SCALING MATHEMATICAL PROOF SUPERVISION FROM STRUCTURED TEXTBOOKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Mathematical supervision for language models is typically flattened into isolated problem–solution pairs, discarding the curricular organization and evidential structure of the original sources. We introduce **TOME-Math** (**T**extbook-**O**riented **M**athematical **E**vidence), a large-scale corpus that converts 2,118 textbooks spanning the core areas of university-level mathematics subjects into approximately 380K structured proof-supervision records while preserving chapter order, proposition–proof links, dependency references, and source provenance. Leveraging these structural signals, we explore curriculum learning across multiple mathematical subjects and characterize scaling patterns with supervision volume and source breadth. At full scale, fine-tuning on TOME-Math raises the unweighted mean score across our held-out evaluations and established public benchmarks by **16.7** percentage points over the base model, while exceeding all three public instruction-data baselines evaluated in aggregate. Our findings highlight structured textbooks as both a scalable source of mathematical supervision and an empirical framework for studying the impact of data organization on mathematical reasoning.

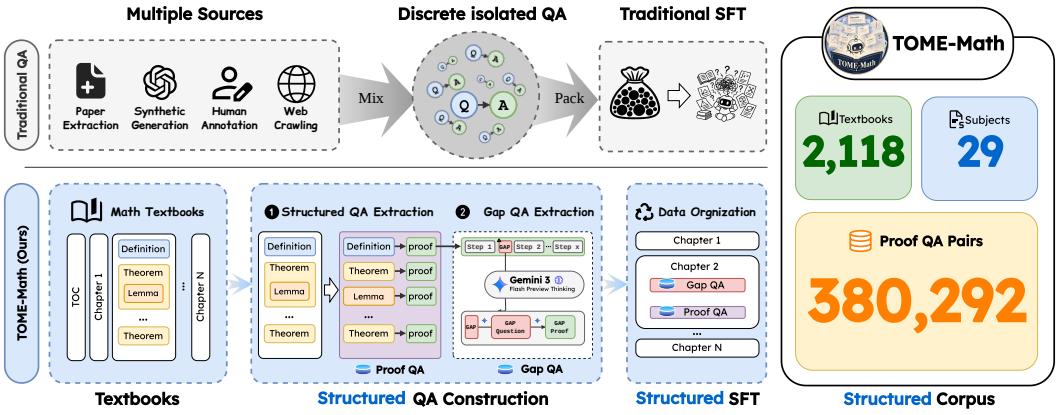


Figure 1: Overview of TOME-Math. **Top-left**: Traditional pipelines flatten heterogeneous sources into isolated QA pairs. **Bottom-left**: TOME-Math preserves textbook structure through structured QA construction and curriculum-based SFT. **Right**: TOME-Math comprises 380,292 QA pairs from 2,118 textbooks spanning 29 subjects.

1 INTRODUCTION

Mathematical reasoning in large language models has advanced rapidly, driven in part by improvements in training data and in part by advances in reasoning-time computation (Shao et al., 2024; Yang et al., 2024; OpenAI, 2024; Guo et al., 2025; Yang et al., 2025). On the data side, recent efforts have expanded mathematical supervision to millions of QA pairs (*i.e.*, Question-Answer pairs) and in doing so have revealed that properties such as question diversity, solution format, and teacher-model quality significantly shape downstream performance (Yu et al., 2023; Yue et al., 2024; Toshniwal

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

et al., 2024a). These findings suggest a broader principle: *The value of mathematical supervision is not determined by the number of training examples alone, but also by how those examples are constructed and organized*.

Despite this, the dominant paradigm for mathematical supervision remains the isolated problem–solution pair. Whether the original knowledge comes from competition archives, grade-school word problems, web exercises, or synthetic augmentation, the resulting training data typically takes the uniform shape of problem $\rightarrow$ solution (Cobbe et al., 2021; Hendrycks et al., 2021; Yu et al., 2023). This representation is convenient to scale, but it does not explicitly leverage the organizational structure that many source materials already possess.

Mathematical textbooks are a notable case in point. A textbook is not merely a collection of solved problems, but a carefully structured body of knowledge. Definitions introduce core concepts, lemmas and theorems develop them, proofs establish their validity, and exercises reinforce understanding. These elements are organized within a chapter hierarchy that reflects prerequisite relations, conceptual dependencies, and pedagogical progression. Prior work has explored textbooks as a source of mathematical pretraining data, problem–solution pairs, and proof supervision (Wang et al., 2024; Tang et al., 2024; Welleck et al., 2021; Azerbayev et al., 2023; Huang et al., 2025a; Lee et al., 2026). However, textbooks are often treated primarily as sources of mathematical content, leaving their native structure underexplored. In particular, it remains unclear how to scale natural-language proof supervision while preserving chapter order, proposition dependencies, source locations, and exercises in a unified framework.

To address this gap, we introduce **TOME-Math** (**T**extbook-**O**riented **M**athematical **E**vidence), a large-scale proof-supervision resource constructed from 2,118 mathematics textbooks. Central to TOME-Math is the notion of *structured* mathematical supervision, which preserves structure at three levels. First, **hierarchical structure** retains the native organization of textbooks, including chapters, propositions, and their ordering. Second, **dependency structure** connects theorem-level proof questions with the intermediate proof obligations derived from gaps in their proofs. Third, **curriculum structure** organizes supervision according to textbook order and proof granularity, enabling controlled training over related examples rather than independent QA pairs. Following these principles, TOME-Math contains 380,292 proof-oriented QA pairs spanning 29 mathematical subjects. Figure 1 provides an overview of the construction and organization of the resulting corpus.

This retained structure enables controlled studies of how the organization of supervision affects mathematical reasoning. Rather than treating scale as the only variable, TOME-Math allows us to examine the roles of retrieval structure, curriculum order, and source coverage under controlled settings. Empirically, fine-tuning on TOME-Math improves the six-benchmark average from 29.17% to 45.83%, while further experiments show consistent benefits from structure-guided retrieval, textbook-aligned curricula, and broader source coverage.

Our contributions are:

- We introduce TOME-Math, a large-scale proof-supervision resource constructed from 2,118 mathematics textbooks. Through a structure-preserving pipeline, it produces 380,292 Concept and Gap QA records while retaining chapter progression, proposition–proof links, dependency references, source provenance, and a separate inventory of textbook-authored exercises.

- We demonstrate the overall effectiveness of TOME-Math through full-corpus fine-tuning. On book-held-out textbook exercises and established public benchmarks, TOME-Math achieves the best performance on five of six evaluations, improving the unweighted six-benchmark mean from 29.17% to 45.83% over the base model and outperforming all three public instruction-data baselines in aggregate.

- We systematically investigate how textbook structure contributes to mathematical reasoning at inference and training time. Structure-guided retrieval improves proof-generation pass@1 from 13.25% to 17.38% under a matched context budget; across six subjects and three data scales, forward textbook curricula consistently outperform shuffled and reverse schedules; and broader source coverage yields further gains under a fixed training-token budget.

Table 1: Mathematical resources by scale, supervision, and structure. Sizes retain reported units; sources include synthetic seeds. NL = natural language; refs = proposition references.

Resource	Size	Source	Supervision	Metadata / structure
Mathematical instruction data				
MATH (2021)	12.5K QA	Competitions	Solutions	Subject, difficulty
MetaMathQA (2023)	395K QA	GSM8K, MATH	Solutions	Seeds, augmentation types
OpenMathInstruct-2 (2024a)	14M QA	GSM8K, MATH	Solutions	Source labels
NuminaMath-CoT (2024)	∼860K QA	Exams, web, synthetic	Solutions	Source labels
Textbook and knowledge-informed resources				
MathPile (2024)	9.5B tokens	Books, papers, web	Text	Sources, file paths
CollegeMath (2024)	4,099 QA	9 textbooks	Short answers	Book/section/exercise IDs
Proof resources and evaluation				
NaturalProofs (2021)	32K proofs	2 textbooks + web	NL proofs	Reference graph
ProofNet (2023)	371 examples	Textbooks, Putnam	NL + Lean stmt.	Book/exercise IDs
Lean-GAP (2026)	430 problems	1 textbook	Lean statements	Section/exercise IDs
AlgGeoTest (2025)	456 questions	Stacks Project	Proof judgments	Statement tags
TOME-Math (ours)	**380,292 QA**	**2,118 textbooks**	**NL + local proofs**	**Chapter order, refs, provenance**

2 Related Work

Mathematical instruction data. GSM8K and MATH provide human-authored problems and solutions for grade-school and competition mathematics (Cobbe et al., 2021; Hendrycks et al., 2021). MetaMathQA and OpenMathInstruct-2 scale these seeds through synthetic augmentation (Yu et al., 2023; Toshniwal et al., 2024a), while NuminaMath-CoT aggregates diverse sources with chain-of-thought solutions (Li et al., 2024). Their primary supervision unit is a problem–solution pair, accompanied by source, seed-question, or difficulty metadata. Table 1 contrasts this organization with the textbook hierarchy and proposition links retained by TOME-Math.

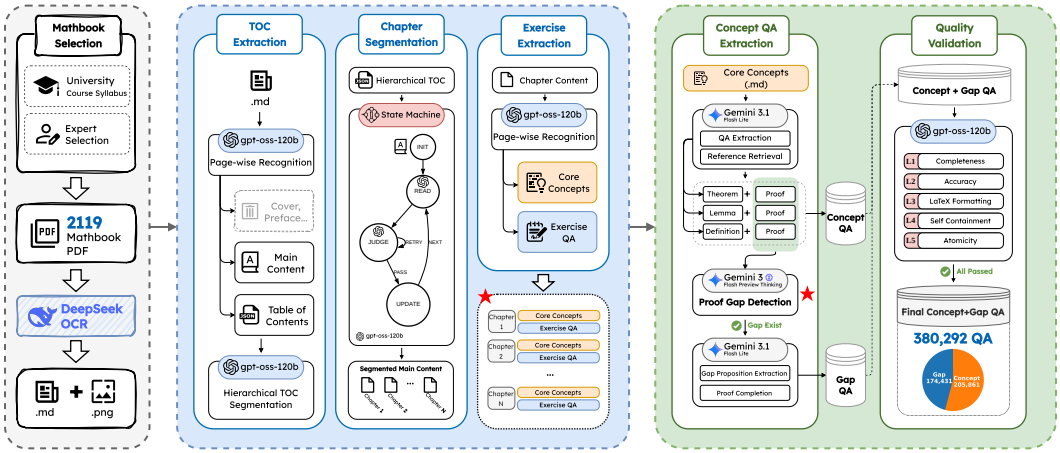
Textbook and knowledge-informed resources. OpenWebMath and MathPile provide mathematical pretraining text, with MathPile incorporating textbooks alongside papers and web sources (Paster et al., 2023; Wang et al., 2024). CollegeMath extracts textbook exercises and short answers, while MathScale synthesizes QA using a graph of topics and knowledge points (Tang et al., 2024). TheoremQA evaluates theorem application across STEM subjects (Chen et al., 2023). These efforts motivate retaining both textbook organization and links to mathematical statements when constructing proof supervision.

Proof resources and evaluation. NaturalProofs preserves statements, proofs, and a reference graph; NaturalProofs-Gen and NaturalProver study reference-grounded proof generation (Welleck et al., 2021; 2022). ProofNet and Lean-GAP connect textbook mathematics to formalization (Azerbayev et al., 2023; Lee et al., 2026), while FormaRL's uproof provides 5,273 proof problems from 14 textbooks for autoformalization evaluation (Huang et al., 2025a). Proof2Hybrid's AlgGeoTest studies proof-derived judgments, and the Open Proof Corpus studies automated assessment of human-evaluated model proofs (Peng et al., 2025; Dekoninck et al., 2025). Building on these directions, TOME-Math combines coverage of 2,118 textbooks with self-contained local proof obligations, chapter order, and proposition references, supporting controlled studies of source coverage, proof-target format, and training order.

3 TOME-Math

3.1 Data Construction

Pipeline overview. We construct TOME-Math through a structure-preserving pipeline that converts raw textbook PDFs into two complementary forms of proof supervision: **Concept QA** derived from explicit proposition–proof units and **Gap QA** derived from missing proof obligations in Concept QAs. The source collection is selected against university course syllabi and reviewed by domain experts, yielding 2,118 input PDFs after canonicalization and deduplication (as shown in Section 3.2).



(a) Data Collection (b) Structured Chapter Extraction (c) Structured Proof QA Extraction

Figure 2: Overview of TOME-Math construction. Details as shown in Section 3.1.

The pipeline generated the full TOME-Math dataset of 380,292 QA pairs, comprising 174,431 Gap QA pairs and 205,861 Concept QA pairs.

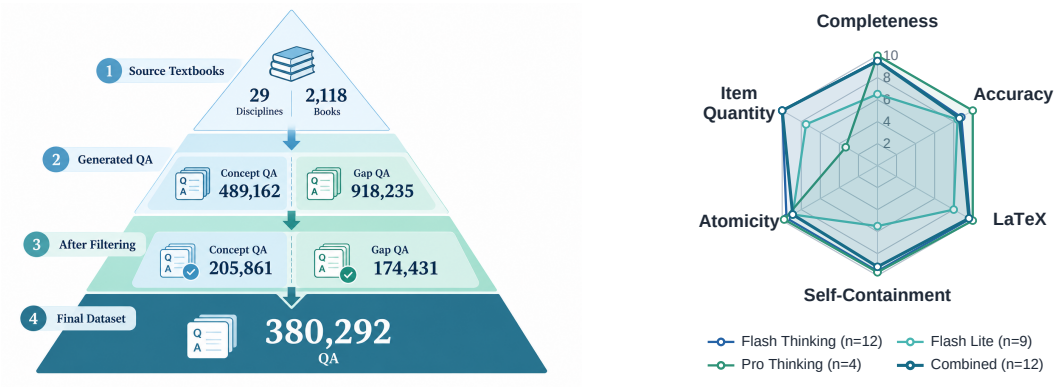
Document parsing and structured chapter extraction. DeepSeek-OCR converts each PDF into multimodal Markdown together with page images. A page-wise `gpt-oss-120b` pass separates the table of contents and main body from covers, prefaces, and other front matter, then reconstructs a hierarchical table of contents. Conditioned on this hierarchy, a state machine reads the body in order, judges whether each page belongs to the current chapter, the next chapter, or contains an internal boundary, and retries ambiguous assignments before updating the chapter state. This produces an ordered set of chapter-level documents while preserving the textbook hierarchy. A second page-wise pass separates expository core concepts from textbook exercises within each chapter. The exercises are retained as an independent reservoir, but are excluded from the Concept QA generation path.

Concept QA Extraction and Gap QA construction. For each core concept chapter, we use `gemini-3.1-flash-lite-preview` to extract theorem–proof, lemma–proof, and definition–proof units and retrieves the earlier propositions needed to interpret each proof. These source-grounded units form the Concept QA stream. We then use `gemini-3-flash-preview-thinking` to inspect each proof for **gaps** (*i.e.*, implicit or missing reasoning steps). When a substantive gap is detected, `gemini-3.1-flash-lite-preview` rewrites the missing obligation as a self-contained proposition and completes its proof, producing the Gap QA stream. This structured construction preserves explicit textbook proofs while adding targeted supervision for intermediate reasoning that the source leaves unstated.

Validation and quality control. Quality control is applied at both the structural and semantic levels. The structural checks validate the table-of-contents hierarchy, ambiguous page boundaries, question continuity across pages, question–answer alignment, and standalone completeness. For proposition extraction, a source-conditioned judge rubric audits candidates along five axes: completeness, accuracy, LaTeX formatting, self-containment, and atomicity. Accuracy is treated as critical: invented proof steps, altered mathematical relations, or other content not supported by the source are failure cases rather than stylistic defects. Full module logic, intermediate artifacts, model-allocation details, and the judging rubric are given in Appendix A.

3.2 Characterization

TOME-Math comprises 380,292 QA records from 2,118 canonical textbooks across 29 subject categories, summarized in Figure 3. Appendix B provides the detailed corpus characterization, including subject and language coverage, task composition, structural complexity, confidence dynamics, quality, and provenance.



(a) Dataset construction and filtering.

(b) Model-quality comparison.

Figure 3: TOME-Math data composition and quality characterization. Left: dataset scale from source textbooks through filtering to the final corpus. Right: model configurations compared across the quality-control criteria.

Generation Composition The corpus combines a Concept QA stream of 205,861 records and a Gap QA stream of 174,431 records, drawing on disjoint model assignments (Appendix B). `gemini-3-flash-preview-thinking` produces 88.9% of Concept QA records, `gpt-5-mini` produces 53.7% of Gap QA records, and the Gap QA stream retains 174,431 of 918,235 generated candidates, a yield of 19.00%.

Provenance and Evaluation Units Each record carries book, chapter, and source location metadata, with proposition identifiers and resolved references for Concept QA records. We define two evaluation scopes at the book level, held out exercises for item level generalization and held out books for source level generalization, both respecting duplicate groups identified by global content hash.

4 Evaluating TOME-Math

4.1 Experimental Setup

Base Model. We use Qwen3-30B-A3B-Base (Yang et al., 2025) as the backbone for all experiments. All SFT variants start from the same checkpoint, and the unmodified checkpoint serves as the base-model baseline.

Evaluation. To obtain a sensitive view of how structured textbook supervision affects learning, we focus on the six core subjects with the largest usable textbook collections in our corpus: abstract algebra, differential geometry, discrete mathematics, mathematical analysis, probability, and topology. We construct **TOME-LongForm** from their textbook-authored exercises: 300 proof-heavy problems from held-out books, balanced across subjects and the earlier and later halves of each book. To enforce book-level separation, we reserve 64 source books for evaluation and exclude all 15,661 QA records associated with those books from the complete 380,292-record TOME-Math corpus. This leaves 364,631 examples in the actual training split. The evaluation problems are selected separately from the native-exercise reservoir of the held-out books. We sample 16 solutions per problem and score them with a reference-guided step verifier; each solution is reviewed eight times by `gpt-5.6-luna` and is accepted only if no review identifies a substantive error. For complementary coverage beyond long-form proofs, **TOME-Objective** contains 145 multidisciplinary objective problems scored by final-answer equivalence. We additionally evaluate on **MiniF2F** (Zheng et al., 2021), **ProofNet** (Azerbayev et al., 2023), **MATH-500** (Hendrycks et al., 2021), and **AIME2026** (Dekoninck et al., 2026) using their standard evaluation procedures. Full construction and scoring details are given in Appendix D.

Table 2: Full-corpus SFT comparison on TOME-LongForm (TOME-LF), TOME-Objective, MiniF2F, ProofNet, MATH-500, and AIME 2026. Scores are reported as percentages; Avg. is the unweighted mean of the six reported scores. Best and second-best results in each column are shown in **bold** and underlined, respectively.

Model (training data)	TOME-LF	TOME-Obj.	MiniF2F	ProofNet	MATH-500	AIME-26	Avg.
Qwen3-30B-A3B-Base	13.25	32.93	34.45	17.14	68.80	8.44	29.17
+ OpenMath1 (1.82M)	20.16	33.86	22.87	19.25	70.40	8.21	29.13
+ MetaMath (395k)	28.73	34.66	40.70	22.55	71.20	9.98	34.64
+ MathInstruct (262k)	35.87	40.69	52.05	32.93	73.05	**11.98**	41.10
+ TOME-Math (ours; 364.6k)	**43.86**	**43.96**	**57.43**	**42.98**	**76.55**	10.19	**45.83**

4.2 Full-Corpus Comparison

We first present the overall effectiveness of the final TOME-Math dataset, before isolating its structural components in Section 5. Table 2 compares the resulting 364,631-example TOME-Math training split with three public mathematics SFT corpora at their available scales. All fine-tuned models start from Qwen3-30B-A3B-Base and use the same fine-tuning setup, differing only in their supervision corpus; each model is trained for one full epoch over its respective corpus, and the unmodified checkpoint is included as a reference. Complete training configurations are provided in Appendix E.2. This is an end-to-end comparison of the complete data resources rather than a matched-size or compute-matched attribution of any single data property.

Aggregate performance. TOME-Math achieves the best result on five of the six benchmarks and ranks second on AIME 2026. Its largest margins over the strongest public baseline appear on the proof-oriented TOME-LongForm and ProofNet evaluations, while its advantages on TOME-Objective, MiniF2F, and MATH-500 show that the benefit extends beyond its own evaluation distribution and is not confined to long-form proof generation. Averaged uniformly across the six reported scores, TOME-Math improves the base-model mean from 29.17% to 45.83%, a gain of 16.66 percentage points (16.7 when rounded to one decimal).

Effect of supervision corpus. The ranking is not explained by corpus size alone. OpenMath1 contains 1.82M examples—approximately $4.8\times$ the size of TOME-Math—yet TOME-Math outperforms it on all six benchmarks. It also exceeds the similarly sized MetaMath corpus throughout and improves consistently over the pretrained backbone. These results demonstrate the overall utility of TOME-Math and suggest that the source and form of supervision matter in addition to raw example count. Because the corpora differ in several respects, the comparison does not by itself isolate textbook structure as the causal factor; we examine that structure through controlled retrieval and training studies in Section 5.

5 Leveraging Textbook Structure

Beyond providing a large source of proof supervision, TOME-Math exposes structural signals—chapter progression, proposition dependencies, and source provenance—that can be used explicitly at inference time and during training. This section investigates when these signals improve mathematical reasoning. We first evaluate structure-guided retrieval without updating model parameters (Section 5.1). We then study whether textbook order provides an effective curriculum under controlled budgets (Section 5.2), and examine source breadth together with transfer to held-out books from other mathematical fields (Section 5.3). Finally, we examine how these structural choices contribute to the full-corpus TOME-Math training recipe (Section 5.4).

5.1 Training-Free Structure-Guided Retrieval

Question and design. Can the structure preserved by TOME-Math improve reasoning at inference time, before any parameter updates? We evaluate the pretrained Qwen3-30B-A3B-Base on all 300 problems in TOME-LongForm under five paired conditions. Closed provides no external context. Random same-book samples entries uniformly from the exercise's source book, testing whether ac-

Table 3: Training-free context augmentation on TOME-LongForm (pass@1, %). Each condition is evaluated on the same 300 problems with the same pretrained solver. Avg. is the unweighted mean across the six equally sized subjects; the best result in each column is bolded.

	Overall	Subject breakdown					
Context	Avg.	Abs. Alg.	Diff. Geom.	Discrete	Math. Anal.	Prob.	Topology
Closed	13.25	8.50	5.25	13.25	25.75	20.00	6.75
Random same-book	14.00	9.75	6.50	12.50	24.75	20.50	10.00
Local chapter	14.71	10.50	6.75	14.50	25.50	20.00	11.00
TOME (props only)	15.54	11.00	8.50	15.00	27.25	20.50	11.00
TOME (props + gap)	**17.38**	**12.75**	**10.00**	**15.25**	**31.75**	**21.75**	**12.75**

cess to any in-book material is sufficient. Local chapter restricts the pool to the exercise chapter and its immediate predecessors, adding curricular locality. TOME (props only) instead retrieves from the extracted proposition–proof inventory using chapter position and recovered dependencies. Finally, TOME (props + gap) applies the same structure-guided retrieval to the union of proposition records and Gap QA, testing whether explicit local proof obligations complement the textbook propositions.

All context-augmented conditions receive the same context-token budget. They also share the same backbone, prompt, decoding configuration, and reference-guided verifier; the target exercise and material overlapping its reference answer are removed from every retrieval pool. The resulting comparison asks how effectively each policy selects a fixed amount of same-source evidence, rather than whether additional test-time tokens alone help. Appendix F gives the complete retrieval, leakage-control, generation, and scoring protocol.

Results. Structure-guided retrieval raises average pass@1 from 13.25% to 17.38% (+4.13 points). Random same-book and local-chapter context reach only 14.00% and 14.71%, respectively, while proposition retrieval reaches 15.54% and adding Gap QA produces the largest incremental gain (+1.84 points). Since all conditions use the same context budget, this progression indicates that the gain comes from selecting structurally relevant evidence and supplementing it with explicit proof obligations, rather than from adding more text.

The full TOME (props + gap) policy is best in every subject, improving over closed-book generation by 1.75–6.00 points; mathematical analysis and topology show the largest gains (+6.00 each). In contrast, intermediate baselines occasionally hurt or leave performance unchanged, underscoring the cross-subject consistency of the complete retrieval policy.

Interpretation. These results establish that TOME-Math's retained structure is useful as a test-time evidence index and that Gap QA complements the proposition inventory. They do not, however, identify a causal effect of dependency edges alone: the conditions change both the admissible candidate pool and the retrieval constraint. We therefore treat the table as an evaluation of the complete retrieval policies, not as a fully factorial ablation of each structural signal.

5.2 Curriculum Learning Across Subjects and Data Scales

Setup and comparisons. We test whether textbook progression provides a useful SFT curriculum across six subjects and three per-subject target-token budgets: 0.75M, 1.8M, and 2.4M. Within each subject–scale cell, four schedules reorder the same frozen QA set: Shuffle, Forward–Gap, Reverse–Gap, and Forward–Mix. Here, Gap QA makes a locally omitted proof obligation explicit. Forward–Gap versus Reverse–Gap isolates chapter direction under a fixed Gap-first policy, while Forward–Gap versus Forward–Mix isolates Gap-first ordering under a fixed forward progression; comparison with Shuffle measures their combined effect. All schedules share the same examples, token budget, initialization, and optimization settings, and are evaluated on the same held-out-book exercises. Appendix G details the normalized chapter stages, schedule construction, nested data scales, and order-preserving training controls.

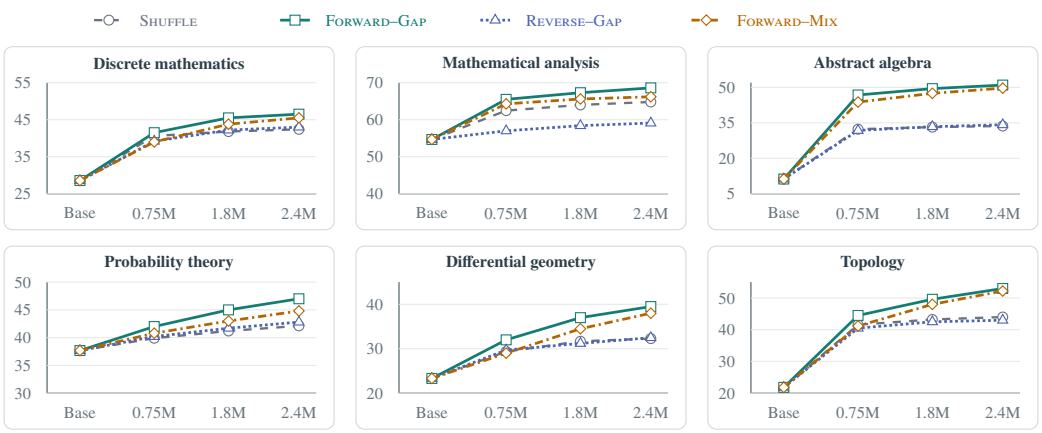


Figure 4: Curriculum performance across subjects and target-token budgets. Pass@16 is the fraction of problems with at least one verifier-accepted solution among 16 samples. Within each subject–scale cell, all schedules use identical examples and budgets; Base denotes the shared pretrained checkpoint. Because panels use subject-specific vertical ranges, slopes should be compared only within panels.

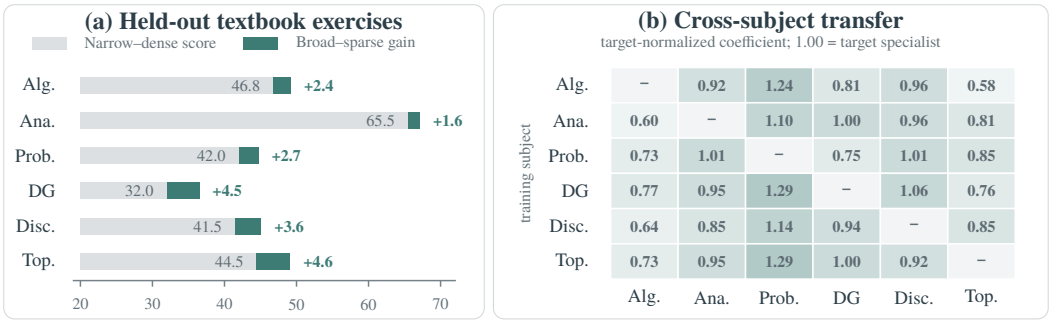


Figure 5: Source breadth and cross-subject performance at the 0.75M-token scale. Panel (a) compares matched-budget source allocations on held-out textbook exercises: gray bars show the 8-book narrow–dense baseline, while teal extensions show the gain from distributing the same token budget across 16 books (broad–sparse). Panel (b) reports cross-subject performance for subject-specific checkpoints, with rows denoting training subjects and columns denoting held-out evaluation subjects. Each cell shows the target-normalized pass@16 coefficient $T_{s\to t}$; 1.0 corresponds to the target-specialized checkpoint, and values above 1 indicate stronger cross-subject performance.

Overall effect of scale and curriculum. Figure 4 shows that every fine-tuned condition outperforms the shared base checkpoint in all six subjects, even at 0.75M tokens, and that every schedule improves with scale in each subject. Across the 0.75M, 1.8M, and 2.4M budgets, macro-average pass@16 rises from 41.04% to 42.52% and 43.21% for Shuffle, and from 45.38% to 48.98% and 50.93% for Forward–Gap. Thus, both supervision scale and structured ordering improve solution coverage.

Effect of chapter direction and Gap-first ordering. Forward–Gap is strongest in all 18 subject–scale cells, and its macro-average margin over Shuffle widens from 4.34 points at 0.75M tokens to 6.46 at 1.8M and 7.72 at 2.4M. Controlled comparisons attribute most of this benefit to chapter direction: relative to Reverse–Gap, the forward schedule gains 5.65, 7.41, and 8.50 points across the three scales. Holding forward progression fixed, Gap-first ordering adds a smaller but consistent 2.37, 1.91, and 1.53 points over Forward–Mix. Thus, forward progression is the primary contributor, with Gap-first ordering providing a complementary gain. Because each cell comes from a single training run, small margins should be interpreted cautiously; the uniform ordering across subjects and scales is stronger evidence.

Table 4: Full-corpus ablations on Qwen3-30B-A3B-Base (pass@1, %). Ordering variants use the same 380K proof-plus-gap corpus; "w/o Gap QA" uses 206K proof-only examples.

Variant	TOME-LF	TOME-Obj.	MiniF2F	ProofNet	MATH-500	AIME-26
w/o Gap QA (Forward; 206K)	38.72	38.91	50.73	36.21	73.84	9.12
w/o curriculum (Shuffle)	39.87	40.42	53.76	39.12	75.22	9.58
w/o gap-first (Forward–Mix)	40.12	41.37	55.61	41.02	75.88	9.93
TOME-Math (Forward–Gap)	43.86	43.96	57.43	42.98	76.55	10.19

5.3 Source Breadth and Cross-Subject Transfer

Having identified Forward–Gap as an effective curriculum, we ask whether a fixed supervision budget benefits from broader source coverage and whether subject-specific training transfers across mathematical fields. Figure 5 addresses both questions.

Source breadth. Figure 5(a) isolates source coverage at a fixed training volume. Narrow–dense is the 8-book, 0.75M-token Forward–Gap condition from Figure 4; Broad–sparse downsamples the nested 16-book pool to the same realized token count while retaining all 16 sources. All other training and evaluation settings are fixed, so this comparison isolates source breadth rather than scale. Broad–sparse improves pass@16 in all six subjects by 1.6–4.6 points, with a mean gain of 3.2 points. The largest gains occur in topology (+4.6), differential geometry (+4.5), and discrete mathematics (+3.6). Table 12 reports the nested source pools and QA counts.

Cross-subject performance. Figure 5(b) evaluates each subject-specific 0.75M-token Forward–Gap checkpoint on 50 held-out-book exercises from each of the other five fields. We normalize pass@16 by the target subject's in-domain score, $T_{s\to t} = P_{s\to t}/P_{t\to t}$, and omit the within-subject diagonal. The macro-average coefficient is 0.92, meaning that cross-subject checkpoints retain 92% of target-specialized performance on average. Topology (0.98) and differential geometry (0.97) have the strongest source-side means, and eight of the 30 ordered pairs exceed 1.0. The largest coefficients occur for probability targets, with both differential geometry and topology reaching 1.29. Because each cell contains only 50 exercises, coefficients near 1.0 should not be treated as meaningful rank differences. Appendix H provides full protocols and pass@k results.

5.4 Full-Corpus Ablation of the TOME-Math Training Recipe

The preceding studies show benefits from within-subject ordering and broad source coverage. We now test whether ordering remains useful in the full multidisciplinary setting by removing curriculum order, stage-wise Gap-first scheduling, or Gap QA from the final Forward–Gap recipe. The ordering ablations retain the same 380K-example corpus, whereas removing Gap QA leaves 206K proof-only examples and is therefore not size matched.

Ablation results. Each ablation lowers pass@1 on all six benchmarks. On the two TOME benchmarks, replacing gap-first ordering with within-stage mixing (Forward–Mix) costs 3.74 and 2.59 points; additionally replacing forward progression with global shuffling costs 0.25 and 0.95 points. Thus, Gap-first scheduling contributes more than chapter progression in this setting, reversing the subject-specific pattern in Section 5.2. Removing Gap QA produces the largest drops on the four proof benchmarks (5.05–6.77 points), but the smaller corpus prevents attributing them to supervision type alone.

6 Conclusion

We introduced TOME-Math, a corpus of 380K proof records from 2,118 textbooks that preserves curricular and dependency structure. It improves performance across textbook-derived and public benchmarks, while controlled studies demonstrate the benefits of structure-guided retrieval, curriculum ordering, and broader source coverage. We hope TOME-Math inspires mathematical reasoning systems that learn not only from more data, but from better-organized knowledge.

AI Use Statement

In this work, we used generative AI tools to generate and filter portions of the synthetic training data; assist with textbook parsing, structural extraction, reference resolution, source-conditioned proof auditing, and dataset cleaning and reformatting; and perform model-based evaluation of generated answers. The models and quality-control procedures used for data construction and evaluation are documented in Sections 3.1 and 4 and in Appendices A and D. AI-generated training examples were subjected to the filtering and auditing procedures described there; model-based judgments are identified as such and are not presented as formal verification. We did not use generative AI to formulate or prove mathematical claims made by the paper or to conduct qualitative or thematic analysis, and the remaining mandatory disclosure categories are not applicable to this work. All AI-assisted content was reviewed by the authors, who take full responsibility for the final text, claims, code, data, and other artifacts reported in this work.

References

Zhangir Azerbayev, Bartosz Piotrowski, Hailey Schoelkopf, Edward W. Ayers, Dragomir Radev, and Jeremy Avigad. Proofnet: Autoformalizing and formally proving undergraduate-level mathematics. *arXiv preprint arXiv:2302.12433*, 2023.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. TheoremQA: A theorem-driven question answering dataset. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://arxiv.org/abs/2305.12524.

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

Jasper Dekoninck, Ivo Petrov, Kristian Minchev, Mislav Balunovic, Martin Vechev, Miroslav Marinov, Maria Drencheva, Lyuba Konova, Milen Shumanov, Kaloyan Tsvetkov, Nikolay Drenchev, Lazar Todorov, Kalina Nikolova, Nikolay Georgiev, Vanesa Kalinkova, and Margulan Ismoldayev. The open proof corpus: A large-scale study of LLM-generated mathematical proofs. *arXiv preprint arXiv:2506.21621*, 2025. URL https://arxiv.org/abs/2506.21621.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: MathArena as an evaluation platform for mathematics with LLMs. *arXiv preprint arXiv:2605.00674*, 2026. URL https://arxiv.org/abs/2605.00674.

Daya Guo, Dejian Yang, He Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

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

Yanxing Huang, Xinling Jin, Sijie Liang, Peng Li, and Yang Liu. FormaRL: Enhancing autoformalization with no labeled data. In *Conference on Language Modeling*, 2025a. URL https://arxiv.org/abs/2508.18914.

Yanxing Huang, Zihan Tang, Zejin Lin, Peng Li, and Yang Liu. Pessimistic verification for open ended math questions. *arXiv preprint arXiv:2511.21522*, 2025b. URL https://arxiv.org/abs/2511.21522.

Seewoo Lee, Byung-Hak Hwang, Hyojae Lim, Jihoon Hyun, Ilkyoo Choi, Yeachan Park, Jineon Baek, Hyukpyo Hong, Keewoo Lee, Jaeseong Heo, Hyungryul Baik, Chul-hee Lee, and Kyu-Hwan Lee. Lean-gap: A dataset of formalized graduate algebra problems. *arXiv preprint arXiv:2606.02588*, 2026.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. NuminaMath. Hugging Face dataset, 2024. URL https://huggingface.co/datasets/AI-MO/NuminaMath-CoT.

OpenAI. Learning to reason with LLMs. *OpenAI Blog*, 2024. https://openai.com/index/learning-to-reason-with-llms/.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. OpenWebMath: An open dataset of high-quality mathematical web text. *arXiv preprint arXiv:2310.06786*, 2023. URL https://arxiv.org/abs/2310.06786.

Yebo Peng, Zixiang Liu, Yaoming Li, Zhizhuo Yang, Xinye Xu, Bowen Ye, Weijun Yuan, Zihan Wang, and Tong Yang. Proof2Hybrid: Automatic mathematical benchmark synthesis for proof-centric problems. *arXiv preprint arXiv:2508.02208*, 2025. URL https://arxiv.org/abs/2508.02208.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Zhengyang Tang, Xingxing Zhang, Benyou Wang, and Furu Wei. MathScale: Scaling instruction tuning for mathematical reasoning. In *Proceedings of the International Conference on Machine Learning*, 2024. URL https://arxiv.org/abs/2403.02884.

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

Shubham Toshniwal, Ivan Moshkov, Sean Narenthiran, Daria Gitman, Fei Jia, and Igor Gitman. Openmathinstruct-1: A 1.8 million math instruction tuning dataset. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2024b. URL https://arxiv.org/abs/2402.10176.

Zengzhi Wang, Xuefeng Li, Rui Xia, and Pengfei Liu. MathPile: A billion-token-scale pretraining corpus for math. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://arxiv.org/abs/2312.17120.

Sean Welleck, Jiacheng Liu, Ronan Le Bras, Hannaneh Hajishirzi, Yejin Choi, and Kyunghyun Cho. Naturalproofs: Mathematical theorem proving in natural language. *arXiv preprint arXiv:2104.01112*, 2021.

Sean Welleck, Jiacheng Liu, Ximing Lu, Hannaneh Hajishirzi, and Yejin Choi. NaturalProver: Grounded mathematical proof generation with language models. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2205.12910.

An Yang, Beichen Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, et al. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

An Yang, Beichen Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengpeng Huang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*, 2024.

Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. Minif2f: A cross-system benchmark for formal olympiad-level mathematics. *arXiv preprint arXiv:2109.00110*, 2021.

A TOME-MATH CONSTRUCTION DETAILS

This appendix expands the construction stages summarized in Section 3.1. We distinguish the artifacts produced by the book-structure modules (M1–M3), exercise and answer processing (M4–M6), and the downstream proposition-extraction track.

A.1 INTERMEDIATE REPRESENTATION

DeepSeek-OCR produces one multimodal-Markdown representation of the complete book together with the extracted images. The automated modules materialize the following intermediate artifacts:

Artifact	Contents
`toc.json`	Multilevel table of contents represented by title and heading depth.
`main_text.md`	Textbook body after removing front matter.
`images/`	Merged image directory with normalized references from the Markdown files.
`content/`	One Markdown file for every recovered chapter.
`content_pure/`	Chapter text after exercise segments are removed.
`exercise/`	Chapter-indexed exercise text and formatted per-chapter JSONL files.
`exercise_index.jsonl`	Multilevel heading path for every exercise section.
`answer.md`	A detached answer chapter when one is present.

A.2 M1–M2: TABLE-OF-CONTENTS AND CHAPTER RECOVERY

M1: front matter and table of contents. Starting from the first parsed page, the structure model decides whether a page contains table-of-contents entries and whether the textbook body has begun. For contents pages it extracts pairs of the form {title, level}. Pages that are neither contents nor body are treated as cover or front matter and are discarded. Once the body is detected, a verification pass corrects invalid heading depths and removes entries that belong to prefaces or notation lists. If the model detects body text before extracting any valid contents entries, the whole module is retried, up to five attempts. M1 emits `toc.json`, `main_text.md`, and the merged `images/` directory.

M2: hierarchy-conditioned segmentation. The recovered contents are traversed as a tree. Internal entries direct the traversal to their children, while leaf entries trigger a search for the corresponding body span. For each page and the current/next chapter titles, the model returns one of three states: ALL-CURRENT (the page belongs to the current chapter), ALL-NEXT (it belongs to the next chapter), or BOUNDARY (the transition occurs within the page). Boundary pages are checked a second time; a failed check is fed back to the assignment step. A successful transition advances the state machine and writes the completed chapter to `content/`.

A.3 M3–M6: EXERCISE EXTRACTION AND ANSWER ALIGNMENT

M3: exercise separation. The structure model first tests whether the table of contents contains dedicated exercise chapters. When it does, those chapters are detached directly. When it does not, each content chapter is scanned for a trailing block of exercises, which is removed from the exposition. The same module constructs the multilevel exercise-title index and detects a separate answer chapter, if present. Its outputs are `content_pure/`, `exercise/`, `exercise_index.jsonl`, and optionally `answer.md`.

M4: exercise serialization. The parser reads each exercise section page by page and emits one JSON record per problem. A record contains `question`, `answer` when an answer is printed immediately after the question, and `number`. The final problem fragment on a page is carried into the next page before serialization so that a page break does not split a question into two records. Problem numbers are preserved for the matching stage.

M5: answer partitioning. The pipeline determines whether numbering is *absolute*—one number identifies a unique problem in the whole book—or *relative*, with numbers restarting across chapters.

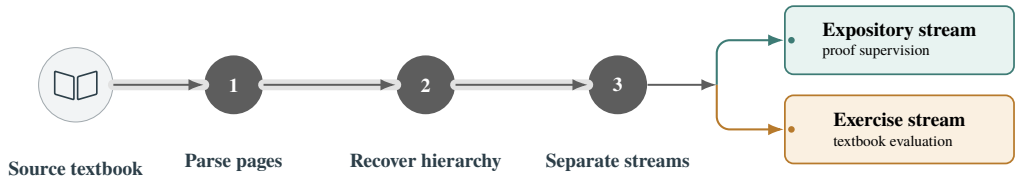


Figure 6: High-level construction flow. Structural recovery separates expository material from native exercises before the two streams are used for proof supervision and textbook evaluation.

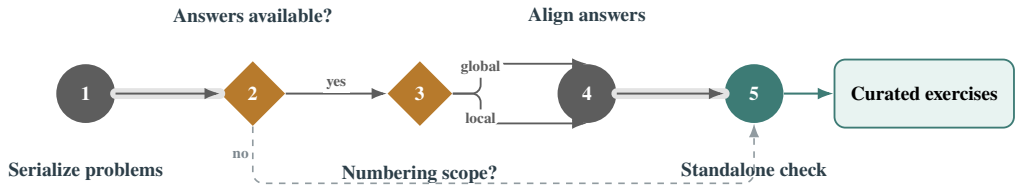


Figure 7: Exercise evaluation track. Problems are serialized, printed answers are aligned under the book's numbering scheme when available, and every candidate passes a standalone-completeness check.

For absolute numbering, the answer chapter is serialized using the same page-wise procedure as M4. For relative numbering, the multilevel heading paths saved by M3 are used to partition the answer chapter before serialization.

M6: matching and filtering. Absolute-number records are joined directly on problem number. Relative-number records are matched within their recovered chapter, using the section content when necessary. Books without printed answers follow a third, unmatched branch. All branches then pass through a completeness check that removes questions whose statements depend on an unprovided theorem or on the conclusion of another exercise. The module produces a report with the number of exercises and answer coverage for each chapter.

A.4 Four-stage proposition pipeline

Each pure-content chapter is first split at Markdown headings while preserving its preamble, and the resulting sections are processed concurrently. The four semantic stages are as follows.

1. **Proposition extraction.** For every section, the model emits structured proposition records with `type`, `number`, `label`, `title`, `content_tex`, and `proof_tex` fields.
2. **Reference resolution.** A book-level proposition library is built from the extracted records. For each proposition with a proof, the model identifies both explicit `\ref` citations and semantic references to earlier results, storing the matches in `resolved_references`.
3. **Proof-gap audit.** Each proof is checked for unverified lemma assumptions, omitted coordinate or operator transformations, and unsupported jumps marked by phrases such as "obviously" or "clearly." A candidate gap stores `gap_statement`, `missing_logic`, `sub_proposition`, and a low/medium/high `difficulty` label.
4. **Self-contained refinement.** Each candidate sub-proposition is rewritten as an independent mathematical statement by declaring symbols, injecting required definitions, and preserving the source language.

The chapter-level structured output contains the extracted propositions, resolved references, and candidate gaps. A separate Concept QA JSONL stream distinguishes original proposition records from candidate sub-propositions with an `is_potential_gap` flag. This schema describes the pipeline output; dataset-level acceptance, deduplication, and split policies are subsequent steps.

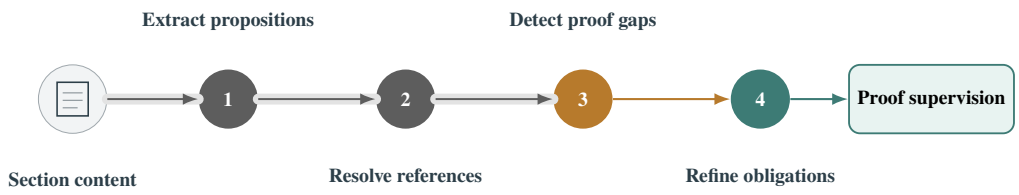


Figure 8: Proposition-centered proof track. Propositions are linked to their dependencies, local proof gaps are isolated, and selected obligations are refined into proof-supervision questions.

A.5 Proposition extraction and model allocation

The pure content chapters feed the proposition-centered training-data pipeline. The construction log compares three individual Gemini configurations and a hybrid allocation. The later cost-optimization note assigns second-stage reference semantics to the lower-cost non-thinking model and pairs it with the thinking model for the remaining reasoning-intensive operations. Because this allocation changed during pipeline development, it is versioned with each run. Table 5 reports the prices and per-book costs recorded in the design document; they are platform- and time-specific rather than current public price claims.

Table 5: Model configurations considered for proposition extraction. Cost estimates are those recorded during pipeline design.

Configuration	Price / 1M tokens	Cost / book
`gemini-3.1-pro-preview-thinking`	$1.0520	$40–50
`gemini-3.1-flash-lite-preview`	$0.1320	approx. $5
`gemini-3-flash-preview-thinking`	$0.2630	approx. $10
Thinking + Flash-Lite hybrid	$0.2630 / $0.1320	approx. $7

A.6 Source-conditioned proposition audit

The audit model, `gemini-3.1-pro-preview-thinking`, receives both the source passage and one extracted structured item. It returns integer scores on five 1–10 scales in a fixed JSON object. Scores 9–10 denote essentially publication-ready extraction, 5–8 identify localized omissions or formatting issues, and 1–4 denote a severe failure. The axes are:

- **Completeness**: all assumptions, scope restrictions, and core formulas in the source proposition are present.
- **Accuracy**: the item preserves the source logic and introduces no unsupported proof, altered relation, or hallucinated statement. This is the critical axis.
- **LaTeX formatting**: mathematical expressions are syntactically valid and use appropriate inline or display delimiters.
- **Self-containment**: variables and assumptions are defined well enough to understand the item without the surrounding page.
- **Atomicity**: the item captures one definition, lemma, theorem, or proof unit rather than merging unrelated material.

The enforced output schema is:

```
{
  "Completeness": <1--10>,
  "Accuracy": <1--10>,
  "LaTeX_Formatting": <1--10>,
  "Self_Containment": <1--10>,
  "Atomicity": <1--10>
}
```

B Extended Characterization Analysis

This appendix provides the detailed corpus characterization summarized in Section 3.2. We first describe subject and language coverage and task composition, then examine quality behavior, provenance, reasoning scope, structural complexity, and confidence.

B.1 Subject and Language Coverage

Figure 9(a)-(b) reports QA volume and distinct source book count for the fifteen most represented subjects. The ten leading subjects account for 84.4% of records. Mathematical analysis draws 66,409 records from 307 books (216 per book), while graph theory draws 24,406 records from only 25 books (976 per book), reflecting broad coverage for analysis-heavy subjects against concentration on specialized books elsewhere. Figure 9(c) crosses source language with educational level: competition records are 92.4% Chinese, whereas graduate records are 66.6% English.

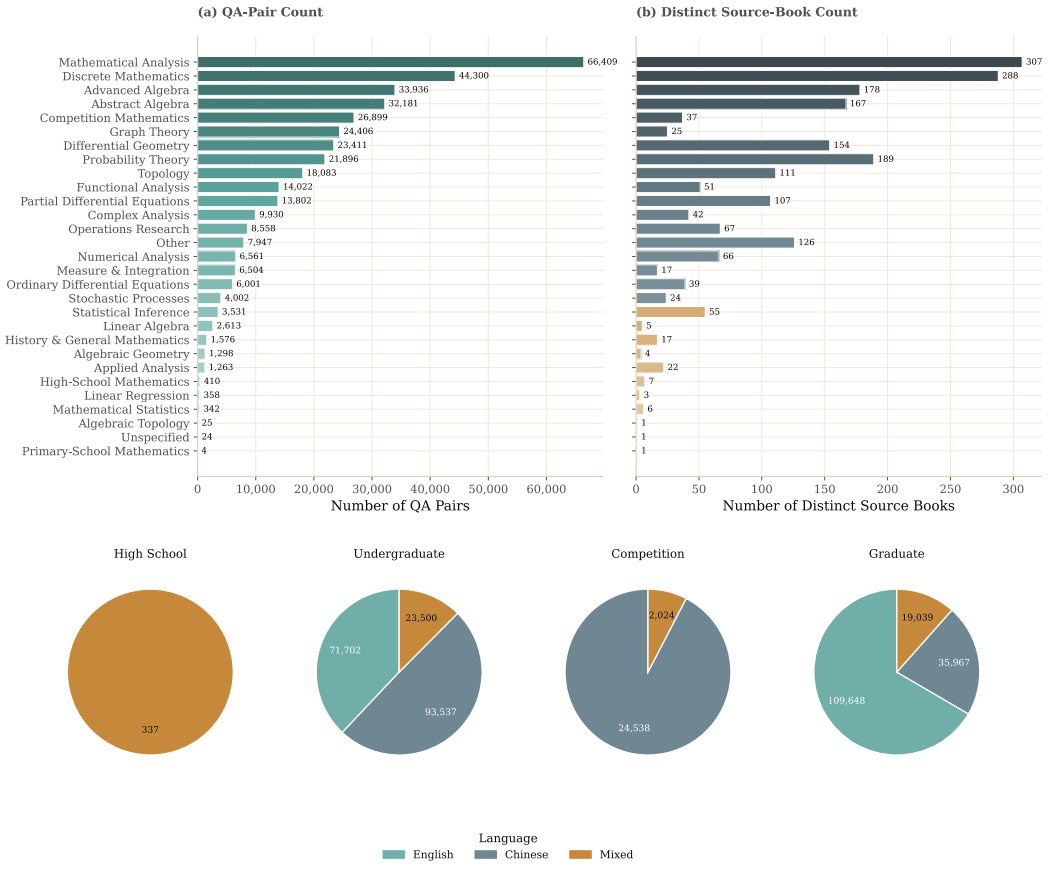


Figure 9: (a) QA volume and (b) distinct source book count across all subjects, sharing a common subject ordering. (c) Source language composition within each educational level.

B.2 Task and Answer-Format Composition

Figure 10(a) shows that proof generation accounts for 79.0% of records, an order of magnitude above every other task type. Figure 10(b) crosses task type with answer format. Derivation and calculation binds almost deterministically to the derivation format at 87.0%, while proof generation splits between full textbook proof (51.0%) and concise final proof (41.9%).

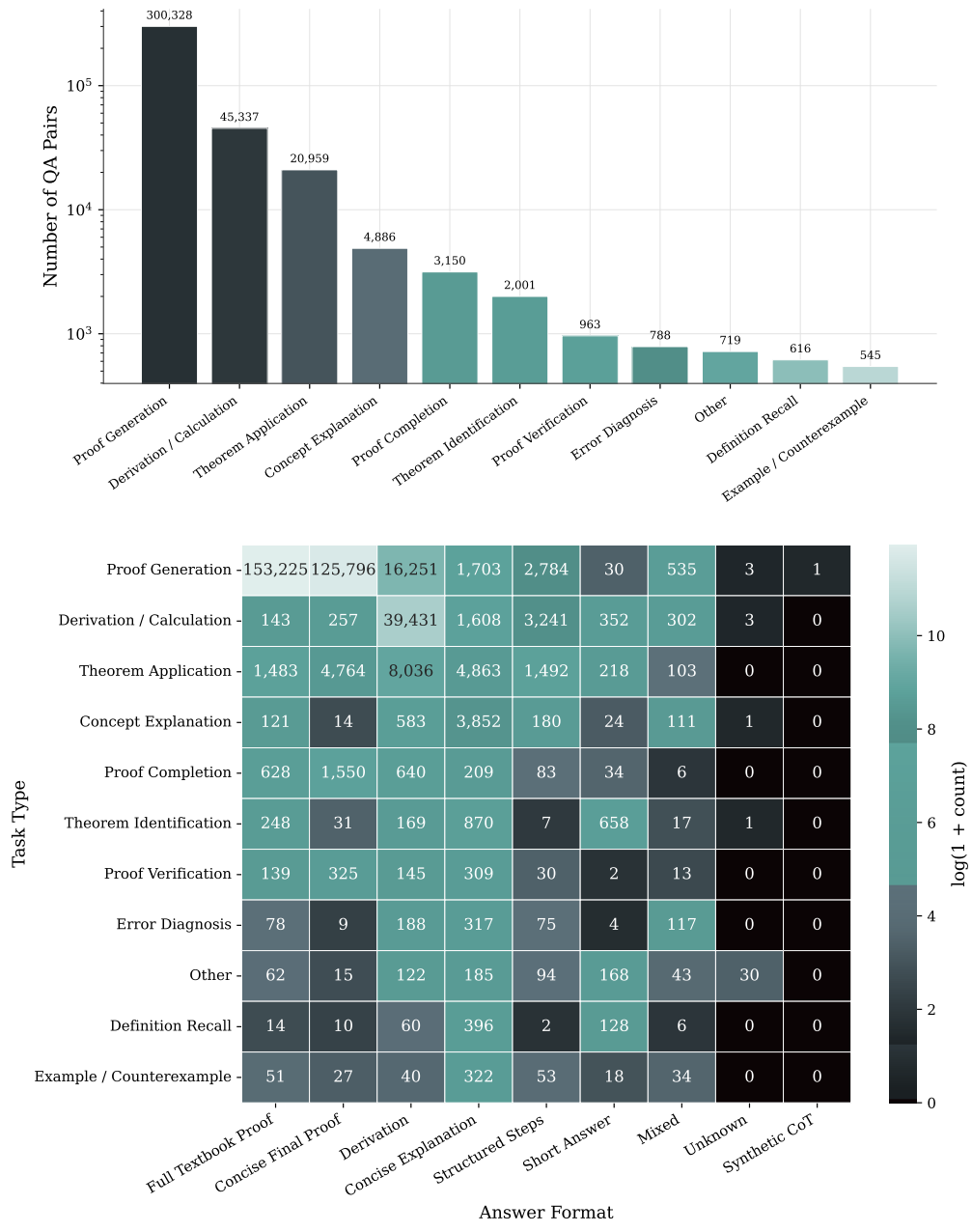


Figure 10: (a) Distribution of eleven fine grained task types on a logarithmic count axis. (b) Joint distribution of task type and answer format on a logarithmic color scale.

B.3 Quality Behavior at Subject and Dimension Granularity

Figure 11(a) shows that tier composition varies substantially across subjects. Competition mathematics has the lowest tier A share at 21.6% and the highest tier C share at 71.5%, while subjects such as abstract algebra and topology maintain tier A shares above 39%, indicating that quality outcomes are subject dependent rather than uniform across the corpus. Figure 11(b) disaggregates the audit into nine automated dimensions. Groundedness and evidence alignment both pass below 50% of records and share an identical fail rate of 28.6%, the weakest link in the pipeline, while surface level dimensions such as ambiguity and question answer consistency pass above 90%.



Figure 11: Quality outcomes decomposed by (a) subject, normalized by row, and (b) pass, uncertain, and fail rates across nine audit dimensions.

Figure 12(a) reports the reasons records are excluded outright. Formula broken and incomplete proof are the two leading causes, together accounting for 36,775 excluded records, with the top four reasons covering 80% of all exclusions. Figure 12(b) corroborates the dimension level asymmetry in Figure 11(b) from an independent signal drawn from human review flags. The flag no verbatim evidence covers 48.4% of the corpus, 2.7 times the frequency of the second most common flag, needs math review, indicating that evidence grounding rather than answer correctness or clarity is the dominant source of quality uncertainty.

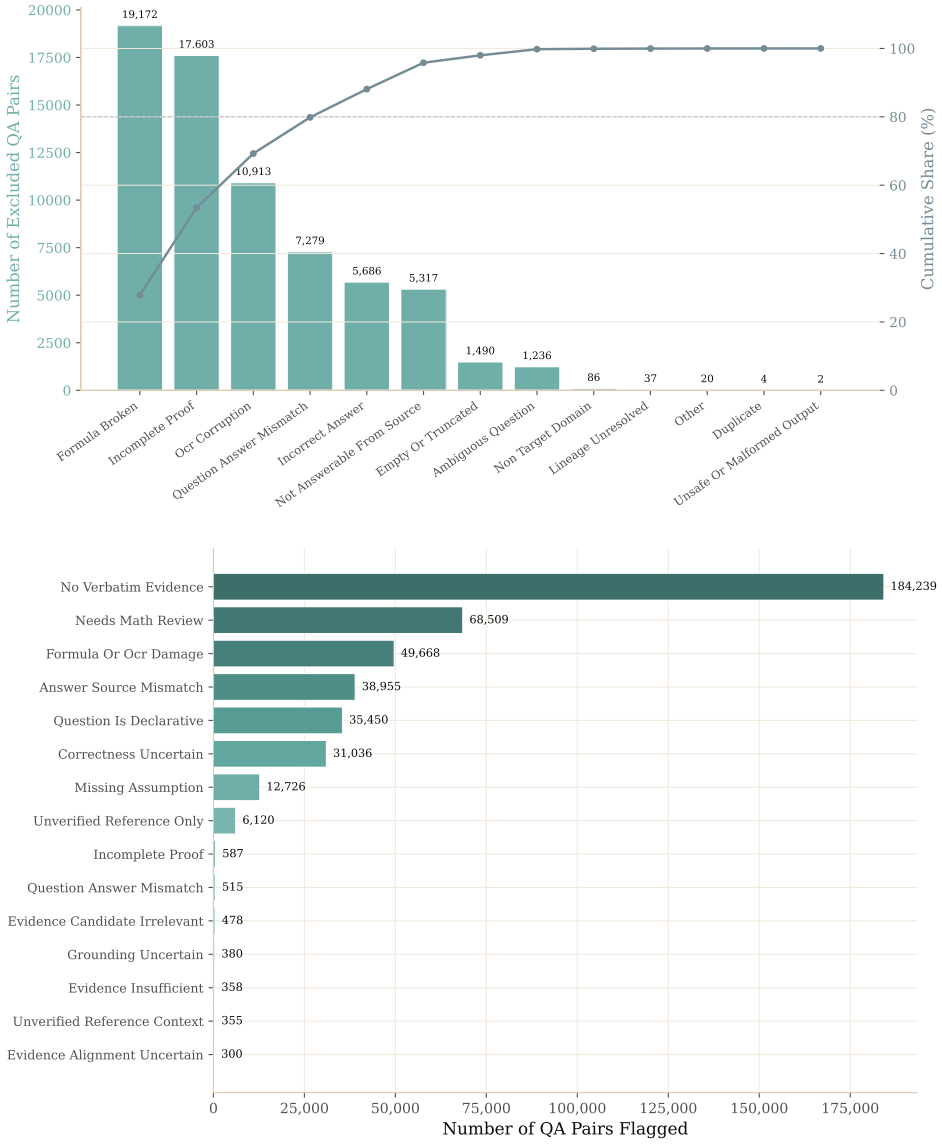


Figure 12: (a) Pareto distribution of record exclusion reasons, with cumulative share on the secondary axis. (b) Fifteen leading human review flags, with multiple flags permitted per record.

B.4 Corpus Redundancy and Generation Provenance

Figure 13(a) reports exact duplicate group sizes. Singleton records account for 99.15% of the corpus and duplicate groups account for only 0.85%, with one outlier group of size twenty one isolated from the remaining groups, all of size four or smaller. Figure 13(b) traces the generation composition introduced in Section 3.2 to individual generator models. The Concept QA and Gap QA streams

draw on disjoint model assignments, with `gemini-3-flash-preview-thinking` responsible for 88.9% of Concept QA records and `gpt-5-mini` responsible for 53.7% of Gap QA records.

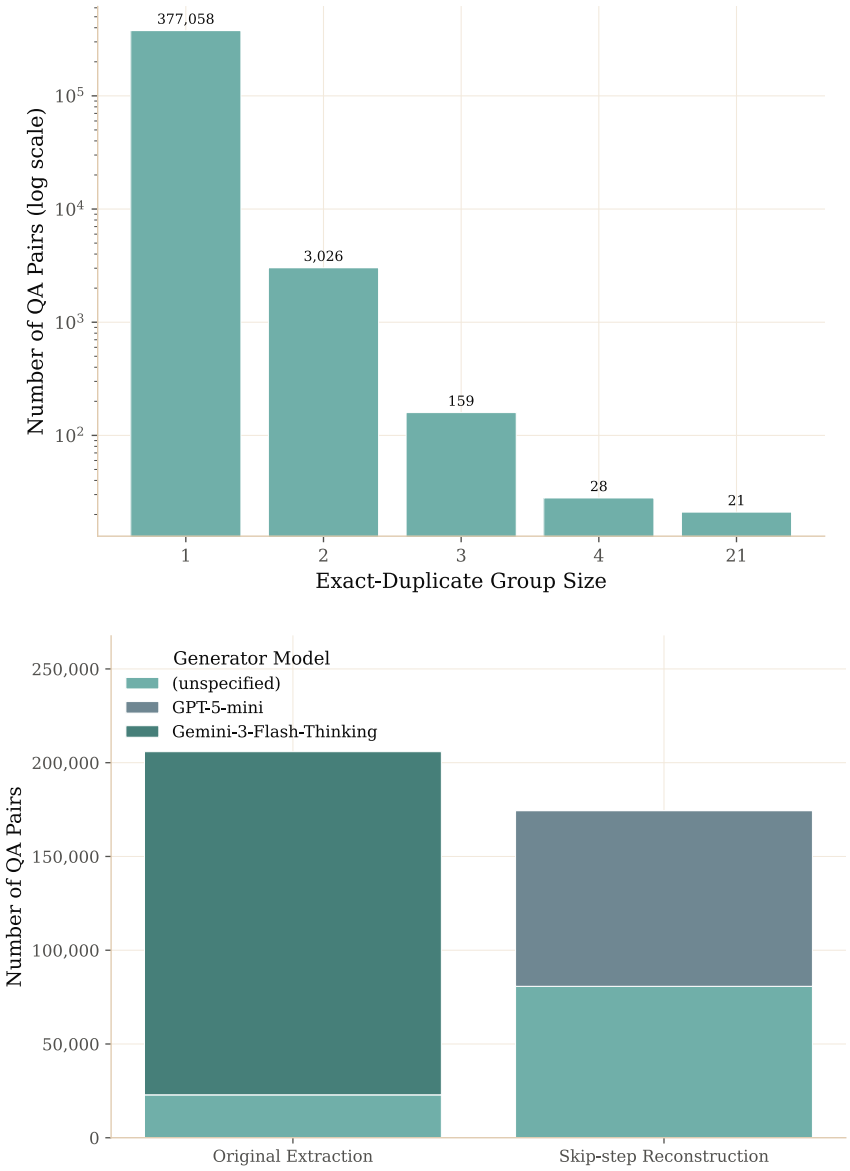


Figure 13: (a) Exact duplicate group sizes on a logarithmic count axis. (b) Generator model assignment by generation stream.

Figure 14 traces the subject-level concentration in Section B.1 to individual sources. The single highest yield book, Jukna's Extremal Combinatorics, contributes 3,049 records, 0.80% of the corpus, and the top twenty books are dominated by English language graph theory and combinatorics textbooks together with Chinese competition mathematics series.

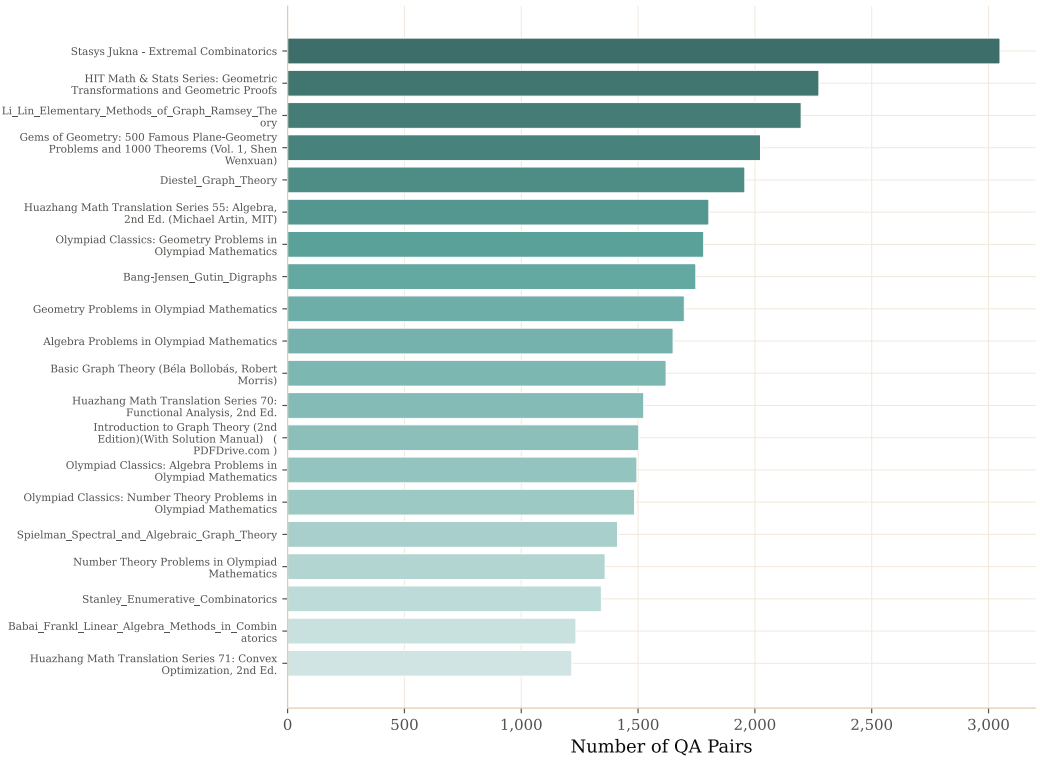


Figure 14: Twenty highest yield source books out of 2,118 distinct books in the corpus.

B.5 Reasoning Scope and Subject Composition Across Level

Figure 15(a) examines how reasoning scope, the breadth of chapters and concepts a record draws on, varies with educational level. The combined share of multi step same chapter and cross chapter reasoning reaches 32.5% at the graduate level, compared with 18.7% for undergraduate and 17.7% for competition. Competition and undergraduate report similar single concept shares despite differing in solving difficulty, indicating that reasoning scope captures curriculum breadth rather than solving skill. Figure 15(b) places this finding in the context of subject coverage from Section B.1. Mathematical analysis is the only leading subject with substantial coverage at both undergraduate and graduate levels, while most other leading subjects concentrate on a single level, explaining why level and subject interact strongly in downstream evaluation splits.



Figure 15: (a) Percentage composition of reasoning scope within each educational level. (b) Joint distribution of the fifteen most represented subjects and educational level.

B.6 Structural Complexity and Confidence Dynamics

Each record carries structural metadata (curriculum stage, structural difficulty, concept count, prerequisite depth) and a model-reported confidence score. Figure 16(a)-(b) shows that prerequisite count correlates with estimated reasoning steps at 0.633, stronger than concept count at 0.522; the fitted surface also rises more steeply along the prerequisite-count axis, making prerequisite depth the dominant driver of reasoning-chain length. Figure 16(c) crosses confidence with curriculum stage and difficulty. Confidence falls with difficulty only at later stages, from 0.876 to 0.807 at stage four and from 0.873 to 0.788 at stage five, while stages one through three remain flat. Figure 16(d) shows that answer length correlates with confidence at -0.231, compared with -0.060 for question length, suggesting that longer derivations drive lower confidence.

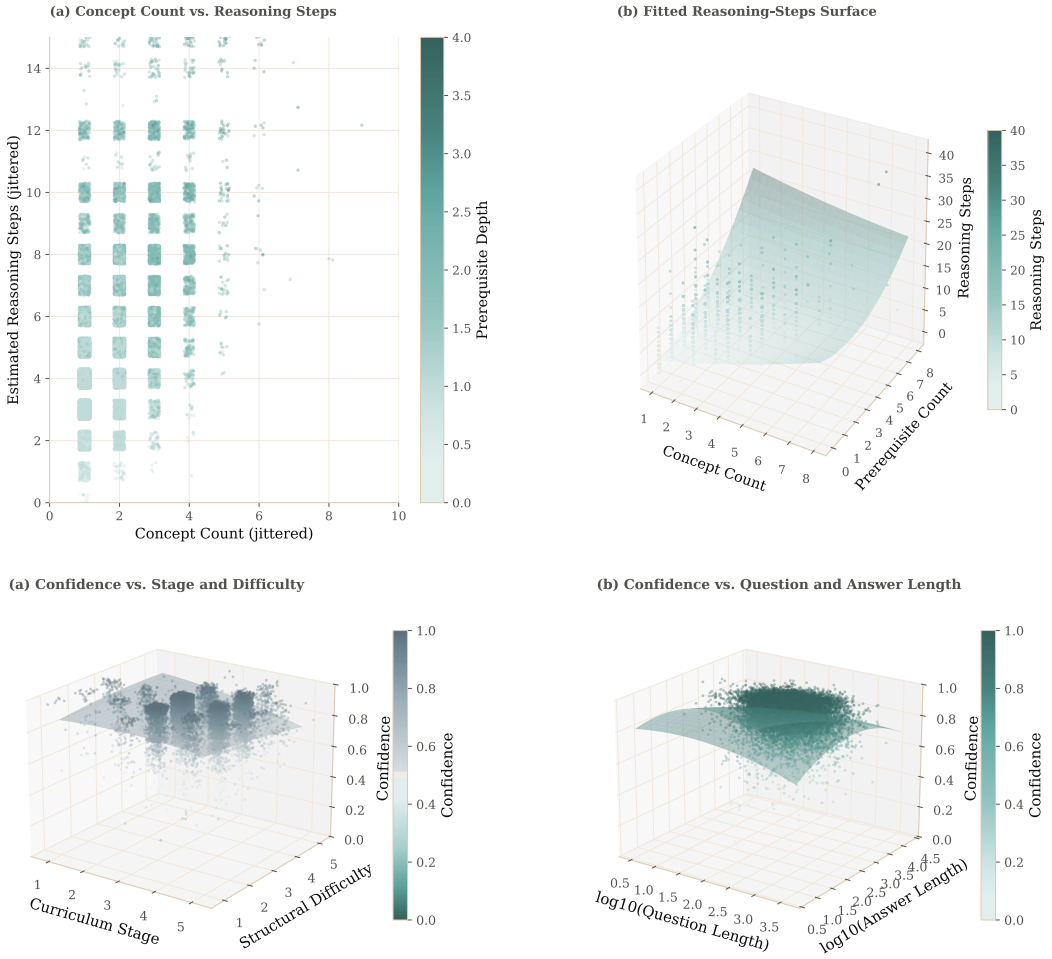
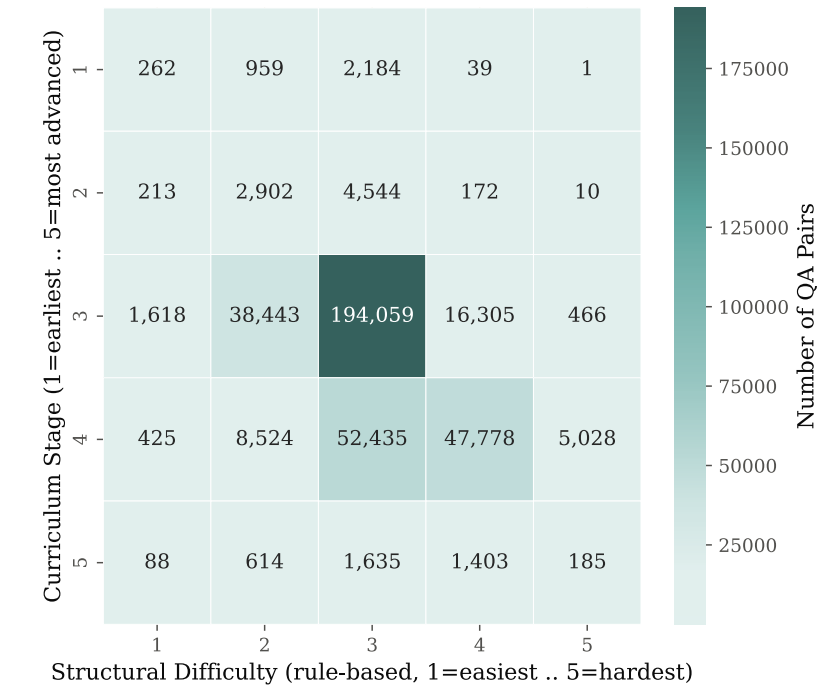


Figure 16: (a) Joint distribution of concept count and estimated reasoning steps, colored by prerequisite depth. (b) Fitted surface of estimated reasoning steps over concept count and prerequisite count. (c) Fitted surface of model confidence over curriculum stage and structural difficulty. (d) Fitted surface of model confidence over log-scaled question length and answer length.

B.7 Structural Difficulty and Record Length

Figure 17(a) reports the underlying joint distribution of curriculum stage and structural difficulty. A single cell at stage three, difficulty three holds 194,059 records, 51.0% of the corpus, and stage three overall accounts for 65.9% of records, indicating that structural difficulty tracks curriculum progression closely rather than varying freely within a stage. Figure 17(b) reports the joint distribution of question and answer length by quality tier. The two lengths are positively associated but retain substantial spread within every tier, so tier assignment cannot be inferred from length alone.



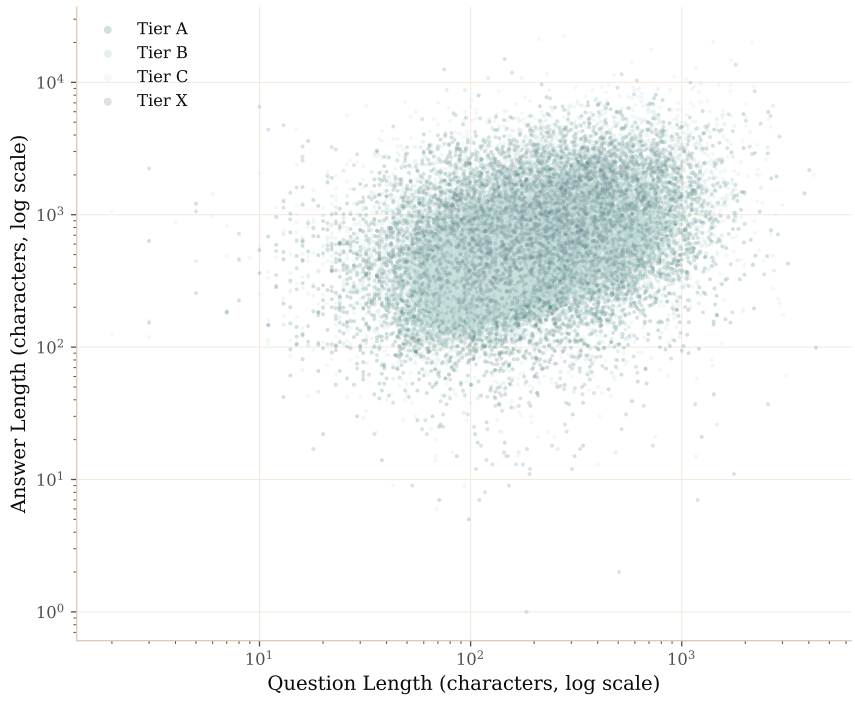


Figure 17: (a) Joint distribution of curriculum stage and structural difficulty. (b) Question and answer length by quality tier, on logarithmic axes.

B.8 Confidence Distributions by Tier and Curriculum Stage

Figure 18(a) reports the marginal confidence distribution by quality tier. Mean confidence follows the order tier A at 0.944, tier B at 0.873, and tier C at 0.821, with tier A showing the tightest spread at a standard deviation of 0.030, indicating that confidence separates tier A cleanly from the lower tiers. Figure 18(b) reports the marginal distribution by curriculum stage, which follows a non monotonic pattern of 0.892, 0.865, 0.882, 0.835, and 0.840 across stages one through five, with stage three holding both the largest sample share at 66.0% of the corpus and the tightest spread.

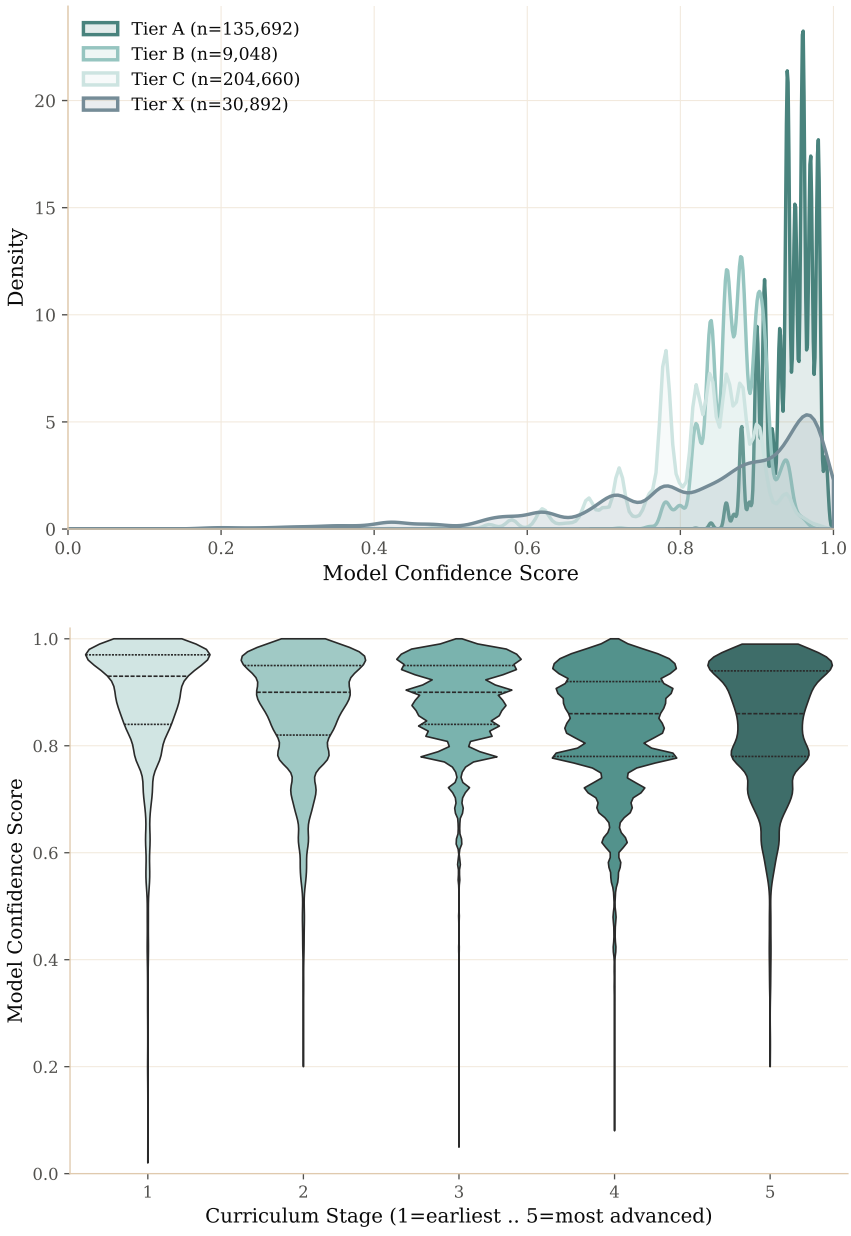


Figure 18: (a) Kernel density estimate of model-reported confidence by quality tier. (b) Distribution of model confidence across the five curriculum stages.

C Synthesized QA Samples

To make the pipeline output concrete, this appendix pairs each synthesized QA with the exact source paragraph it was derived from. We draw all five items from a single textbook—Diestel's *Graph Theory*, one of the graph-theory books in the corpus—so that the reader can trace how the same book supplies items at every level of the L1–L5 ladder used in Section 3.2. Each block below reproduces the source excerpt (grey) and the QA record (blue) verbatim from the intermediate representation and the release table; only whitespace, LaTeX escaping, and page-break markers were adjusted for typesetting.

L1 – Direct fact recall (Section 12.3, Tree-decompositions)

Source excerpt

Let G be a graph, T a tree, and let $\mathcal{V} = (V_t)_{t\in T}$ be a family of *bags* $V_t \subseteq V(G)$ indexed by the nodes t of T. The pair $(T, \mathcal{V})$ is called a *tree-decomposition* of G if it satisfies the following three conditions:

(T1) $V(G) = \bigcup_{t\in T} V_t$;
(T2) for every edge $e \in G$ there exists a $t \in T$ such that both ends of e lie in V_t;
(T3) $V_{t_1} \cap V_{t_3} \subseteq V_{t_2}$ whenever $t_1, t_2, t_3 \in T$ satisfy $t_2 \in t_1 T t_3$.

Synthesized QA

Question. Let G be a graph and let $(T, \mathcal{V})$ be a tree decomposition of G, where $\mathcal{V} = \{V_t : t \in V(T)\}$ is a family of bags such that $\bigcup_{t\in T} V_t = V(G)$. Given a set of vertices $W \subseteq V(G)$, prove that for any $w \in W$ there exists a node $t' \in V(T)$ with $w \in V_{t'}$.

Answer. By definition of a tree decomposition (condition (T1)), $V(G) = \bigcup_{t\in V(T)} V_t$, i.e. the bags cover all vertices of G. Since $W \subseteq V(G)$, every $w \in W$ lies in $V(G) = \bigcup_{t\in V(T)} V_t$, so there exists a node $t' \in V(T)$ with $w \in V_{t'}$. □

L2 – Concept explanation (Section 5.4, List colouring)

Source excerpt

In principle, showing that a given graph is k-choosable is more difficult than proving it to be k-colourable: the latter is just the special case of the former where all lists are equal to $\{1, \dots, k\}$. Thus,

$$\mathrm{ch}(G) \geq \chi(G) \quad \text{and} \quad \mathrm{ch}'(G) \geq \chi'(G)$$

for all graphs G.

Synthesized QA

Question. Explain, in the setting of list colouring, why $\mathrm{ch}(G) \geq \chi(G)$ and $\mathrm{ch}'(G) \geq \chi'(G)$ for every graph G.

Answer. In principle, showing that a given graph is k-choosable is more difficult than proving it to be k-colourable: the latter is just the special case of the former where all lists are equal to $\{1, \dots, k\}$. Hence any list of size k can be specialized to this uniform list, and the two inequalities follow.

L3 – Single-concept short proof (Section 1.2, Proposition 1.2.1)

Source excerpt

Summing $d(v)$ over all $v \in V$ counts every edge twice, so

$$|E| = \tfrac{1}{2} \sum_{v\in V} d(v).$$

Proposition 1.2.1. *The number of vertices of odd degree in a graph is always even.*

Proof. As $|E| = \frac{1}{2}\sum_{v\in V} d(v)$ is an integer, $\sum_{v\in V} d(v)$ is even. □

Synthesized QA

Question. Prove that the number of vertices of odd degree in a graph is always even.

Answer. Since $|E| = \frac{1}{2}\sum_{v\in V} d(v)$ is an integer, $\sum_{v\in V} d(v)$ is even. Reducing this sum modulo 2, each even-degree vertex contributes 0 and each odd-degree vertex contributes 1, so $|V_{\text{odd}}| \equiv 0 \pmod 2$; the set of odd-degree vertices has even cardinality. □

L4 – Chapter-local multi-step proof (Section 1.9, Lemma 1.9.3)

Source excerpt

Proposition 1.9.2. Together with $\emptyset$, the cuts in G form a subspace $\mathcal{B}(G)$ of $\mathcal{E}(G)$. This space is generated by atomic cuts. *(A minimal non-empty cut is a bond.)*

Lemma 1.9.3. *Every cut is a disjoint union of bonds.*

Proof. We apply induction on the size of the cut F considered. For $F = \emptyset$ the assertion is trivial (with the empty union). If $F \neq \emptyset$ is not itself a bond, it properly contains some other non-empty cut F'. By Proposition 1.9.2, $F \setminus F' = F + F'$ is also a smaller non-empty cut. By the induction hypothesis, both F' and $F \setminus F'$ are disjoint unions of bonds, and hence so is F. □

Synthesized QA

Question. Prove that every cut in a graph is a disjoint union of bonds.

Answer. We apply induction on the size of the cut F. For $F = \emptyset$ the assertion is trivial (with the empty union). If $F \neq \emptyset$ is not itself a bond, it properly contains some other non-empty cut F'. By Proposition 1.9.2, $F \setminus F' = F + F'$ is also a smaller non-empty cut. By the induction hypothesis, both F' and $F \setminus F'$ are disjoint unions of bonds, and hence so is F. □

L5 – Cross-chapter synthesis (Section 12.7, Corollary 12.7.3)

Source excerpt

Corollary 12.7.3. *For every surface S there exists a finite set of graphs $H_1, \ldots, H_n$ such that a graph is embeddable in S if and only if it contains none of $H_1, \ldots, H_n$ as a minor.*

While Corollary 12.7.3 is immediate from the graph minor theorem, it can also be proved more directly. The main step is to prove that the graphs in $\mathcal{K}_{\mathcal{P}(S)}$ do not contain arbitrarily large grids as minors (Lemma 12.7.4). Then their tree-width is bounded (Theorem 12.6.3), so $\mathcal{K}_{\mathcal{P}(S)}$ is well-quasi-ordered (Theorem 12.4.2) and therefore finite.

Synthesized QA

Question. Show that for every surface S there exists a finite set of graphs $H_1, \ldots, H_n$ such that a graph is embeddable in S if and only if it contains none of $H_1, \ldots, H_n$ as a minor.

Answer. By their minimality, the graphs in the Kuratowski set $\mathcal{K}_{\mathcal{P}(S)}$ are incomparable under the minor relation. If their tree-width is bounded, then $\mathcal{K}_{\mathcal{P}(S)}$ is well-quasi-ordered by the minor relation (Theorem 12.4.2), and hence must be finite. So assume their tree-width is unbounded, and let r be as in Lemma 12.7.4. By Theorem 12.6.3, some $H \in \mathcal{K}_{\mathcal{P}(S)}$ has a grid minor large enough to contain H^r. By Proposition 1.7.3, H^r is a topological minor of H, contrary to the choice of r. □

Taken together, the five blocks show the ladder concretely: L1 only unrolls a definition (T1 of tree-decomposition); L2 verbalizes the "why" behind an inequality already stated in the section; L3 is a one-line proof anchored in a single identity from the same section; L4 composes two consecutive results within one section by induction; and L5 chains a grid-minor lemma, a tree-width theorem, and a well-quasi-ordering theorem from three different chapters of the same book.

D Evaluation Protocols

D.1 Evaluation set construction

TOME-LongForm. The long-form component of TOME-Eval contains 300 textbook-native problems with reference solutions. It covers six subjects, with 25 problems sampled from the first half and 25 from the second half of the chapter sequence in each subject (Table 6). For a problem in the zero-based chapter position i of a book with N chapters, we define the first half by $i/N < 0.5$ and the second half by $i/N \geq 0.5$. This split is a coarse measure of chapter position rather than a claim about problem difficulty or prerequisite depth. This 300-problem, six-subject suite is the frozen version used in the completed experiments; it supersedes an earlier 500-problem evaluation snapshot.

Table 6: Composition of the 300-problem book-held-out evaluation set.

Subject	First	Second	Books	Proof	Other	Total
Discrete mathematics	25	25	10	39	11	50
Mathematical analysis	25	25	10	29	21	50
Abstract algebra	25	25	10	46	4	50
Probability theory	25	25	10	28	22	50
Differential geometry	25	25	12	43	7	50
Topology	25	25	10	42	8	50
Total	**150**	**150**	**62**	**227**	**73**	**300**

We freeze source books before constructing any training subset. The held-out manifest contains 64 books: ten per subject except differential geometry, for which fourteen books are held out to provide sufficient coverage of later chapters. All 15,661 training records associated with these books are removed from every training scale, leaving 364,631 eligible records. The final problems come from 62 of the held-out books and 212 distinct book–chapter pairs; the two remaining books stay excluded from training although none of their problems is selected.

Filtering and selection. Starting from 5,058 problems in the held-out books, deterministic filters retain 4,316 candidates. We require a 21–4,000 character question and a 200–8,000 character reference solution, and exclude empty, image-dependent, context-dependent, duplicated, and exact train-overlapping items. A Luna-based screen then annotates reference completeness, self-containment, mathematical relevance, task type, and difficulty. We retain only candidates judged self-contained and mathematical with a usable reference derivation. Within each subject–chapter-half cell, we sample across books, cap each book at six problems, and stratify by the auxiliary difficulty label. The base model is not used to select problems.

All 300 selected items are subsequently reviewed and accepted by a mathematics PhD. They comprise 227 proof, 50 calculation, 18 construction, and five explanation problems. We therefore describe this suite as *book-held-out* and *proof-heavy*, rather than as a general-purpose test of all mathematical capabilities. We freeze the resulting set before running any model evaluation and use the same version for all comparisons.

TOME-Objective. The objective component of TOME-Eval contains 145 unique problems from 19 subject categories: 81 fill-in-the-blank, 45 multiple-choice, and 19 true–false questions. The suite is complemented by the 30 problems from AIME 2026 and the 500-problem MATH-500 subset of MATH (Hendrycks et al., 2021). Every item has a single reference answer; these benchmarks evaluate final-answer correctness rather than agreement with a particular derivation.

Formal benchmarks. We also evaluate on MiniF2F (Zheng et al., 2021) and ProofNet (Azerbayev et al., 2023). For both datasets we retain the official task definition, split, prompt format, checker, and native metric. In particular, we do not apply either of our natural-language judges to these benchmarks.

D.2 Response generation

For TOME-LongForm, the compared models receive the same question-only prompt and decoding configuration. The reference solution is withheld from the solver. Sixteen rollouts over 300 problems produce 4,800 responses per model.

Table 7: Client-side settings for generating evaluation responses.

Setting	Value
Responses per question	16
Temperature	0.6
Top-p	0.8
Maximum output tokens	16,384
Completions per request	1
Presence / frequency penalty	0 / 0

The objective benchmarks use fixed, benchmark-specific decoding configurations. We sample eight responses per TOME-Objective item, 32 per AIME 2026 item, and four per MATH-500 item, keeping the corresponding configuration fixed across compared models.

D.3 Reference-guided step verification

Input preparation. Before grading, the verifier removes marked thinking segments and retains the final answer when explicit final-channel delimiters are present. It also truncates text at an end-of-message marker. Existing numbered step tags are sorted by their index; otherwise, the answer is split at blank lines. The resulting segments are wrapped in zero-based step tags. Thus, an untagged answer is graded at paragraph-level granularity rather than using an additional model to infer mathematical steps. The judge receives the problem, the reference solution, and the segmented model response.

Rubric. The rubric prioritizes mathematical validity, problem-specific constraints, and sufficient justification. A reference solution is an anchor, not the only acceptable proof: alternative methods, reordered steps, and valid shortcuts are allowed. For numerical or algebraic questions, the final answer must be mathematically correct relative to the reference; a matching conclusion alone is insufficient if its derivation is invalid. Unsupported claims and unjustified leaps are treated conservatively. Invoking an external theorem is insufficient without the hypotheses and explanation needed to check its application. Proof-embedded instructions attempting to relax these standards are ignored. The requested output contains `assessment`, `errors`, and `first_error_step` XML fields; the last field is -1 if no error is found.

Eight-review aggregation. Following pessimistic verification (Huang et al., 2025b), each response receives eight separate reviews from `gpt-5.6-luna`. Let e_{ijr} be the first-error index from review r of response j to problem i. For eight valid reviews, the binary correctness decision is

$$y_{ij} = \prod_{r=1}^{8} \mathbf{1}[e_{ijr} = -1]. \tag{1}$$

Thus, one error vote is sufficient to reject a response; this is pessimistic aggregation rather than majority voting. For rejected responses, we retain the earliest flagged step. The evaluation controller accepts a judgment only when all eight reviews complete successfully and the parsed record contains no API error; incomplete rows are removed and retried for up to three attempts.

D.4 Human calibration of the pessimistic verifier

Scope and sampling. We use a frozen human-calibration protocol to measure both erroneous acceptance and the over-rejection risk introduced by eight-review pessimistic aggregation. Before inspecting any human label, we freeze the verifier prompts, decoding configuration, aggregation rule, model outputs, and sampling procedure. We then draw 600 responses from the pooled TOME-

LongForm evaluations, with 100 responses from each of the six subjects. Let

$$a_{ij} = \sum_{r=1}^{8} \mathbf{1}[e_{ijr} = -1] \tag{2}$$

be the number of reviews that find no error. Within each subject, we target 25 responses from each of four agreement strata: $a_{ij} = 8$, $a_{ij} = 7$, $1 \leq a_{ij} \leq 6$, and $a_{ij} = 0$. The $a_{ij} = 7$ stratum is sampled separately because a single error vote changes the final pessimistic decision. Within each stratum, responses are distributed across the compared models and major experiment families as evenly as availability permits. If a stratum contains fewer than 25 responses in a subject, we annotate all available cases and reallocate the remainder to the nearest agreement stratum. We retain each response's sampling probability so that population-level estimates can be inverse-probability weighted rather than treating this deliberately balanced sample as representative of the raw response distribution.

Blinding and annotation materials. Annotators are mathematics PhD holders, or comparably qualified researchers, with expertise appropriate to the assigned subject. For each response, an annotator receives the problem, the reference solution, and the same segmented, step-indexed model response supplied to the automatic verifier. Model identity, training condition, automatic judgments, review agreement, and aggregate score are hidden. The reference is treated as an anchor rather than a unique target: alternative proof methods, reordered arguments, and valid shortcuts are accepted when they establish the requested result. This response-level audit is separate from the earlier human review of the evaluation questions and reference solutions.

Human labels. Each response receives exactly one of the following labels.

- **Fully correct:** the conclusion and supporting argument are mathematically valid and sufficiently justified.
- **Correct with a non-substantive issue:** the response contains only a stylistic, notational, typographical, or presentation defect that does not affect mathematical validity.
- **Substantive error:** the response contains a false statement, invalid inference, missing required hypothesis, circular argument, incorrect calculation, materially incomplete proof, or incorrect final answer.
- **Unjudgeable:** ambiguity or a defect in the problem, reference, or response prevents a reliable correctness decision.

The first two labels map to human acceptance and *Substantive error* maps to rejection. *Unjudgeable* cases require adjudication and are not mapped to a binary label beforehand. For every rejected response, the annotator records the zero-based index of the first substantive error and selects an error category: false claim, invalid inference, missing hypothesis, theorem misuse, incomplete proof, algebraic or numerical error, final-answer error, or other. Every annotation also includes a short justification and a high, medium, or low confidence rating. Purely stylistic differences and harmless notation changes must not trigger rejection.

Independent review and adjudication. One qualified annotator labels all 600 responses. A second qualified annotator independently labels a stratified 30% subset (180 responses), preserving the subject and agreement-stratum proportions and remaining blind to the first annotation. All binary disagreements, all *Unjudgeable* labels, and all low-confidence cases are adjudicated by a third domain-qualified annotator. If the adjudicator finds the evaluation item itself invalid, that problem is removed for every model and all affected aggregate scores are recomputed.

Calibration statistics. Let $h_{ij} = 1$ denote the adjudicated human acceptance label and let y_{ij} be the pessimistic verifier decision in Equation 1. Using the recorded sampling weights, we report the false-acceptance rate $\Pr(h_{ij} = 0 \mid y_{ij} = 1)$, the false-rejection rate $\Pr(h_{ij} = 1 \mid y_{ij} = 0)$, overall agreement, and the human acceptance rate within each value of a_{ij}. We additionally report raw agreement and Cohen's κ between the two human annotators, together with exact and within-one-step agreement for the first-error index when both reject. All quantities are reported overall and by subject with 95% bootstrap confidence intervals clustered by problem. Finally, we recompute

model-level correctness on the weighted human-labeled subset to test whether the principal model ranking and pairwise conclusions agree with the automatic evaluation.

Table 8: Human calibration of the eight-review pessimistic verifier. All rate entries are percentages and should be reported as the weighted point estimate followed by the problem-clustered 95% bootstrap confidence interval in brackets. H–L denotes human–Luna agreement; H–H denotes agreement between the two human annotators.

Panel A: Calibration by subject

Group	Annotated n	Human accept	False accept	False reject	H–L agree
Abstract algebra	100	42.1 [35.8, 48.7]	9.8 [4.2, 17.6]	19.3 [13.1, 26.5]	84.7 [79.6, 89.2]
Differential geometry	100	37.8 [31.5, 44.4]	11.4 [5.6, 19.2]	20.7 [14.5, 28.0]	83.5 [78.3, 88.1]
Discrete mathematics	100	52.3 [45.8, 58.7]	7.2 [2.6, 14.0]	16.9 [11.2, 23.7]	87.4 [82.6, 91.5]
Mathematical analysis	100	54.6 [48.1, 60.9]	6.8 [2.4, 13.4]	15.8 [10.3, 22.4]	88.1 [83.5, 92.0]
Probability	100	55.4 [48.9, 61.6]	7.9 [3.0, 15.0]	17.5 [11.7, 24.3]	86.6 [81.7, 90.8]
Topology	100	45.2 [38.7, 51.8]	9.1 [3.8, 16.8]	18.6 [12.5, 25.7]	85.2 [80.2, 89.6]
Overall	**600**	**47.9 [44.6, 51.2]**	**8.7 [6.3, 11.5]**	**18.1 [15.3, 21.1]**	**85.9 [83.8, 87.9]**

Panel B: Calibration by number of no-error votes

Luna votes a_{ij}	Annotated n	Population share	Human accept	Human substantive error	H–L agree
8 of 8	150	42.3 [40.9, 43.7]	91.3 [86.4, 94.8]	7.5 [4.1, 12.3]	91.3 [86.4, 94.8]
7 of 8	150	7.8 [7.0, 8.7]	68.4 [60.5, 75.6]	30.2 [23.1, 38.2]	31.6 [24.4, 39.5]
1–6 of 8	150	18.1 [17.0, 19.3]	22.1 [15.9, 29.4]	76.4 [69.0, 82.7]	77.9 [70.6, 84.1]
0 of 8	150	31.8 [30.5, 33.1]	2.8 [0.9, 6.4]	96.4 [92.3, 98.6]	97.2 [93.6, 99.1]

Panel C: Human–human reliability on the double-coded subset

Subset	Double-coded n	H–H agree	Cohen's κ	Exact error step	Within one step
Stratified 30% subset	180	89.4 [84.9, 93.1]	0.771 [0.687, 0.842]	52.8 [43.2, 62.1]	76.9 [68.5, 84.1]

The unanimous eight-review decision rule is fixed before the human labels are opened; the human study validates that rule rather than selecting a favorable threshold. Any subsequent change to the aggregation rule must be disclosed and applied to every evaluated model and condition.

D.5 Objective and formal scoring

For TOME-Objective, a low-temperature Qwen2.5-32B-Instruct judge compares the extracted final answer with the reference. It accepts mathematically equivalent expressions and requires every component of a multipart or multiple-selection answer to be correct. The judge receives the problem, reference answer, and prediction, but not the worked reference solution. AIME 2026 and MATH-500 first use the rule-based `MATHVerifyEvaluator`; cases not resolved by the rule-based parser fall back to the same low-temperature LLM equivalence judge. These evaluators score answer correctness only and do not assess the generated reasoning path. MiniF2F and ProofNet are instead scored by their official benchmark evaluators.

D.6 Metrics and validity checks

For a problem with n sampled responses, of which c_i are correct, we use the standard unbiased estimator (Chen et al., 2021)

$$\widehat{\text{pass@}k} = \frac{1}{N}\sum_{i=1}^{N}\left(1 - \frac{\binom{n-c_i}{k}}{\binom{n}{k}}\right). \tag{3}$$

For TOME-LongForm, $n = 16$ and we report pass@1, pass@2, pass@4, pass@8, and pass@16, with pass@1 and pass@16 as the primary metrics. Because $n = k = 16$, pass@16 is also the observed fraction of problems solved at least once. We report per-subject scores and their macro average; the equal 50-problem subject sizes make this macro average equal to the overall problem average. For TOME-Objective, the mean correctness across its eight samples estimates pass@1; pass@k is computed separately when reported.

For every reported model, we require the complete expected number of rollouts for every problem. For natural-language proof judgments, we additionally require eight successful reviews per response and rejudge any incomplete row. We preserve raw judge outputs and parsed first-error indices for

audit. Since the pessimistic verifier can amplify judge false positives, we interpret its absolute scores conservatively and use the same frozen judge configuration for all paired model comparisons.

E Full-Corpus Comparison: Extended Results

This appendix expands Section 4.2 ("Full-Corpus Comparison") with uncertainty estimates and per-corpus training configurations that were omitted from the main text for space. All fine-tuned models share the Qwen3-30B-A3B-Base initialization and the evaluation protocol described in Appendix D; the compared corpora differ only in their supervision data.

E.1 Compared corpora

Public SFT corpora. We use the official Hugging Face releases of OpenMathInstruct-1 (`nvidia/OpenMathInstruct-1`, revision `4627efae`; abbreviated OpenMath1 in our tables), MetaMathQA (`meta-math/MetaMathQA`, revision `aa4f34d3`), and MathInstruct (`TIGER-Lab/MathInstruct`, revision `b4fdc323`) (Toshniwal et al., 2024b; Yu et al., 2023; Yue et al., 2024). OpenMathInstruct-1 is distributed under the NVIDIA Dataset License and MetaMathQA is labeled MIT; MathInstruct combines source datasets under their respective licenses, including MIT, Apache-2.0, and CC BY-NC 4.0 components. We retain the released training examples (the correct-solution subset for OpenMathInstruct-1), map their native question and response fields to the same user–assistant training template, and apply no additional cross-corpus deduplication.

TOME-Math (ours). We use the frozen full-corpus snapshot produced by the pipeline in Appendix A: 380,292 records before evaluation holdout filtering. Removing all 15,661 records from held-out evaluation books leaves 364,631 training examples. We retain all remaining quality tiers and the corpus's natural subject distribution, without subject rebalancing or further deduplication.

Table 9: Training corpora compared in Section 4.2.

Corpus	Training examples	Training tokens	Notes
OpenMath1	1.82M	368.00M	Full corpus
MetaMath	395k	96.94M	Full corpus
MathInstruct	262k	64.10M	Full corpus
TOME-Math (ours)	364,631	170.92M	Held-out books excluded

E.2 Training configuration

Each model is trained for exactly one full epoch over its respective training corpus. We use the same optimizer, learning-rate schedule, sequence length, effective batch size, and data format across corpora. Because the corpora differ in size and response length, their total training tokens and numbers of optimization steps are not matched. The comparison therefore measures the end-to-end utility of each complete data resource under a uniform one-pass recipe rather than compute-matched per-example data quality. The complete TOME-Math release contains 380,292 examples and 195.18M tokens; excluding the 15,661 records associated with the held-out evaluation books leaves 364,631 training examples and approximately 170.92M training tokens.

Table 10: Fine-tuning configuration for the full-corpus comparison.

Setting	Value
Base model	Qwen3-30B-A3B-Base
Optimizer	AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$; weight decay $= 0.1$)
Learning rate	8×10^{-5}, decayed to a minimum of 8×10^{-6}
Warmup / schedule	3% warmup followed by cosine decay (0 warmup steps in the 18-step run)
Sequence length	131,072 tokens; global packing
Effective batch size	64 (micro-batch size 1/GPU; gradient accumulation 4)
Epochs	1
Hardware	16× NVIDIA H200 (2 nodes × 8 GPUs)

E.3 All benchmark results

For completeness, Table 11 reproduces the six-benchmark comparison from Table 2. We estimate pass@1 from 16 responses per problem for TOME-LF, MiniF2F, and ProofNet, eight for TOME-Objective, four for MATH-500, and 32 for AIME 2026. MiniF2F and ProofNet are evaluated as natural-language proof tasks with the Luna proxy judge rather than their official formal checkers. We compute 95% problem-clustered bootstrap confidence intervals using 2,000 resamples. AIME 2026 has the widest intervals because it contains only 30 problems. The remaining scoring details follow Appendix D.

Table 11: Full-corpus SFT performance (%) across six benchmarks. Each entry reports pass@1 $\pm$ the average half-width of its 95% problem-clustered bootstrap confidence interval.

Model (training data)	TOME-LF	TOME-Obj.	MiniF2F	ProofNet	MATH-500	AIME-26
Qwen3-30B-A3B-Base	13.25 $\pm$3.35	32.93 $\pm$8.20	34.45 $\pm$6.00	17.14 $\pm$3.95	68.80 $\pm$4.40	8.44 $\pm$8.10
OpenMath1 (1.82M)	20.16 $\pm$4.10	33.86 $\pm$8.30	22.87 $\pm$5.40	19.25 $\pm$4.15	70.40 $\pm$4.30	8.21 $\pm$8.05
MetaMath (395k)	28.73 $\pm$4.85	34.66 $\pm$8.35	40.70 $\pm$6.15	22.55 $\pm$4.40	71.20 $\pm$4.30	9.98 $\pm$8.70
MathInstruct (262k)	35.87 $\pm$5.20	40.69 $\pm$8.65	52.05 $\pm$6.20	32.93 $\pm$5.25	73.05 $\pm$4.20	11.98 $\pm$9.10
TOME-Math (ours; 364.6k)	43.86 $\pm$5.40	43.96 $\pm$8.75	57.43 $\pm$6.10	42.98 $\pm$5.45	76.55 $\pm$3.90	10.19 $\pm$8.75

F Training-Free Dependency-Context Study

Evaluation set. The study in Section 5.1 uses the 300-problem TOME-LongForm evaluation set described in Appendix D. It contains 50 textbook-native problems from each of abstract algebra, differential geometry, discrete mathematics, mathematical analysis, probability, and topology. The source books are excluded from the TOME-Math training corpus. Every problem is evaluated under all five context conditions, making the comparison paired at the problem level.

Retrieval inventories. Candidate context is drawn only from TOME-Math records associated with the exercise's source book. The proposition inventory contains extracted definitions, lemmas, theorems, propositions, corollaries, and their associated proof records. The expanded inventory adds the self-contained local proof obligations and answers from the Gap QA stream. Source type, chapter position, and recovered dependency metadata are retained for retrieval.

The five conditions are constructed as follows.

- **Closed:** provide only the exercise.
- **Random same-book:** sample entries uniformly from the same source book, without using chapter or dependency information.
- **Local chapter:** restrict candidates to the exercise chapter and its immediately preceding chapters, adding locality while ignoring recovered cross-chapter dependencies.
- **TOME (props only):** retrieve from the proposition inventory using chapter position and the recovered proposition dependencies.
- **TOME (props + gap):** use the same structure-guided retrieval procedure over the union of the proposition and Gap QA inventories.

Context and leakage controls. All four context-augmented conditions use the same context-token budget and the same prompt template; only the candidate pool and retrieval constraint change. Retrieved entries are placed before the exercise and identified by their source type. The reference solution is never shown to the solver. Candidates that reproduce the target exercise or overlap with its reference answer under the frozen leakage filter are removed before retrieval. The same filtered inventory is used to construct the random and local baselines, so the comparison does not give the TOME conditions access to otherwise unavailable source material.

Generation and scoring. The solver is the pretrained Qwen3-30B-A3B-Base in every condition. Each problem receives 16 sampled responses under the decoding configuration in Table 7; adding

context does not change the model weights, prompt instruction, or sampling parameters. Responses are graded with the same reference-guided eight-review verifier used for TOME-LongForm. We compute pass@1 as the mean binary correctness across the 16 responses for each problem, average within each of the six equally sized subjects, and report their unweighted mean in the Overall column of Table 3.

Result interpretation. The aggregate score increases from 13.25% without context to 14.00% with random same-book context and 14.71% with local context. Proposition-based TOME retrieval reaches 15.54%, and adding the Gap QA inventory raises the score to 17.38%. The final condition is best in every subject, although the intermediate conditions are not monotonically ordered within every subject. Accordingly, the result supports the complete structure-guided retrieval design and the complementary value of Gap QA; it does not by itself attribute the gain exclusively to any single dependency edge or retrieval component.

Scope of the comparison. The two TOME conditions jointly change the retrieval inventory and the use of structural metadata. The experiment therefore establishes the utility of the deployed retrieval configurations, rather than a causal effect for dependency edges alone. Isolating that effect would require holding the candidate set fixed while adding or perturbing expert-validated edges. We likewise avoid making evidence-use claims here because the current table reports correctness only.

G Curriculum-Scaling Protocol

Subject-wise design. The curriculum study in Section 5.2 covers discrete mathematics, mathematical analysis, abstract algebra, probability, differential geometry, and topology. We train a separate model for each subject so that curriculum order is not confounded with the composition of a multidisciplinary corpus. Every condition is evaluated on the same textbook-native exercises from held-out books in the corresponding subject.

Normalized chapter stages. Textbooks contain different numbers of chapters, so raw chapter indices are not directly comparable across books. For an example x from book b, let $i(x)$ be its zero-based chapter index and N_b the number of chapters in the book. We define its normalized position as $p(x) = i(x)/N_b$ and assign it to progress stage $S_{\lfloor 10p(x) \rfloor}$, yielding ten stages $S_0, \ldots, S_9$. Examples from different books are interleaved within each stage, preserving their approximate within-book progression without imposing an arbitrary order over books. Gap QA makes a locally omitted proof obligation explicit; all remaining examples are treated as Concept QA.

Exposure schedules. The four schedules differ only in presentation order:

- Shuffle globally shuffles all examples.
- Forward–Gap traverses stages from S_0 to S_9 and presents Gap before Non-gap examples within each stage.
- Reverse–Gap traverses stages from S_9 to S_0 while retaining the same Gap-first policy.
- Forward–Mix follows the forward stage order but mixes Gap and Non-gap examples within each stage.

Thus, Forward–Gap versus Reverse–Gap is the controlled comparison for chapter direction, and Forward–Gap versus Forward–Mix is the controlled comparison for stage-wise Gap-first ordering. The comparison with Shuffle changes both factors and is interpreted as the combined effect of the structured schedule.

Scales and matched controls. We use per-subject target-token budgets of 0.75M, 1.8M, and 2.4M. Within every subject, both the source-book sets and the QA sets are strictly nested across scales: the 8-book set is contained in the 16-book set, which is contained in the 24-book set, and the corresponding QA sets follow the same nesting. Thus, the three columns in Table 12 are cumulative rather than disjoint. All held-out evaluation books are excluded from every training scale.

Table 12: Cumulative composition of the nested subject-specific training sets used in the curriculum-scaling study. Within each subject and scale, all four schedules use the same books and QA IDs and differ only in presentation order.

	0.75M tokens		1.8M tokens		2.4M tokens	
Subject	Books	QA	Books	QA	Books	QA
Discrete mathematics	8	2,273	16	5,258	24	6,847
Mathematical analysis	8	2,387	16	5,809	24	7,630
Abstract algebra	8	2,235	16	4,899	24	6,831
Probability	8	2,062	16	4,694	24	6,230
Differential geometry	8	1,899	16	4,033	24	5,537
Topology	8	1,930	16	4,877	24	6,571
Total	48	12,786	96	29,570	144	39,646

Target tokens, rather than the number of books or QA pairs, define supervision scale because answer lengths vary across sources and subjects. The realized token totals across the six subjects are 4,499,926, 10,799,935, and 14,399,975 for the three scales, respectively; each subject-level total differs from its nominal target by at most 33 tokens. Within each subject–scale cell, all schedules use identical QA IDs, source books, realized token counts, model initialization, and optimization hyperparameters. Data-loader shuffling, sequence packing, and length-based grouping are disabled so that the prescribed exposure order is preserved. Each subject–scale–schedule cell contains one training run; consequently, small differences are interpreted cautiously even when evaluation problems are shared across conditions.

Matched-budget source breadth. The source-breadth comparison in Section 5.3 reuses the 8-book, 0.75M-token FORWARD–GAP condition as NARROW–DENSE. For BROAD–SPARSE, we take the corresponding nested 16-book pool and downsample its QA records to the realized token count of the 8-book condition, while retaining examples from all 16 books. The matched pair uses the same optimization steps, schedule construction, initialization, and held-out-book evaluation set. Consequently, this comparison doubles source coverage without increasing the training-token budget; it is distinct from comparing the 0.75M- and 1.8M-token checkpoints in the scale study.

H CROSS-SUBJECT TRANSFER PROTOCOL AND RESULTS

Checkpoint selection. The study in Section 5.3 uses one 0.75M-token FORWARD–GAP checkpoint per subject from Section 5.2. Each checkpoint is trained exclusively on data from its corresponding subject; initialization, optimization, and schedule construction are shared across subjects.

Evaluation sets. We evaluate each checkpoint on 50 textbook-native held-out-book exercises in each of the other five subjects. Drawing 16 responses per exercise with the configuration in Table 7 yields 4,000 responses per checkpoint and 24,000 overall. The within-subject diagonal is not evaluated; the corresponding results appear in Section 5.2.

Aggregation. For each ordered source–target pair, pass@16 is computed over 50 exercises. Source and target means average their five off-diagonal scores, and the macro mean weights all 30 pairs equally. Table 13 reports the matrix underlying Figure 5(b).

Table 13: Exact cross-subject pass@16 scores (%) for the 0.75M-token FORWARD–GAP checkpoints. Rows denote training subjects and columns denote held-out evaluation subjects. The diagonal was not evaluated; marginal means exclude it.

Train \ Eval	Alg.	Ana.	Prob.	DG	Disc.	Top.	Mean
Alg.	–	60.0	52.0	26.0	40.0	26.0	40.8
Ana.	28.0	–	46.0	32.0	40.0	36.0	36.4
Prob.	34.0	66.0	–	24.0	42.0	38.0	40.8
DG	36.0	62.0	54.0	–	44.0	34.0	46.0
Disc.	30.0	56.0	48.0	30.0	–	38.0	40.4
Top.	34.0	62.0	54.0	32.0	38.0	–	44.0
Mean	32.4	61.2	50.8	28.8	40.8	34.4	41.4

Scope. Because the matrix contains absolute scores rather than paired base-model differences, it measures cross-subject performance but does not by itself separate transfer strength from target-set difficulty. Likewise, the training-scale grid changes source-book count and target tokens jointly and therefore is not a matched-budget breadth intervention.

Table 14: Complete cross-subject pass@k scores (%) for the six 0.75M-token FORWARD–GAP checkpoints. Each block fixes the training subject and reports performance on the other five held-out subject sets. Every target set contains 50 exercises with 16 sampled responses per exercise.

Train	Eval.	pass@1	pass@2	pass@4	pass@8	pass@16
Trained on Abstract Algebra						
Alg.	Ana.	26.88	36.92	45.42	53.05	60.00
Alg.	Prob.	20.00	27.95	36.09	43.80	52.00
Alg.	DG	8.13	12.03	16.47	20.82	26.00
Alg.	Disc.	15.88	21.50	27.38	33.20	40.00
Alg.	Top.	10.00	15.25	20.06	23.39	26.00
Trained on Mathematical Analysis						
Ana.	Alg.	11.00	15.80	20.10	24.15	28.00
Ana.	Prob.	20.13	27.95	34.63	40.43	46.00
Ana.	DG	8.50	13.13	19.05	25.75	32.00
Ana.	Disc.	17.38	22.62	28.30	34.17	40.00
Ana.	Top.	10.25	16.35	23.92	31.13	36.00
Trained on Probability Theory						
Prob.	Alg.	10.50	16.23	22.37	28.38	34.00
Prob.	Ana.	26.25	34.98	44.34	54.38	66.00
Prob.	DG	10.13	14.58	18.73	22.07	24.00
Prob.	Disc.	16.88	23.17	29.20	35.53	42.00
Prob.	Top.	12.63	19.38	26.09	32.22	38.00
Trained on Differential Geometry						
DG	Alg.	13.25	19.22	25.08	30.56	36.00
DG	Ana.	29.13	38.42	46.22	54.16	62.00
DG	Prob.	24.63	34.03	42.09	48.15	54.00
DG	Disc.	20.88	26.83	32.84	38.46	44.00
DG	Top.	10.88	16.43	22.55	28.94	34.00
Trained on Discrete Mathematics						
Disc.	Alg.	9.38	14.85	20.60	25.46	30.00
Disc.	Ana.	23.63	32.52	40.80	48.50	56.00
Disc.	Prob.	19.50	27.25	34.25	40.73	48.00
Disc.	DG	7.63	11.57	16.50	22.85	30.00
Disc.	Top.	10.88	16.90	23.61	30.99	38.00
Trained on Topology						
Top.	Alg.	11.88	17.53	23.35	28.54	34.00
Top.	Ana.	26.38	36.17	45.75	54.90	62.00
Top.	Prob.	23.25	31.08	38.55	46.46	54.00
Top.	DG	7.38	11.50	17.02	23.83	32.00
Top.	Disc.	16.00	21.02	26.21	32.31	38.00