my draft into a highly complete, sharp, stand-alone research question that stays as close as possible to the paper’s actual intent.

Paper to inspect:
arXiv:{arxiv_id}

Draft question:
{question_text}

Instructions:

1. Search the paper and identify the exact statement, conjecture, or open problem that my draft is trying to capture.
   - Prefer the nearest official formulation in the paper.
   - Do not replace it with a broader neighboring problem.

2. Read the exact statement and the nearby context that defines notation, scope, and hidden assumptions.
   - Check the introduction first.
   - Then check any proposition/theorem/conjecture/remark/hypothesis that supplies equivalent conditions, notation, or a more precise reformulation.

3. Evaluate how complete my draft question is on a 0 to 10 scale.
   - Say what is already correct.
   - List every missing or implicit detail needed to make it fully stand-alone and paper-faithful.
   - Include domain-specific definitions (e.g. variables, operators, experimental conditions, model assumptions, scope restrictions, quantifiers) that a reader outside the paper would need to understand and attempt the question.

4. Rewrite the question as one refined research question that is:
   - self-contained (no unexplained symbols, jargon, or implicit scope),
   - tightly scoped,
   - as close as possible to the paper’s original intent,
   - not broader than the paper,
   - not vague.

5. Give at most one alternate formulation only if the paper itself explicitly proves the equivalence.
   - Do not invent new variants.
   - Do not generalize unless the paper itself does so in the exact nearby discussion.

6. Guardrails:
   - Do not broaden a specific narrow question into a more general problem unless the paper explicitly treats that

```
      generalization as its main target.
   - Do not drift into side problems, variants, or adjacent subfields unless my draft explicitly asks for them.
   - Do not add speculative assumptions or remove paper-critical restrictions (e.g. specific parameter ranges, experimental
      constraints, model assumptions).
   - Keep the result narrow.
   - The "refined_question" field must contain only plain text. Do not include any URLs, hyperlinks, DOIs, PDF references,
      arXiv links, or citations of any kind inside the refined question itself.

7. Support every nontrivial claim by citing the exact place in the paper:
   - section number,
   - conjecture/proposition/theorem/open problem number,
   - and page number if available.

Output requirements:

Return ONLY a valid JSON object.

The JSON must contain the following fields:

- "target_in_paper": string
- "completeness_score": number
- "correct_elements": string
- "missing_details": array of strings
- "refined_question": string
- "equivalent_formulation": string or null
- "justification": string
- "citations": array of objects with:
    - "section": string
    - "label": string
    - "page": string or number

Do NOT include any text outside the JSON.
Do NOT include markdown formatting.
Ensure the JSON is syntactically valid.
```

Table 12: Prompt for the Refiner Agent

Prompt for Difficulty Comparison

```
You are an expert mathematics judge comparing two math problems.

Your task is to decide which problem is more difficult in each dimension below.
A "win" means the problem is more difficult for that dimension; a "lose" means it is less difficult.
Use the labels "problem_1" and "problem_2" exactly. Use "draw" when neither problem is meaningfully more difficult in that
      dimension.
Do not solve the problems fully. Reason only enough to compare relative difficulty.

Return only a valid JSON object with exactly these top-level keys:
- procedural_difficulty
- memory_or_knowledge_difficulty
- novelty_difficulty

Each value must be an object with this exact shape:
{{
  "winner": "problem_1, problem_2, or draw",
  "problem_1": "win, lose, or draw",
  "problem_2": "win, lose, or draw",
  "reason": "one concise sentence"
}}

Example output format:
{{
  "procedural_difficulty": {{
    "winner": "problem_2",
    "problem_1": "lose",
    "problem_2": "win",
    "reason": "Problem 2 requires more algebraic manipulation and case bookkeeping."
  }},
  "memory_or_knowledge_difficulty": {{
    "winner": "draw",
    "problem_1": "draw",
    "problem_2": "draw",
    "reason": "Both problems require a comparable level of standard background knowledge."
  }},
  "novelty_difficulty": {{
    "winner": "problem_2",
    "problem_1": "lose",
    "problem_2": "win",
    "reason": "Problem 2 hides the relevant method behind a less familiar formulation."
  }}
}}

Problem 1:
```

```
{problem_1}

Problem 2:
{problem_2}

Dimensions:

## Procedural difficulty

How many steps or calculations are required.

A problem can be conceptually easy but procedurally long.

Example: expanding a large polynomial, doing long algebra, or computing a determinant by hand.

This includes:

| Aspect | Meaning |
|---|---|
| Number of steps | More steps means more chances to make errors |
| Algebraic manipulation | Simplifying, rearranging, expanding, factoring |
| Calculation burden | Arithmetic, fractions, signs, exponents |
| Bookkeeping | Keeping track of cases, variables, or expressions |

## Memory or knowledge difficulty

How much prior knowledge the solver needs.

Example: a problem may be easy if you know the AM-GM inequality, but very hard if you do not.

This includes:

| Aspect | Meaning |
|---|---|
| Formula recall | Remembering identities or standard formulas |
| Theorem recall | Knowing which results are available |
| Technique familiarity | Having seen similar patterns before |
| Domain-specific knowledge | Number theory, combinatorics, topology, etc. |

## Novelty difficulty

How different the problem is from familiar examples.

A routine integral may be easy. A strange-looking integral requiring an unusual substitution may be hard.

This includes:

| Aspect | Meaning |
|---|---|
| Familiarity | Whether the solver has seen similar problems |
| Surface disguise | Familiar idea hidden under unusual wording |
| Creative leap | Need to invent or adapt a method |
| Transfer | Applying a known method in a new context |
```

Table 13: Prompt for Difficulty Comparison

Prompt for Response Generation

```
Solve the following problem.

Problem:
{problem}
```

Table 14: Prompt for Response Generation

Prompt for Factuality Metric: Reference Span Extraction

```
You are an exact-span annotation tool. Return only a JSON array of strings.

Task: identify external references in one block from a mathematical reasoning trace.

Definition:
An external reference is any arXiv/paper/book mention, website/URL, named author/source used as support.

Examples of external references:
- "the paper by Smith et al."
- "according to Wikipedia"
- "from the Art of Problem Solving wiki"
- "in the book 'Affine Algebraic Geometry' by Gutierrez, Masuda, etc."
```

```
DO NOT extract:
- Theorems or lemmas with names, such as the Cauchy-Schwarz inequality, Descartes' theorem, or Burnside's lemma
- mathematical objects, variables, equations, or theorems used without an outside source
- references to prior reasoning attempts

Rules:
1. Copy the complete exact substring from the block for each external reference.
2. Every string must be a contiguous substring that appears verbatim in the block.
3. Do not paraphrase, normalize spelling, add positions, or add text that is not in the block.
4. Do not output overlapping strings. If one citation contains another marker, output only the larger complete citation/source
      phrase.
5. If there is no external reference, output [].
6. Output JSON only. No Markdown, no explanation.


Block:
<<<BLOCK
{block}
BLOCK
```

Table 15: Prompt for factuality reference-span extraction

Prompt for Factuality Metric: Agent Verification

```
You are verifying one detected external-reference span from a mathematical reasoning trace.

Important:
- You MUST use web search before deciding.
- Return exactly one lowercase boolean token: true or false.

Decision rule:
- Return true only if the target is a purported specific external reference or citation
  (for example a named paper, book, author/person used as a source, website/page, or
  similar source) and web search indicates that this cited reference is likely
  fabricated, nonexistent, or materially mismatched.
- Return false if the target is a real existing reference/source/author/website/page.
- Return false if the target is not actually an external reference, such as a theorem,
  concept, generic phrase, internal note, or ordinary problem text.
- If the evidence is ambiguous and the target is not clearly a fabricated specific
  citation, return false.

Remember: true means "keep this as a fake reference"; false means "filter it out".

Problem:
<<<PROBLEM
{problem_text}
PROBLEM

Target Reference:
<<<REFERENCE
{reference}
REFERENCE

Exact Detected Block:
<<<EXACT_BLOCK
{block_text}
EXACT_BLOCK
```

Table 16: Prompt for factuality agent verification

Short-Block Prompt for Factuality Metric: Agent Verification

```
You are verifying one detected external-reference span from a mathematical reasoning trace.

Important:
- You MUST use web search before deciding.
- Return exactly one lowercase boolean token: true or false.

Decision rule:
- Return true only if the target is a purported specific external reference or citation
  (for example a named paper, book, author/person used as a source, website/page, or
  similar source) and web search indicates that this cited reference is likely
  fabricated, nonexistent, or materially mismatched.
- Return false if the target is a real existing reference/source/author/website/page.
- Return false if the target is not actually an external reference, such as a theorem,
  concept, generic phrase, internal note, or ordinary problem text.
- If the evidence is ambiguous and the target is not clearly a fabricated specific
  citation, return false.
```

```
Remember: true means "keep this as a fake reference"; false means "filter it out".

Problem:
<<<PROBLEM
{problem_text}
PROBLEM

Target Reference:
<<<REFERENCE
{reference}
REFERENCE

Exact Detected Block:
<<<EXACT_BLOCK
{block_text}
EXACT_BLOCK

Surrounding Reasoning Window Containing The Target:
<<<BLOCK
{prompt_context}
BLOCK
```

Table 17: Short-block prompt for factuality agent verification