DAVINCI-MATH: A ROUTING-AWARE DATA PIPELINE FOR END-TO-END MATHEMATICAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Training a strong mathematical reasoning model from a base model requires three stages—mid-training, SFT, and RL—with fundamentally different data needs. Most prior work curates data one stage at a time, so locally optimal choices fail to compose end-to-end; we instead take a **joint multi-stage** view, placing every problem where it is most valuable to the final model. We instantiate this view with DAVINCI-MATH, a unified pipeline built on two pillars: (i) **single-source preparation**—cleaning, deduplication, and filtering shared across all three stages are applied once to a corpus aggregated from 28 public sources, producing a 2.7M-problem pool; and (ii) **stage-aware routing**—each problem is labeled via multi-criteria classification and answer verification, then routed to mid-training, post-training, or filtered out, with post-training further split into SFT and a rule-verifiable RL subset. The pipeline yields **62B** mid-training tokens, **3.8M** SFT trajectories, and **39K** RL prompts. We instantiate the full pipeline at both the 3B and 30B-A3B scales: DAVINCI-MATH-3B reaches **86.6** on AIME 2025 and **75.1** on HMMT Feb 2025, and DAVINCI-MATH-30B-A3B reaches **90.4** average over four benchmarks, setting a new state of the art at each scale. Controlled head-to-head comparisons further show that DAVINCI-MATH provides stronger training data than open math datasets at both the mid-training and SFT stages, and ablation studies confirm the effectiveness of our pipeline design.

1 INTRODUCTION

Training a strong mathematical reasoning model on top of a base model (Yang et al., 2024b; Shao et al., 2024; Guo et al., 2025; Olmo et al., 2025) typically requires a multi-stage training pipeline: **midtraining** adapts the model to the mathematical domain (Wang et al., 2025; Zhou et al., 2025; 2026; Han et al., 2024), **supervised fine-tuning (SFT)** teaches structured solution generation (Moshkov et al., 2025; Guha et al., 2025; Hugging Face, 2025), and **reinforcement learning (RL)** sharpens reasoning against verifiable outcomes (Luo et al., 2025; He et al., 2025a; An et al., 2025). Recent open-source models have closed the gap with top-tier proprietary systems on mathematical reasoning, driven by two parallel data-centric efforts: distilling reasoning trajectories from frontier models for SFT (Moshkov et al., 2025; Du et al., 2025) and curating large pools of verifiable problems for RL (Luo et al., 2025; An et al., 2025). Data is the decisive driver of this progress.

Most prior work optimizes data for a single training stage in isolation. SFT demonstrations (Guha et al., 2025; Ye et al., 2025; Muennighoff et al., 2025), RL prompts (Albalak et al., 2025; He et al., 2025b), midtraining corpora (Paster et al., 2023; Wang et al., 2024)each is curated separately. Yet the model is ultimately trained end-to-end. What is optimal for a single stage need not be optimal for the pipeline as a whole (Akter et al., 2025; Zhang et al., 2025). What is needed, then, is a shift in perspective: from a stage-isolated view to a joint multi-stage view of data curation.

To this end, we present DAVINCI-MATH, a unified multi-stage data pipeline built on two pillars. (i) **Single-source preparation**: cleaning, deduplication, are needed by all three stages alike, so we apply them once to a single corpus aggregated from 28 public sources, producing a 2.7M-problem processed pool. (ii) **Stage-aware routing**: each problem in the pool is labeled through multi-criteria classification and answer verification, then routed to mid-training, post-training, or discard. Within post-training, all problems are used for SFT, and the verifiable hard subset is additionally used

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.

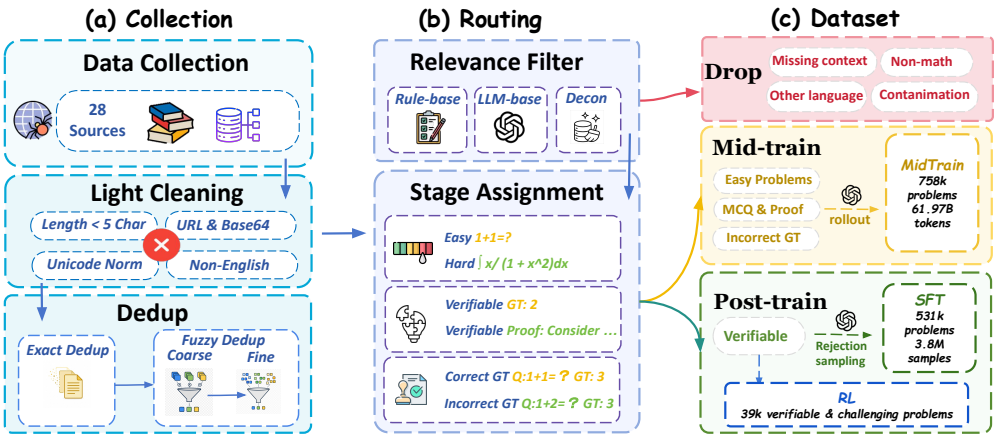


Figure 1: Overview of the DAVINCI-MATH pipeline. Problems aggregated from public sources are deduplicated into a pool, then routed to mid-training, post-training, or drop via rule-based filters and an LLM judge; the post-training pool serves as SFT data, with a verifiable hard subset used for RL.

for RL. The pipeline yields 758K mid-training and 531K post-training problems, producing a final training corpus of 62B mid-training tokens, 3.8M SFT reasoning trajectories, and 39K RL prompts.

To validate the effectiveness of DAVINCI-MATH, we instantiate the full midtraining–SFT–RL pipeline at both the 3B and 30B-A3B scales, on top of daVinci-LLM-3B (Qin et al., 2026) and Qwen3-30B-A3B respectively. DAVINCI-MATH-3B achieves **86.6** on AIME 25, outperforming strong open baselines such as Qwen3.5-4B and Qwen3-4B-Thinking-2507, while DAVINCI-MATH-30B-A3B reaches **90.4** average accuracy, outperforming comparable 30B-A3B-scale open models such as Qwen3.5-35B-A3B and Nemotron-Cascade-2-30B-A3B (§3.2). We then conduct controlled comparisons for both midtraining and SFT, demonstrating that DAVINCI-MATH provides stronger training data than state-of-the-art open-source reasoning datasets at both stages (§3.3). Finally, we ablate the routing design at two levels: that routing problems to the right stage matters at all (§4.1), and that the LLM-based classification and strict answer verification driving these routing decisions are each necessary (§4.2).

We summarize our contributions as follows:

- **A joint multi-stage data pipeline.** We propose DAVINCI-MATH, a unified pipeline that curates data for mid-training, SFT, and RL jointly rather than one stage at a time.
- **An open multi-stage math reasoning dataset.** We release the resulting corpus of **62B** mid-training tokens, **3.8M** SFT trajectories, and **39K** rule-verifiable RL prompts.
- **State-of-the-art reasoning models at both the 3B and 30B-A3B scales.** DAVINCI-MATH-3B reaches **86.6** on AIME 2025 and **75.1** on HMMT Feb 2025, and DAVINCI-MATH-30B-A3B reaches **90.4** average accuracy across four competition benchmarks — each setting a new SOTA at its scale.

2 DAVINCI-MATH DATA PIPELINE

Our pipeline (Figure 1) operates in three phases. We first aggregate problem-solving data from 28 publicly available sources and remove redundancy through deduplication (§2.1). Each remaining problem is labeled along relevance, verifiability, and difficulty by rule-based filters combining with LLM judge, routing to mid-training, post-training, or drop (§2.2). The routed problems are then paired with reasoning trajectories synthesized from GPT-OSS-120B(§2.3). The full pipeline yields 758K mid-training problems and 531K post-training problems, with statistics summarized in §2.4.

2.1 DATA COLLECTION, UNIFICATION, AND DEDUPLICATION

The 28 source datasets (full list in Table 6) follow heterogeneous formats and conventions. We first unify them through a light cleaning pass: Unicode normalization, full-width-to-half-width con-

version, removal of URLs and invisible characters, and filtering of overly short questions. These operations target only surface noise and make no judgment about content.

After cleaning, the raw pool contains heavy redundancy across sources. As existing datasets frequently cross-reference one another, exact character-level deduplication already yields substantial gains. We then apply two rounds of MinHash LSH fuzzy deduplication on n-grams: the first round uses coarser parameters to efficiently prune the bulk of near-duplicates, and the second uses more sensitive parameters to catch the remainder. When two problems are identified as duplicates, we retain the highest-priority copy according to a three-tier ranking: RL datasets (carefully filtered, often with verified answers) first, math-specific SFT corpora second, general-purpose corpora last (full ranking and tie-breaking rule in Table 7). Deduplication hyperparameters are listed in Appendix A.2.

2.2 Classification and Routing

Different training stages impose fundamentally different requirements on data. Mid-training benefits from broad mathematical coverage and diverse reasoning patterns (Akter et al., 2025; Wang et al., 2025); SFT requires high-quality demonstrations with reliable solution trajectories (Ye et al., 2025; Muennighoff et al., 2025); and RL depends on prompts with verifiable outcomes and meaningful difficulty to drive reward signal (An et al., 2025; He et al., 2025b).

Guided by these stage-level requirements, we adopt three routing principles:

- **Drop** problems with no mathematical content or that risk contaminating evaluation.
- **Post-training** (SFT and RL) is reserved for problems are verifiable and challenging.
- **Mid-training** absorbs all remaining content-rich problems.

We implement these principles in two labeling passes: relevance filtering (§2.2.1) identifies problems that should be dropped, and stage assignment (§2.2.2) decides between post-training and mid-training for the rest. Labels are produced by a combination of rule-based filters and an LLM judge, with rules handling reliably detectable patterns and the LLM handling cases that resist simple heuristics. The full post-training pool serves as SFT supervision; a challenging, rule-verifiable subset is further selected as RL prompts (§2.2.3).

2.2.1 Relevance Filtering

This pass identifies problems that are either unrelated to mathematics or risk contaminating held-out evaluation sets. Problems marked here are excluded from training entirely.

Rule-based and LLM-assisted filtering. We first apply a set of rule-based detectors. A problem is retained as genuine math content when its text exhibits mathematical signalsLaTeX math markup (e.g., `\frac`, `\sum`, `\int`), or math-specific instruction verbs (*prove*, *evaluate*, *compute*) paired with math objects (*polynomial*, *triangle*, *prime*) or numeric content. Problems are dropped if they fail this signal check, are written in languages other than English or Chinese, contain broken external references (links, images, or tables), or fall under non-mathematical task types such as translation, free-form writing, and code generation. A small language model then performs a second pass over the survivors to catch edge cases that rules alone miss. The full list of trigger patterns appears in Appendix A, and Table 1 reports the volume of problems assigned each drop label.

Benchmark decontamination. To prevent test-set leakage, we collect problems from 13 evaluation benchmarks: MATH500, OlympiadBench, IMO-AnswerBench, Minerva, BeyondAIME, AMC 2023, AIME 2024, AIME 2025, AIME 2026, HMMT-Feb 2025, HMMT-Nov 2025, HMMT-Feb 2026, and IMO 2025. For each benchmark problem, we retrieve the 20 nearest training problems via embedding-based search and pass them to an LLM judge that decides whether each retrieved problem is the same question as the benchmark one. Appendix A.4 details the embedding model, retrieval index, and judge configuration.

2.2.2 Stage Assignment

This pass identifies problems that carry genuine mathematical value but are incompatible with our post-training verification protocol. Rather than discarding them, we route them to mid-training,

where their content still contributes to the model's mathematical fluency. The full list of mid-training routing tags, the verification cascade (rule-based equivalence followed by an LLM judge), and the per-sample SFT supervision protocol are detailed in Appendix A.5.

Low-challenge problems. Some problems are mathematically valid but lack sufficient challenge for effective post-training. Multiple-choice questions allow a model to exploit answer options through elimination rather than genuine reasoning; we attempt to convert them into open-ended form via template-based transformations, routing those that resist conversion to mid-training. Answer leakage—problems that embed the ground-truth or solution steps directly in the question text—similarly reduces difficulty, and such problems are routed to mid-training as well. Both types retain mathematical content valuable for mid-training, even when unsuitable for post-training supervision.

Hard-to-Verify problems. Both SFT and RL in the post-training stage require verifiable answers: without them, SFT learns from incorrect trajectories and RL cannot compute reliable rewards. Introducing unverifiable problems into post-training therefore injects noise that is more damaging here than in mid-training. A large share of mathematical content—proofs, derivations, and open-ended conceptual questions—resists reduction to a single comparable value. We route all such problems to mid-training, where extended reasoning chains remain valuable even without a verifiable endpoint.

Ground-truth correctness. Problems that carry no label from the two preceding quality checks are candidates for post-training. For each such problem, we draw $n = 8$ rollouts from GPT-OSS-120B and take the majority answer—the most common answer across the 8 rollouts under mathematical equivalence. The majority answer is then compared against the ground-truth in a two-stage cascade: a rule-based verifier handles surface-form matches, and an LLM judge is invoked only when the rule-based check fails, to recover cases such as equivalent expressions in different formats (e.g., $\frac{5}{3}$ vs. $1\frac{2}{3}$). A problem enters the post-training pool when either stage confirms the match; otherwise it is routed to mid-training.

2.2.3 RL Subset Selection

RL prompts are drawn from the post-training pool under two additional constraints: (i) the prompt must yield learning signal—we run a second $n = 8$ rollout pass with an intermediate DAVINCI-MATH checkpoint and drop problems with pass ratio 1.0; and (ii) the reward must come from a rule- or code-based verifier rather than a reward model; problems with non-verifiable answer formats are excluded. The remaining subset forms the RL prompt set, while SFT uses the full post-training pool.

2.3 Reasoning Trajectory Synthesis

We pair each routed problem with reasoning trajectories from GPT-OSS-120B. For post-training problems, we reuse the 8 rollouts produced during answer verification (§2.2.2); only trajectories whose final answer matches the verified ground-truth enter the SFT pool, so supervision is grounded in verified solutions. For mid-training problems, we draw $n = 8$ fresh rollouts and keep every one that fits within the context-length budget regardless of answer correctness, since broad coverage of reasoning patterns matters more here than answer accuracy.

In addition to GPT-OSS-120B, we synthesize a second set of trajectories for the same problem pool using DeepSeek-V4-Flash (DeepSeek-AI, 2026) as an alternative CoT teacher model; this DeepSeek-V4-Flash variant is used for the 30B-A3B scaling results reported in §3.2.

2.4 Final Statistics

Table 1 traces the full pipeline from raw input to the final training corpora. We aggregate 42.79M mathematical problems from 28 publicly available sources; a single pass of exact deduplication leaves 10.71M problems, and two subsequent rounds of n-gram deduplication leave 2.71M canonical problems. Classification, routing, and answer verification then partition the canonical pool into 1.42M discarded problems, 758K mid-training problems, and 531K post-training problems. After trajectory synthesis (§2.3), the routed problems yield 62B mid-training tokens, 3.8M SFT samples, and 39K RL prompts. Complete per-source statistics—curation breakdown (Table 15), filter-label inventory (Table 17), and rollout statistics (Table 16)—are provided in Appendix C.

Table 1: DAVINCI-MATH pipeline funnel from raw input to final routing destinations. "Share of canonical" is the percentage of the 2.71M canonical pool—i.e., problems remaining after exact and two-round fuzzy deduplication—routed to each bucket.

Stage / Bucket	Problems	Share of canonical
Raw (28 sources)	42,794,402	—
Collection and deduplication:		
+ Exact dedup	10,713,392	—
+ Two-round fuzzy dedup	2,710,101	100.0%
Classification, routing, and verification:		
`drop`	1,421,116	52.4%
`midtrain`	757,783	28.0%
`posttrain` (used for SFT)	531,202	19.6%
↪ challenging subset (used for RL)	39,106	1.4%

3 EXPERIMENTS

In this section, we first describe the experimental setup (§3.1) and then evaluate DAVINCI-MATH from three complementary perspectives: (1) its end-to-end performance on a transparent base model, along with stage-specific analyses of midtraining and SFT(§3.2), (2) detailed controlled comparisons with representative open datasets(§3.3), and (3) the strength of its full midtraining+SFT recipe against other datasets under the same two-stage setup (§3.3). We further evaluate its out-of-distribution generalization ability in §3.4.

3.1 SETUP

Base Models Most of the experiments are conducted on top of DAVINCI-LLM-3B (Qin et al., 2026), which serves as our primary base model because its training pipeline is fully transparent and does not involve additional math-specific adaptation. To test whether our pipeline scales beyond 3B, we additionally instantiate the full recipe on Qwen3-30B-A3B-Base (Yang et al., 2025) (§3.2).

Training Paradigm The midtraining split of DAVINCI-MATH contributes the problem-solving (QA) portion of midtraining. To isolate this QA component while reflecting standard practice, all midtraining runs mix a *fixed* document corpus—MegaMath-Web-Pro-Max (Zhou et al., 2025), UltraData-Math (Zhou et al., 2026), Nemotron-CC-Math (Mahabadi et al., 2025)—with a *variable* QA corpus drawn either from DAVINCI-MATH's midtraining split or from open alternatives. All SFT experiments share the same hyperparameters (Appendix B.2).

Baselines We assess DAVINCI-MATH along two axes: the strength of the final model, comparing DAVINCI-MATH-3B against open reasoning models of comparable or larger scale—Qwen3.5-4B, Qwen3-4B-Thinking-2507, Olmo-3-7B-Think, R1-Qwen3-8B, Qwen3-30B-A3B, and Qwen3-32B (Olmo et al., 2025; Yang et al., 2025)—and the quality of the data itself, comparing against representative open math datasets—OpenMathReasoning (OMR), Nemotron-Math (NemoMath), and the Math-CoT portion of Nemotron-Cascade-2 (NemoCas) (Moshkov et al., 2025; Du et al., 2025; Yang et al., 2026)—at the midtraining stage, the SFT stage, and as an end-to-end recipe (Appendix B.3). For the 30B-A3B-scale comparison, we additionally compare against Qwen3.5-35B-A3B, Nemotron-Cascade-2-30B-A3B, NVIDIA-Nemotron-3-Nano-30B-A3B (NVIDIA, 2025), and Qwen3-30B-A3B-Thinking-2507.

Evaluation Protocol Benchmarks include AIME 25, AIME 26, HMMT Feb 2025, HMMT Nov 2025, MATH-500 for in-domain math reasoning (Hendrycks et al., 2021; Balunović et al., 2025), and GPQA-Diamond, SciBench, PIQA, FinQA (Rein et al., 2023; Wang et al., 2023; Bisk et al., 2020; Chen et al., 2021) for OOD evaluation (§3.4). The main model comparison in §3.2 uses an 81,920-token context; controlled data comparisons (midtraining and SFT) use 30,720 tokens. Sampling budgets, decoding configurations, and the verifier are summarized in Appendix B.4.

Table 2: Results on mathematical reasoning benchmarks. DAVINCI-MATH-3B achieves the best average score against strong open-source baselines (top block); the lower two blocks report controlled midtraining and SFT comparisons on the same DAVINCI-LLM-3B base model.

Setting	AIME 25	AIME 26	HMMT Feb	HMMT Nov	MATH-500	AVG
Open Reasoning Models						
Qwen3.5-4B	79.90	82.81	69.58	**71.15**	94.80	79.65
Qwen3-4B-Thinking-2507	79.58	80.52	54.69	67.50	95.20	75.50
Olmo-3-7B-Think	69.27	73.75	48.85	53.44	94.60	67.98
R1-Qwen3-8B	74.48	76.35	58.54	66.98	95.00	74.27
Qwen3-30B-A3B	71.77	72.81	50.20	60.94	95.00	70.15
Qwen3-32B	71.67	73.23	50.52	59.48	94.80	69.94
Midtraining Data Comparison						
daVinci-MT-Nemo-OMR	36.25	40.52	26.77	26.98	87.80	43.66
daVinci-MT-daVinciMath	41.98	48.13	29.27	31.67	87.20	47.65
SFT Data Comparison						
daVinci-SFT-OMR	48.44	56.15	33.85	34.79	92.20	53.09
daVinci-SFT-NemoMath	25.83	27.92	20.10	18.02	78.20	34.02
daVinci-SFT-NemoCas	63.54	63.44	57.29	52.81	92.60	65.94
daVinci-SFT-daVinciMath	71.04	75.42	57.92	56.77	93.60	70.95
Final model trained on daVinciMath						
DAVINCI-MATH-3B	**86.56**	**85.94**	**75.10**	69.38	**95.20**	**82.44**

Table 3: Comparison of DAVINCI-MATH-30B-A3B with open 30B-A3B-scale reasoning models.

Model	AIME 25	AIME 26	HMMT Feb	HMMT Nov	Avg.
Qwen3-30B-A3B-Thinking-2507	84.90	86.98	70.63	78.02	80.13
NVIDIA-Nemotron-3-Nano-30B-A3B	87.81	91.04	83.65	80.94	85.86
Nemotron-Cascade-2-30B-A3B	91.25	87.50	88.44	83.13	87.58
Qwen3.5-35B-A3B	**91.88**	90.73	86.77	88.02	89.35
DAVINCI-MATH-30B-A3B	91.04	**91.88**	**89.27**	**89.48**	**90.42**

3.2 MAIN RESULTS

For the final DAVINCI-MATH-3B and DAVINCI-MATH-30B-A3B models, we instantiate the full three-stage training pipeline—midtraining on the midtraining split of DAVINCI-MATH, supervised fine-tuning on its curated posttrain split, and reinforcement learning with GRPO on its verifiable RL subset—on top of DAVINCI-LLM-3B and Qwen3-30B-A3B respectively, with DAVINCI-MATH-30B-A3B using the DeepSeek-V4-Flash trajectory variant (§2.3) in place of GPT-OSS-120B. Full hyperparameter details for all three stages are provided in Appendix B.2.

As shown in Table 2, compared with the open-source reasoning models in the top block, the full DAVINCI-MATH-3B model achieves the highest average score. It surpasses Qwen3.5-4B by 2.79 points on average across the full evaluation suite and also outperforms substantially larger open models such as Qwen3-30B-A3B and Qwen3-32B. DAVINCI-MATH-30B-A3B reaches **90.42** average accuracy across the four benchmarks in Table 3, outperforming comparable 30B-A3B-class open models including Qwen3.5-35B-A3B (89.35) and Nemotron-Cascade-2-30B-A3B (87.58).

Beyond the final model results, the controlled stage-wise comparisons further support the effectiveness of DAVINCI-MATH. Because we evaluate end-to-end performance rather than the midtrained checkpoint alone, each midtraining run is followed by a fixed SFT stage on a 100K-example NemotronMath subset, and the reported accuracy reflects the full midtrain $\rightarrow$ SFT pipeline. Under this protocol, using DAVINCI-MATH as the problem-solving component leads to stronger downstream post-training performance than the open alternative. In the SFT comparison, DAVINCI-MATH outperforms representative open mathematical reasoning datasets, including OpenMathReasoning, NemotronMath, and the Math-CoT portion of Nemotron-Cascade-2. Together, these results show that DAVINCI-MATH is effective both as a complete end-to-end recipe and as stage-specific training data for midtraining and supervised fine-tuning.A per-stage breakdown of how mid-training, SFT, and RL each contribute to the final DAVINCI-MATH-3B score is reported in Appendix D.2.

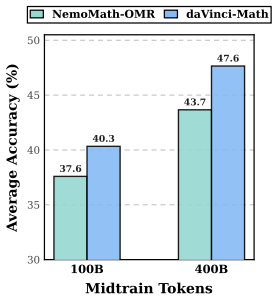


Figure 2: DAVINCI-MATH outperforms the open-data baseline across various midtrain scales.

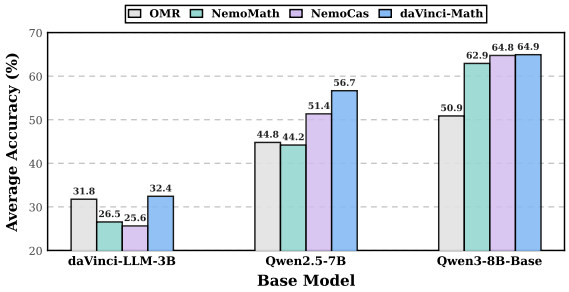


Figure 3: Matched-budget SFT comparison across base models. Models trained on DAVINCI-MATH achieve the best average performance on all three base models, demonstrating consistent transfer across model families.

3.3 CONTROLLED COMPARE WITH OPEN DATASETS

In this section, we perform controlled studies to systematically analyze the contributions of DAVINCI-MATH at different stages of the training pipeline, including midtraining, SFT, and their composition as an end-to-end data recipe.

daVinci-Math Provides Stronger Midtraining Data than Open Alternatives To assess the quality of DAVINCI-MATH as a midtraining resource, we compare it with representative open alternatives under a controlled setup. Each midtraining run is evaluated under the same midtrain $\rightarrow$ SFT protocol as in §3.2. Across all runs, the math-corpus component is kept fixed and only the problem-solving data in midtraining varies between DAVINCI-MATH and an open baseline built from a mix of NemotronMath and OpenMathReasoning, denoted as NemoMATH-OMR.

Both the setting with 100B midtraining tokens and the setting with 400B tokens are reported. As summarized in Figure 2, DAVINCI-MATH delivers stronger downstream performance than the open-data baseline in both settings. DAVINCI-MATH improves average performance from 37.59 to 40.33 at 100B tokens and from 43.66 to 47.65 at 400B tokens. Detailed results are provided in Table 18.

daVinci-Math Provides Stronger SFT Data across Base Models To assess the quality of DAVINCI-MATH as an SFT resource, we compare posttrain spilt of DAVINCI-MATH against representative open datasets. We evaluate transfer across base models daVinci-LLM-3B, Qwen2.5-7B (Yang et al., 2024a), and Qwen3-8B-Base (Yang et al., 2025), while matching the SFT token budget across datasets for a fair comparison ($\sim$1.892B tokens), which corresponds to a 300K-sample subset of daVinci-Math's SFT split. Detailed dataset information is provided in Table 12.

As shown in Figure 3, DAVINCI-MATH achieves the best average performance on every base model, reaching 32.43 on daVinci-LLM-3B, 56.66 on Qwen2.5-7B, and 64.93 on Qwen3-8B-Base. Averaging over the three base models, DAVINCI-MATH attains an overall score of 51.34, outperforming OpenMathReasoning (42.47), NemotronMath (44.55), and Nemotron-Cascade-2 (47.24). These results show that the advantage of DAVINCI-MATH transfers across model families even under a matched small-scale SFT budget. Detailed results are provided in Table 19.

daVinci-Math Forms the Strongest End-to-End Data Recipe To test whether the gains from the two stages compose into a stronger end-to-end recipe, we compare all combinations of two midtraining sources and four SFT datasets on top of DAVINCI-LLM-3B. As shown in Figure 4, the strongest result is obtained when both stages use DAVINCI-MATH, reaching 47.89 average accuracy. Holding the SFT dataset fixed, the DAVINCI-MATH midtrained checkpoint outperforms the NemoMath-OMR checkpoint in all four cases; holding the midtraining checkpoint fixed, DAVINCI-MATH is also the strongest SFT dataset in both rows. Detailed results are provided in Table 20.

3.4 OOD GENERALIZATION OF DAVINCI-MATH

To test whether the advantage of DAVINCI-MATH transfers beyond in-domain math benchmarks, we further evaluate models trained with different full split SFT datasets on a broader set of reasoning-domain tasks, including GPQA-D, SciBench, PIQA, and FinQA. As shown in Figure 5, the model

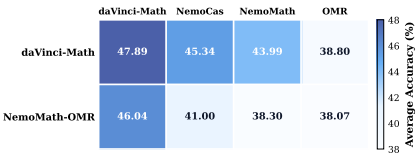


Figure 4: Combined midtraining and SFT comparison on DAVINCI-LLM-3B. Rows denote the midtraining data, columns denote the SFT dataset, and each cell reports the average accuracy across the five math benchmarks.

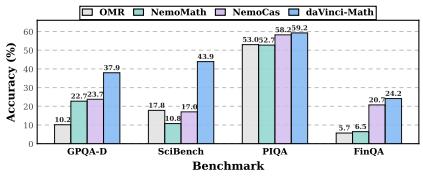


Figure 5: OOD generalization on four out-of-domain reasoning benchmarks. Models trained on DAVINCI-MATH consistently outperform alternative SFT datasets on all four benchmarks.

trained on DAVINCI-MATH achieves the best result on all four generalize reasoning benchmarks, reaching 37.94 on GPQA-D, 43.91 on SciBench, 59.25 on PIQA, and 24.16 on FinQA. These results show that DAVINCI-MATH yields the strongest generalization to out-of-domain reasoning tasks among the compared datasets.

4 ABLATION STUDIES

We ablate DAVINCI-MATH's routing system in two steps. We first ask whether routing matters at all—whether which stage receives which problem actually affects final performance (§4.1). We then ask whether the two key labeling components driving these routing decisionsLLM-based relevance classification and strict answer verificationare each necessary. (§4.2).

4.1 VALIDATING THE ROUTING ASSIGNMENT

The ablations in this section and the next adopt a reduced-scale protocol: starting from DAVINCI-LLM-3B, we midtrain on 100B tokens of candidate midtraining data and SFT on candidate post-training data under a matched ∼1.89B-token budget—the size of the 300K-sample DAVINCI-MATH SFT pool—with only the data assignment varied, across three independent training seeds. The full DAVINCI-MATH pipeline serves as the baseline (Table 4, top row); full hyperparameters are in Appendix B.2, and the complete four-benchmark breakdown is in Appendix D.2.

Table 4: Factorized routing ablations under the reduced-scale protocol, averaged over three seeds.

Setting	AIME 25	HMMT Feb
DAVINCI-MATH	**36.15**	**24.69**
w/o low-challenge routing	33.23	21.98
distribution-matched random routing	32.40	22.29
w/o routing	29.69	22.71
w/o hard-to-verify routing	31.25	20.73
swap midtrain ↔ SFT	32.81	19.69

Within this protocol, five ablations probe the routing design: *w/o routing* pools all surviving problems and samples both midtraining and SFT at random; *swap* reverses the routing partition; *w/o low-challenge routing* and *w/o hard-to-verify routing* respectively admit MCQ/answer-leakage and proof-style/open-ended problems into SFT instead of midtraining; and *distribution-matched random routing* assigns problems at random while matching the source, length, and label distribution of the true partition, controlling for the possibility that any partition would help. As shown in Table 4, full routing wins on both benchmarks and still outperforms distribution-matched random routing, showing the gain comes from the specific criteria rather than partitioning alone—consistent with prior findings that broad data suits midtraining while curated data suits post-training (Akter et al., 2025).

4.2 VALIDATING THE PIPELINE DESIGN

Two operations distinguish our labeling pipeline from prior efforts in SFT data construction: relevance filtering combines rule-based heuristics with an LLM judge, and post-training answers undergo strict verification. We ablate each component under the same reduced-scale setup (§4.1), but here we hold the baseline midtraining checkpoint fixed and vary only the SFT pool.

Without LLM-based classification. We restrict relevance filtering to the rule-based heuristics, removing the LLM judge that catches semantically non-mathematical content, broken/missing context, and other edge cases that surface rules miss. Without it, problems that the full pipeline would drop or route to midtraining instead leak into the post-training pool. We then sample SFT data from this contaminated pool under the baseline's token budget. The result falls below the baseline (Table 5),

indicating that finer-grained classification removes a meaningful share of the noise that rules alone would admit into supervision.

Without answer verification. Many SFT construction efforts skip answer filtering or accept the distillation model's majority answer as ground-truth. We instead require the ground-truth and rollout consensus to agree before a trajectory enters the SFT pool, so every learned trajectory is verifiably correct. Dropping this verification step—admitting every candidate regardless of whether its answer is confirmed—drops performance well below the baseline (Table 5): enough wrong or unverifiable answers enter supervision to degrade it, confirming that strict verification is worth its compute cost.

Table 5: Pipeline-component ablations under the reduced-scale protocol, averaged over three seeds.

Setting	AIME 25	HMMT Feb
DAVINCI-MATH	**36.15**	**24.69**
w/o LLM classification	32.60	22.29
w/o answer verification	31.46	20.00

5 Related Work

Data Engineering for Mathematical Reasoning Mathematical data has long been central to improving language models' reasoning ability. Early efforts emphasized broad-coverage corpora—OpenWebMath, MathPile, MegaMath, Nemotron-CC-Math—collecting math-rich web, textbook, and code content to expose models to diverse mathematical knowledge Paster et al. (2023); Wang et al. (2024); Zhou et al. (2025); Mahabadi et al. (2025). Problem-solving datasets such as OpenMathInstruct, OpenMathReasoning, OpenThoughts, and NemotronMath instead pair challenging problems with solutions or reasoning traces, often refined through filtering, curation, or distillation from stronger teachers Toshniwal et al. (2024); Moshkov et al. (2025); Du et al. (2025); Guha et al. (2025).

Multi-Stage Training for Reasoning Models Post-training is now central to eliciting reasoning ability in pretrained language models: SFT teaches step-by-step solutions from high-quality demonstrations, while RL optimizes against reliable or verifiable reward signals Toshniwal et al. (2024); Moshkov et al. (2025); Du et al. (2025); Hugging Face (2025); He et al. (2025a); Yu et al. (2025). Mid-training has likewise become an important intermediate adaptation stage before post-training, and recent work studies its relationship to later stages: Akter et al. (2025) study pretraining–post-training synergy, Zhang et al. (2025) analyze the pre-training/mid-training/RL interplay, and Wang et al. (2025) show mid-training improves subsequent RL scaling.

Our work complements this training-centric view from the data perspective: rather than scheduling mid-training, SFT, and RL, we study how heterogeneous problem-solving data should be assigned across these stages.

6 Conclusion

We presented DAVINCI-MATH, a routing-aware data pipeline taking an end-to-end view of mathematical reasoning training. DAVINCI-MATH-3B, trained through the full midtraining–SFT–RL stack, reaches 82.44 average accuracy across five benchmarks and surpasses substantially larger open baselines, validating our pipeline's effectiveness.

AI use statement

We used generative AI tools to assist with parts of the code development for our data-construction pipeline (e.g., deduplication, classification, and verification scripts), and reviewed all AI-assisted code for correctness.

Ethics statement

Not applicable. This work does not involve human subjects, and we do not identify concerns related to dataset licensing, privacy, dual-use risk, or other topics covered by the ICLR Code of Ethics.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide the exact prompts used for data collection, classification, and trajectory synthesis in Appendix A.6, which together with the filtering and routing rules described in §2.2 fully specify our data-construction pipeline and support dataset reproduction. Complete training hyperparameters for the midtraining, SFT, and RL stages are provided in Appendix B.2, and the evaluation protocol—benchmarks, sampling budgets, decoding configuration, and verifier—is detailed in Appendix B.4, together supporting reproduction of our experimental results.

REFERENCES

Syeda Nahida Akter, Shrimai Prabhumoye, Eric Nyberg, Mostofa Patwary, Mohammad Shoeybi, Yejin Choi, and Bryan Catanzaro. Front-loading reasoning: The synergy between pretraining and post-training data, 2025. URL https://arxiv.org/abs/2510.03264.

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

Chenxin An, Zhihui Xie, Xiaonan Li, Lei Li, Jun Zhang, Shansan Gong, Ming Zhong, Jingjing Xu, Xipeng Qiu, Mingxuan Wang, and Lingpeng Kong. Polaris: A post-training recipe for scaling reinforcement learning on advanced reasoning models, 2025. URL https://hkunlp.github.io/blog/2025/Polaris.

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. Matharena: Evaluating llms on uncontaminated math competitions. *arXiv preprint arXiv:2505.23281*, 2025.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan R Routledge, et al. Finqa: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3697–3711, 2021.

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

Wei Du, Shubham Toshniwal, Branislav Kisacanin, Sadegh Mahdavi, Ivan Moshkov, George Armstrong, Stephen Ge, Edgar Minasyan, Feng Chen, and Igor Gitman. Nemotron-math: Efficient long-context distillation of mathematical reasoning from multi-mode supervision. *arXiv preprint arXiv:2512.15489*, 2025.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. Openthoughts: Data recipes for reasoning models. *arXiv preprint arXiv:2506.04178*, 2025.

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

Xiaotian Han, Yiren Jian, Xuefeng Hu, Haogeng Liu, Yiqi Wang, Qihang Fan, Yuang Ai, Huaibo Huang, Ran He, Zhenheng Yang, et al. Infimm-webmath-40b: Advancing multimodal pretraining for enhanced mathematical reasoning. *arXiv preprint arXiv:2409.12568*, 4, 2024.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, et al. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*, 2025a.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025b.

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.

Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL https://github.com/huggingface/open-r1.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025a.

Jiacai Liu, Yingru Li, Yuqian Fu, Jiawei Wang, Qian Liu, and Zhuo Jiang. When speed kills stability: Demystifying RL collapse from the training-inference mismatch, September 2025b. URL https://richardli.xyz/rl-collapse.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025c.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005b 2025. Notion Blog.

Rabeeh Karimi Mahabadi, Sanjeev Satheesh, Shrimai Prabhumoye, Mostofa Patwary, Mohammad Shoeybi, and Bryan Catanzaro. Nemotron-cc-math: A 133 billion-token-scale high quality math pretraining dataset. *arXiv preprint arXiv:2508.15096*, 2025.

Ivan Moshkov, Darragh Hanley, Ivan Sorokin, Shubham Toshniwal, Christof Henkel, Benedikt Schifferer, Wei Du, and Igor Gitman. Aimo-2 winning solution: Building state-of-the-art mathematical reasoning models with openmathreasoning dataset. *arXiv preprint arXiv:2504.16891*, 2025.

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

NVIDIA. Nemotron 3 Nano: Open, efficient mixture-of-experts hybrid Mamba-Transformer model for Agentic reasoning, 2025. URL https://arxiv.org/abs/2512.20848. Technical report.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, 2025. URL https://arxiv.org/abs/2512.13961.

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.

Yiwei Qin, Yixiu Liu, Tiantian Mi, Muhang Xie, Zhen Huang, Weiye Si, Pengrui Lu, Siyuan Feng, Xia Wu, Liming Liu, et al. davinci-llm: Towards the science of pretraining. *arXiv preprint arXiv:2603.27164*, 2026.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

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*, 2024.

Xiaoxuan Wang, Ziniu Hu, Pan Lu, Yanqiao Zhu, Jieyu Zhang, Satyen Subramaniam, Arjun R Loomba, Shichang Zhang, Yizhou Sun, and Wei Wang. Scibench: Evaluating college-level scientific problem-solving abilities of large language models. *arXiv preprint arXiv:2307.10635*, 2023.

Zengzhi Wang, Xuefeng Li, Rui Xia, and Pengfei Liu. Mathpile: A billion-token-scale pretraining corpus for math. *Advances in Neural Information Processing Systems*, 37:25426–25468, 2024.

Zengzhi Wang, Fan Zhou, Xuefeng Li, and Pengfei Liu. Octothinker: Mid-training incentivizes reinforcement learning scaling. *arXiv preprint arXiv:2506.20512*, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024a.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024b.

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

Zhuolin Yang, Zihan Liu, Yang Chen, Wenliang Dai, Boxin Wang, Sheng-Chieh Lin, Chankyu Lee, Yangyi Chen, Dongfu Jiang, Jiafan He, et al. Nemotron-cascade 2: Post-training llms with cascade rl and multi-domain on-policy distillation. *arXiv preprint arXiv:2603.19220*, 2026.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning. *arXiv preprint arXiv:2502.03387*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Charlie Zhang, Graham Neubig, and Xiang Yue. On the interplay of pre-training, mid-training, and rl on reasoning language models. *arXiv preprint arXiv:2512.07783*, 2025.

Chuyue Zhou, Hongya Lyu, Xinle Lin, Hengyu Zhao, Junshao Guo, Xueren Zhang, Shuaikang Xue, Qiang Ma, Jie Zhou, Yudong Wang, and Zhiyuan Liu. Ultradata-math, 2026. URL `https://huggingface.co/datasets/openbmb/UltraData-Math`.

Fan Zhou, Zengzhi Wang, Nikhil Ranjan, Zhoujun Cheng, Liping Tang, Guowei He, Zhengzhong Liu, and Eric P Xing. Megamath: Pushing the limits of open math corpora. *arXiv preprint arXiv:2504.02807*, 2025.

Zilin Zhu, Chengxing Xie, Xin Lv, and slime Contributors. slime: An llm post-training framework for rl scaling. https://github.com/THUDM/slime, 2025. GitHub repository. Corresponding author: Xin Lv.

Appendix Contents

A Data Pipeline Details
- A.1 Data Sources
- A.2 Deduplication Hyperparameters
- A.3 Relevance Filtering
- A.4 Benchmark Decontamination Setup
- A.5 Stage Assignment
- A.6 Prompts

B Training and Evaluation Details
- B.1 Training Infrastructure
- B.2 Training Hyper-parameters
- B.3 Training Data
- B.4 Evaluation Parameters

C Detailed Dataset Statistics
- C.1 Per-source curation breakdown
- C.2 Per-source rollout statistics
- C.3 Routing-label and verification statistics

D Full Evaluation Results
- D.1 Full main results
- D.2 Full ablation results

E Additional Robustness Analyses
- E.1 Mid-Training Topic Diversity
- E.2 Controlled Comparisons with Standard Errors
- E.3 Output-Token-Length Comparisons
- E.4 Paired Bootstrap Confidence Interval
- E.5 Evaluation-Protocol Robustness Check

A Data Pipeline Details

A.1 Data Sources

We aggregate problem-solving data from 28 publicly available mathematical sources, totaling 42,794,402 raw records before any cleaning or deduplication. Provenance and licenses for each source are listed in Table 6; per-source curation statistics (raw size, deduplication, and routing destinations) are reported separately in Appendix C.

License and attribution. All 28 upstream sources are publicly available on the HuggingFace Hub; their canonical identifiers and declared licenses are reported in Table 6. The most restrictive upstream license among our sources is CC-BY-NC-2.0 (*stepfun_sft*), and we therefore release the derived DAVINCI-MATH dataset under **CC-BY-NC-4.0** for research use, which honors the non-commercial constraint while remaining compatible with the permissive (Apache-2.0, MIT, CC-BY-4.0, ODC-BY) and NVIDIA Open Model License terms of the other sources. Where upstream sources require share-alike or specific attribution—notably the Llama-Nemotron-Post-Training-Dataset and Nemotron-Cascade-2 families—we provide attribution in the released dataset card and refer users to the original sources.

Table 6: Provenance and license for each upstream source. The HuggingFace identifier (clickable) maps each name in Table 15 to its canonical release on the HuggingFace Hub. Licenses are taken from the dataset's HuggingFace metadata; DAVINCI-MATH is itself released under CC-BY-NC-4.0 (see Appendix A.1).

Dataset	HuggingFace identifier	License
Ring-lite-sft	`inclusionAI/Ring-lite-sft-data`	Apache-2.0
dolphin-r1	`cognitivecomputations/dolphin-r1`	Apache-2.0
stepfun_sft	`stepfun-ai/Step-3.5-Flash-SFT`	CC-BY-NC-2.0[†]
Nemotron-Math-v2	`nvidia/Nemotron-Math-v2`	CC-BY-4.0
numinamath	`AI-MO/NuminaMath-CoT`	Apache-2.0
OpenMathReasoning	`nvidia/OpenMathReasoning`	CC-BY-4.0
Llama-Nemotron-math_v1	`nvidia/Llama-Nemotron-Post-Training-Dataset-v1`	CC-BY-4.0[‡]
Nemotron-Cascade-2-SFT-Data	`nvidia/Nemotron-Cascade-2-SFT-Data`	NVIDIA OML
OpenR1-math	`open-r1/OpenR1-Math-220k`	Apache-2.0
DeepMath-103K	`zwhe99/DeepMath-103K`	MIT
Llama-Nemotron-math_v1.1	`nvidia/Llama-Nemotron-Post-Training-Dataset-v1.1`	CC-BY-4.0[‡]
OpenThoughts-114k	`open-thoughts/OpenThoughts-114k`	Apache-2.0
OpenThoughts-3-1.2M	`open-thoughts/OpenThoughts3-1.2M`	Apache-2.0
Skywork-OR1-RL-Data	`Skywork/Skywork-OR1-RL-Data`	Apache-2.0
SYNTHETIC-2	`PrimeIntellect/SYNTHETIC-2-SFT-verified`	Apache-2.0
openreasoning_57k_72k	`Open-Reasoner-Zero/orz_math_72k_collection_extended`	MIT
MiroMind-m1-62k	`miromind-ai/MiroMind-M1-RL-62K`	Apache-2.0
polaris-data-53K	`POLARIS-Project/Polaris-Dataset-53K`	Apache-2.0
Dolci-Instruct-RL	`allenai/Dolci-Instruct-RL`	ODC-BY
deepscaler	`agentica-org/DeepScaleR-Preview-Dataset`	MIT
dapo-math-17k	`BytedTsinghua-SIA/DAPO-Math-17k`	Apache-2.0
STILL-3-RL	`RUC-AIBOX/STILL-3-Preview-RL-Data`	MIT
openreasoning_13k_hard	`Open-Reasoner-Zero/orz_math_13k_collection_hard`	MIT
Dolci-Think-RL-7B	`allenai/Dolci-Think-RL-7B`	ODC-BY
omega	`allenai/omega-explorative`	MIT
Dolci-Think-RL-32B	`allenai/Dolci-Think-RL-32B`	ODC-BY
Dolci-RL-Zero-Math-7B	`allenai/Dolci-RL-Zero-Math-7B`	ODC-BY
KlearReasoner	`Kwai-Klear/KlearReasoner-MathSub-30K`	Apache-2.0

[†] Dual-licensed Apache-2.0 + CC-BY-NC-2.0; the non-commercial constraint is binding and propagates to DAVINCI-MATH. [‡] Predominantly CC-BY-4.0; contains ODC-BY and CC-BY-SA-4.0 sub-collections that we honor via attribution.

Source priority. Tier definitions and the survivor-selection rule used during fuzzy deduplication are summarised in Table 7. Within each connected component the survivor is taken in ascending tier

Tier	Source	Raw rows	After dedup
Tier 1 *RL-grade*	DeepMath-103K	103,022	51,895
	MiroMind-m1-62k	62,118	16,810
	Skywork-OR1-RL-Data	105,055	23,191
	OpenR1-math	450,258	66,660
	Dolci-Instruct-RL	63,998	6,441
	Dolci-RL-Zero-Math-7B	13,314	1,346
	Dolci-Think-RL-7B	30,180	2,077
	Dolci-Think-RL-32B	30,182	2,015
	dapo-math-17k	1,791,700	3,972
	DeepScaleR	40,315	4,435
	polaris-data-53K	341,373	9,615
	STILL-3-RL	29,925	3,097
	openreasoning_zero_13k_hard	13,451	2,102
	openreasoning_zero_57k_72k	129,322	18,347
	Tier-1 subtotal	3,204,213	212,003
Tier 2 *Math-specific SFT*	Llama-Nemotron-math_v1	19,840,970	133,541
	Llama-Nemotron-math_v1.1	2,225,427	39,077
	OpenMathReasoning	5,678,317	154,864
	Nemotron-Math-v2	324,185	200,910
	Nemotron-Cascade-2-SFT-Data[†]	1,044,114	78,791
	NuminaMath	859,594	185,711
	OpenThoughts-114k	113,957	29,508
	OpenThoughts-3-1.2M	850,000	29,223
	Omega	52,218	2,042
	KlearReasoner	30,000	675
	Tier-2 subtotal	31,018,782	854,342
Tier 3 *General-purpose*	Ring-lite-sft	6,238,992	876,653
	dolphin-r1	814,334	429,561
	stepfun-sft	1,293,523	317,819
	SYNTHETIC-2	224,558	19,723
	Tier-3 subtotal	8,571,407	1,643,756
Total	28 sources	**42,794,402**	**2,710,101**

Table 7: Three-tier source-priority assignment used during deduplication, with raw and post-deduplication record counts. Tier 1 (RL-grade) covers small, carefully filtered RL datasets with verified answers; Tier 2 (math-specific SFT) covers reasoning-distillation and math-specialized SFT corpora; Tier 3 (general-purpose) covers large mixed-content corpora. When a fuzzy-deduplication component spans multiple sources, the survivor is selected by ascending tier.

order (Tier 1 over Tier 2 over Tier 3); ties within the same tier break on `record_id`, ensuring that the canonical pool retains the highest-quality copy of each duplicated problem.

A.2 Deduplication Hyperparameters

The deduplication stage runs in three steps over the post-cleaning pool. Reported settings correspond to the production configuration used to produce the canonical set of 2,710,101 problems in Table 1.

Exact deduplication. Problems whose normalized text shares an identical `xxhash64` digest are collapsed into a single record before fuzzy deduplication.

Fuzzy deduplication. We then apply two rounds of MinHash LSH on character n-grams, with each round running the full *signature* $\rightarrow$ *banded LSH query* $\rightarrow$ *exact-Jaccard verify* $\rightarrow$ *union-find* pipeline. Both rounds share the same configuration except for the Jaccard threshold:

- **Tokenization.** Character 3-grams over the normalized question text.

- **LSH index.** A banded MinHash index with **32 bands**.
- **Round 1 (coarse).** Jaccard threshold $J \geq 0.7$, which removes the bulk of obvious near-duplicates that arise from cross-citation between sources.
- **Round 2 (sensitive).** Jaccard threshold $J \geq 0.5$, run on the survivors of Round 1 to capture the remaining paraphrastic duplicates that fall below the stricter threshold.
- **Survivor selection.** Within each connected component, the survivor is chosen by source-priority tier (see Appendix A.1); ties break on `record_id`.

The two-stage threshold schedule was validated by manual inspection of the Round 1 vs. Round 2 difference set: relaxing Round 2 below 0.5 produced visibly non-duplicate pairs, while tightening Round 1 above 0.7 left obvious near-duplicates intact.

A.3 Relevance Filtering

Relevance filtering is the first routing decision: it determines whether a deduplicated problem should be removed from training entirely or retained for a later stage decision. We implement it with a conservative two-pass design. Deterministic rules handle cases whose failure mode is explicit from surface form, and a unified LLM classifier handles the remaining semantic edge cases. Both passes write into the same label/tag space, so downstream code only reads a single `filter_tag` field.

Rule-based pass. The rule pass removes records that clearly fail basic dataset quality requirements. It is intentionally high-precision rather than exhaustive: any mildly ambiguous case is deferred to the LLM classifier. The detectors cover five broad cases:

- **`non_math_signal`.** Non-mathematical content (coding, writing, advice, business, etc.).
- **`missing_context`.** Unresolved figures, tables, or links referenced from the question text.
- **`broken_format`.** Corrupted, truncated, or otherwise unreadable text (e.g. base64 artifacts, leaked chat tokens).
- **`trivial_content`.** Degenerate fragments such as bare expressions or severe n-gram repetition.
- **`other_non_english`.** Languages outside the English/Chinese scope.

In practice the rule pass removes most obvious junk before model-based classification; the `non_math_signal` bucket alone accounts for 909,285 removals in Table 1.

Unified classifier. Records surviving the rule pass are sent to a single GPT-OSS-20B classifier that makes a three-way decision with the following label semantics:

- **`drop`.** Same drop categories as the rule pass, applied to records whose failure mode requires semantic understanding rather than surface matching.
- **`midtrain`.** Mathematically relevant records that are still unsuitable for verifiable post-training (proofs, MCQs, leaked answers, meta-tasks, Chinese math).
- **`posttrain`.** Self-contained English math problems with concise, auto-checkable answers and no obvious contamination.

The classifier outputs a strict JSON object `{label, tag}`, which makes routing decisions traceable to named criteria rather than free-form generations. The exact prompt, output schema, and few-shot examples are listed in Appendix A.6 (`posttrain_filter`).

Midtrain refinement. Because the unified classifier is deliberately permissive on the `midtrain` side, records labelled `midtrain` receive one additional keep/drop check. This refinement stage uses the same model family and output schema, but only asks whether the candidate still has enough mathematical substance to justify keeping it for SFT-style supervision. Its purpose is not to revisit stage assignment, but to strip residual non-math records that slipped through because they contained superficial mathematical vocabulary. Records already labelled `posttrain` bypass this step, since their requirements are strictly stronger than the refinement criterion.

A.4 BENCHMARK DECONTAMINATION SETUP

Decontamination runs in three independently checkpointed steps: (i) *load*, which materialises the 2,176 benchmark questions across the 13 evaluation sets; (ii) *embed*, which produces a single dense vector per benchmark and per training problem; and (iii) *search-and-judge*, which retrieves nearest neighbours and dispatches them to an LLM judge.

Embedding model. Qwen3-Embedding-4B, served via SGLang's OpenAI-compatible `/v1/embeddings` endpoint (`encoding_format = float`, TP=2, DP=4). Both the benchmark and training questions are embedded under identical settings to keep the search symmetric.

Retrieval. A FAISS `IndexFlatIP` (CPU) over L2-normalized embeddingsequivalent to exact cosine nearest-neighbour searchindexes all training problems whose current routing label is `posttrain` or `midtrain`. For each benchmark question we take the **top-20 candidates** and forward all of them to the judge, with *no* hard similarity cutoff, to avoid threshold-induced misses near the decision boundary.

LLM judge. gpt-oss-20b with reasoning enabled. Decoding uses temperature 0.6, $top-p = 0.95$, and `max_tokens` $= 4096$ to leave room for the reasoning trace.

Action. A training problem flagged `decision="yes"` against any of its top-20 benchmark queries is reassigned `training_phase = "drop"` with `filter_tag = "contaminated"`; per-benchmark drop rates are written to `benchmark_drop_stats.json` for downstream analysis of which evaluations were most exposed in the raw pool.

A.5 STAGE ASSIGNMENT

Stage assignment is the second routing decision: among records that survive relevance filtering, it decides whether the sample is better suited to mid-training or to post-training. Conceptually, the split is simple. Mid-training absorbs mathematically useful but weakly supervised problems; post-training is reserved for problems whose answers support reliable verification and reward computation. We implement this decision with two shared signals: the fine-grained tags produced by the unified classifier in Appendix A.3, and an explicit rollout-based check of the labeled ground-truth.

When a problem is routed to mid-training. Two broad families of records are kept for mid-training rather than post-training. The first is *low-challenge* problems, whose format makes them poor RL targets even if they are mathematically valid; the second is *hard-to-verify* problems, whose supervision is useful but whose ground-truth cannot be reduced to a single dependable comparison target. The corresponding tags are:

- **`boolean_and_mcq`.** Multiple-choice or true/false questions, where answer options make elimination strategies possible.
- **`answer_leakage`.** The question text already reveals the target answer or key solution steps.
- **`empty_ground_truth`.** Records with no labeled answer at all (107,586 in Table 17).
- **`non_verifiable_answer`.** Proof-style, explanatory, or otherwise non-comparable answers; by far the largest mid-training bucket (236,014 records).
- **`meta_task`.** Math-adjacent meta-tasks such as grading a solution, comparing approaches, or tutoring step-by-step.

When a problem is routed to post-training. A record is admitted to post-training only if its labeled answer appears trustworthy under independent verification. For each candidate we draw $n = 8$ rollouts from GPT-OSS-120B, extract the final boxed answer from each rollout, and compare the resulting majority answer against the labeled ground-truth. Verification proceeds as a two-stage cascade:

- **Rule-based equivalence.** Handles straightforward numeric and symbolic matches after normalization and simplification.
- **LLM judge.** Resolves the cases that remain ambiguous after the rule pass; the full prompt and JSON output schema are listed in Appendix A.6 (`major_answer_judge`).

The operational point is that post-training requires agreement between the independently generated majority answer and the dataset label, rather than trusting the source annotation by default.

Verification outcomes. A sample enters the post-training pool as soon as either stage of the cascade confirms equivalence. Otherwise it is diverted to mid-training under one of three interpretable outcomes:

- **`answer_mismatch`.** The majority answer disagrees with the labeled ground-truth.
- **`gt_unclear`.** The labeled answer is itself malformed or unparseable.
- **`low_confidence`.** The eight rollouts fail to produce even a two-way agreement (`majority_count` ≤ 1); numerically small (2,269 records, 0.08% in Table 17) but informative—rollouts here are much longer on average (~46,900 tokens, about $7.8\times$ the global mean), suggesting problems near or beyond the current solver frontier.

Per-sample SFT supervision. For records that do enter post-training, we keep the eight rollout trajectories and reuse them for SFT supervision (§2.3). Downstream RL needs correctness at the trajectory level rather than only at the record level, so we run an additional pairwise equivalence pass that checks each rollout answer against the verified reference. The pass follows a rule-first protocol:

- Samples already confirmed by the rule-based check skip the LLM entirely.
- Only rule-failed cases consume GPT-OSS-20B judge capacity.

In our internal evaluation on 662,512 rollouts, this hybrid protocol raises the mean `pass_ratio` from 0.921 (rule-only) to 0.943, largely by rescuing format-equivalent answers that surface rules miss (Table 16).

A.6 Prompts

A.6.1 Relevance Filter Prompt

Decontamination prompt

Prompt: `decontamination_judge` (judge)

```
[SYSTEM PROMPT]
You are a strict benchmark decontamination judge for a math training
    corpus.
You will be given one BENCHMARK QUESTION and one TRAINING QUESTION.
Decide whether the training question is contaminated by the benchmark
     question.

Answer "yes" if ANY of the following is true:
- They are the same problem with only superficial wording changes
- One is a paraphrase, translation, or lightly reformatted version of
     the other
- The mathematical task, givens, and required answer are effectively
    the same
- The training question is an obvious template-derived rewrite of the
     benchmark question

Answer "no" if ANY of the following is true:
- They are only topically related but clearly different problems
```

```
- They share a skill area or theorem but have materially different
    givens or asks
- They use the same style or structure but different numbers/
    conditions such that they are distinct training examples

Be conservative: only output "yes" when the overlap is strong enough
    that using the training question would contaminate benchmark
    evaluation.

[USER TEMPLATE]
Benchmark source: {benchmark_source}
Benchmark id: {benchmark_id}

[Benchmark Question]
{benchmark_question}

[Training Question]
{training_question}

Decide whether the training question is contaminated by the benchmark
     question.

[OUTPUT FORMAT]
Respond with EXACTLY one JSON object on a single line, nothing else:
{"decision": "<yes_or_no>", "reason": "<brief reason>"}

Rules:
- decision must be one of: yes, no
- reason must be short and concrete
- Do NOT output any explanation outside the JSON object
```

Relenvance filter for mid-train

Prompt: `midtrain_filter` (filter)

```
[INSTRUCTION]
You are a strict data quality filter for a math training dataset.
You will be given a QUESTION (and its ANSWER if available) that has
    already been classified as "midtrain" - math-related but not
    suitable for RL.
Your job is simple: decide whether this sample should be KEPT for
    supervised fine-tuning, or DROPPED entirely because it has no
    training value.

[CATEGORY DEFINITIONS]
keep:
  The sample contains meaningful math-related content worth keeping
    for supervised fine-tuning.
  Even though it is not suitable for RL, it still has educational or
    training value. Keep it if:
  - It is a legitimate math problem written in English or Chinese,
    even if the answer is a proof, explanation, or boolean
  - It is a valid multiple-choice math question in English or Chinese
  - It is a conceptual math question ("Explain why...", "Define...")
    in English or Chinese
  - It is a math-adjacent STEM problem with significant mathematical
    content, in English or Chinese
  - It contains answer leakage or prompt contamination, but the
    underlying math question is still meaningful
  - It is a math meta-task (grading, comparing) that still
    demonstrates mathematical reasoning
```

```
  Tag: keep

drop:
  The sample should be removed because it has no useful training
    value. Drop it if:
  - [non_math_signal]   Content has nothing to do with math (coding,
    writing, advice, business, legal, medical, roleplay, etc.) in any
    language
  - [missing_context]   References external figures, tables, images,
    or prior context not provided - the question is unanswerable
    without them
  - [broken_format]     Corrupted, truncated, or unreadable text;
    HTML/base64 artifacts; diff patches; garbled characters
  - [trivial_content]   Bare expression, single number, meaningless
    fragment, or too short/nonsensical to be useful
  - [other_non_english] Math content written in a language other than
     English or Chinese (e.g. French, German, Spanish, Japanese,
    Korean, Russian, Arabic, Hindi, etc.) - we only keep English and
    Chinese math

[FEW-SHOT EXAMPLES]
Q: Prove that $\sqrt{2}$ is irrational.
A: Assume for contradiction that sqrt(2) = p/q in lowest terms...
-> {"label": "keep", "tag": "keep"}

Q:  $a + b = 5$$ab = 6$ $a^2 + b^2$
A: 13
-> {"label": "keep", "tag": "keep"}

Q: Which of the following is a prime number? (A) 15 (B) 21 (C) 23 (D)
     27
A: C
-> {"label": "keep", "tag": "keep"}

Q: Explain the difference between permutations and combinations.
A: Permutations consider order while combinations do not...
-> {"label": "keep", "tag": "keep"}

Q: Write a cover letter for a marketing manager position.
A: Dear Hiring Manager, I am writing to express my interest...
-> {"label": "drop", "tag": "non_math_signal"}

Q: Based on Table 2, calculate the mean and standard deviation.
A: Mean = 45.3, SD = 12.1
-> {"label": "drop", "tag": "missing_context"}

Q: \x00\x01\x02 garbled text NEC broken
A:
-> {"label": "drop", "tag": "broken_format"}

Q: 42
A:
-> {"label": "drop", "tag": "trivial_content"}

Q: Trouvez toutes les solutions reelles de $x^2 - 4 = 0$.
A: $x = \pm 2$
-> {"label": "drop", "tag": "other_non_english"}

[USER SUFFIX]
Now decide whether to keep or drop the following sample.

Question: {question}
Answer:   {answer}
```

```
[OUTPUT FORMAT]
Respond with EXACTLY one JSON object on a single line, nothing else:
{"label": "<LABEL>", "tag": "<TAG>"}

LABEL must be one of: keep, drop
TAG must be one of the following (and must belong to the chosen LABEL)
    :
  keep -> keep
  drop -> non_math_signal | missing_context | broken_format |
    trivial_content | other_non_english

Do NOT output any explanation, reasoning, or extra text - only the
    JSON object.
```

Relenvance filter for post-train

Prompt: `posttrain_filter` (filter)

```
[INSTRUCTION]
You are a strict math-dataset quality classifier.
You will be given a QUESTION and its ANSWER from a math training
    dataset.
Classify the sample into exactly one category and assign the most
    fitting tag from that category.

[CATEGORY DEFINITIONS]
posttrain:
  A well-formed, self-contained math problem whose answer can be
    automatically verified - suitable for reinforcement-learning
    training.
  ALL of the following must hold:
  - Clearly about mathematics (algebra, geometry, number theory,
    combinatorics, probability, calculus, linear algebra, discrete
    math, etc.)
  - Self-contained: all necessary information is provided within the
    question
  - The answer is a definite, concise, verifiable value (number,
    algebraic expression, specific mathematical object, short concrete
     conclusion)
  - The answer can be automatically checked by string/numeric
    matching - NOT a multi-step proof, long explanation, or subjective
     judgment
  - Written in English (minor inline LaTeX is fine)
  - No answer or solution steps leaked into the question text
  - No broken formatting, truncation, or corruption
  Tag: ok

midtrain:
  Math-related but NOT suitable for RL training. Route to supervised
    fine-tuning instead.
  Classify as midtrain and pick the matching tag:
  - [non_verifiable_answer] Answer is a proof, long derivation, multi-
    sentence explanation, or conceptual explanation that cannot be
    auto-checked ("Prove that...", "Show that...", "Explain why...", "
    What is the definition of...")
  - [boolean_and_mcq]       Answer is true/false/yes/no, OR multiple-
    choice question with options embedded in the text (A/B/C/D style)
    - not suitable for open-ended RL
  - [answer_leakage]        Answer or solution steps already
    partially/fully present in the question text
```

```
  - [prompt_contamination]  Reasoning traces, special tokens (<|
    im_start|>), or chat role formatting leaked into the question
  - [meta_task]             Math meta-task: grading a solution,
    comparing approaches, tutoring step-by-step, evaluating
    correctness
  - [chinese]               Math question written in Chinese - still
    useful for SFT

drop:
  Should be completely removed.
  Classify as drop and pick the matching tag:
  - [non_math_signal]   Content is not about math at all (coding,
    writing, creative writing, advice, business, legal, medical,
    roleplay, etc.) in any language; also includes STEM problems (
    physics, chemistry, engineering, biology) that are primarily about
     another field with little mathematical content
  - [missing_context]   References external figures, tables, images,
    prior problems, or context not provided ("as shown above", "Refer
    to Figure 3")
  - [broken_format]     Corrupted, truncated, or unreadable text;
    HTML/base64 artifacts; diff patches; code fences
  - [trivial_content]   Bare expression, single number, or too short/
    nonsensical to be a meaningful problem
  - [other_non_english] Math content written in a language other than
     English or Chinese (e.g. French, German, Spanish, Japanese,
    Korean, Russian, Arabic, etc.) - only English and Chinese math are
     accepted

[FEW-SHOT EXAMPLES]
Q: Find all real solutions to $x^3 - 6x^2 + 11x - 6 = 0$.
A: 1, 2, 3
-> {"label": "posttrain", "tag": "ok"}

Q: A bag contains 5 red balls and 3 blue balls. Two balls are drawn
    without replacement. What is the probability that both balls are
    red?
A: 5/14
-> {"label": "posttrain", "tag": "ok"}

Q: Prove that for all positive integers n, $1 + 2 + \cdots + n = \
    frac{n(n+1)}{2}$.
A: We proceed by induction. Base case: n=1, 1 = 1(2)/2 = 1. Inductive
     step: ...
-> {"label": "midtrain", "tag": "non_verifiable_answer"}

Q: True or false: Every continuous function on a closed interval is
    bounded.
A: True
-> {"label": "midtrain", "tag": "boolean_and_mcq"}

Q: Which is the derivative of $x^3$? (A) $x^2$ (B) $3x^2$ (C) $3x$ (D)
     $x^4$
A: B
-> {"label": "midtrain", "tag": "boolean_and_mcq"}

Q:  $f(x) = x^2 - 3x + 2$ $f(f(1))$
A: 0
-> {"label": "midtrain", "tag": "chinese"}

Q: Write a Python function that computes the Fibonacci sequence.
A: def fib(n): ...
-> {"label": "drop", "tag": "non_math_signal"}
```

```
Q: Refer to Figure 3.2 and find the area of the shaded region.
A: 12.5
-> {"label": "drop", "tag": "missing_context"}

[USER SUFFIX]
Now classify the following sample.

Question: {question}
Answer:   {answer}

[OUTPUT FORMAT]
Respond with EXACTLY one JSON object on a single line, nothing else:
{"label": "<LABEL>", "tag": "<TAG>"}

LABEL must be one of: posttrain, midtrain, drop
TAG must be one of the following (and must belong to the chosen LABEL)
    :
  posttrain -> ok
  midtrain  -> non_verifiable_answer | boolean_and_mcq |
    answer_leakage | prompt_contamination | meta_task | chinese
  drop      -> non_math_signal | missing_context | broken_format |
    trivial_content | other_non_english

Do NOT output any explanation, reasoning, or extra text - only the
    JSON object.
```

A.6.2 Stage Assignment Prompt

Answer judge prompt

Prompt: `major_answer_judge` (judge)

```
[SYSTEM PROMPT]
You are a careful mathematical answer assessor.
You will receive a math problem, a reference answer, and multiple
    candidate answers from rollout sampling.
Your job is to group equivalent candidate answers, determine the
    majority answer, and judge its relationship with the reference
    answer.

[USER TEMPLATE]
You will perform the following tasks for a math problem with multiple
     candidate answers.

Task 1: Group equivalent candidate answers.
Two answers are "equivalent" if they represent the same mathematical
    value,
even if written differently (e.g. fractions vs decimals, simplified
    vs unsimplified,
LaTeX vs plain text, reordered terms, or equivalent percentage forms).

Assign each candidate answer to a group. Answers that are empty or
    unparseable form their own singleton groups.

Task 2: Determine the majority answer.
The majority answer is the representative answer from the largest
    equivalence group.
Report the group size as majority_count.

Task 3: Judge whether the Reference Answer is clear.
```

```
"clear" means it is a legible, complete, well-formed answer to the
    problem.
"unclear" means it is empty, blank, malformed, truncated, obviously
    irrelevant,
or not a recognizable mathematical answer.
Do NOT judge whether it is mathematically correct in Task 3.

Task 4: Only if the Reference Answer is clear, judge whether the
    majority answer
(from Task 2) is mathematically equivalent to the Reference Answer.
If the Reference Answer is unclear, return "not_applicable" for the
    equivalence judgement.

Problem:
{question}

Reference Answer:
{gt_answer}

Candidate Answers (from {num_samples} rollout samples):
{candidate_answers}

[OUTPUT FORMAT]
Respond with EXACTLY one JSON object on a single line, nothing else:
{"majority_answer": "<MAJORITY>", "majority_count": <COUNT>, "
    gt_quality": "<QUALITY>", "judgement": "<JUDGEMENT>"}

MAJORITY: the representative answer string from the largest
    equivalence group.
COUNT: how many candidate answers are in that largest group (integer).


QUALITY must be one of:
  clear
  unclear

JUDGEMENT must be one of:
  equivalent
  not_equivalent
  not_applicable

Rules:
- If gt_quality is "unclear", judgement MUST be "not_applicable".
- Do not output any explanation, reasoning, or extra text.
```

A.6.3 ROLLOUT PROMPT

Prompt: `rollout_correctness` (rollout)

```
[USER TEMPLATE]
{problem}
Please reason step by step, and put your final answer within \boxed{}.
```

B TRAINING AND EVALUATION DETAILS

B.1 TRAINING INFRASTRUCTURE

We use Megatron (Shoeybi et al., 2019) as the codebase for mid-training. The base model daVinci-LLM-3B is a fully open-source model based on the Qwen2.5-3B model architecture, and it is also trained with Megatron.

We use Slime (Zhu et al., 2025) as the codebase for SFT. All SFT experiments are based on Slime, including those for daVinci-LLM, Qwen2.5-7B, and Qwen3-8B.

We use veRL (Sheng et al., 2024) as the codebase for RL and evaluation.

Compute resources. All training is performed on NVIDIA H800 GPUs. The mid-training run on the full 400B-token budget for the 3B model uses approximately 128 GPUs for 6 days ($\approx$ 18.4K GPU-hours). Full-scale SFT—both on DAVINCI-MATH and on the open-source SFT datasets used in the controlled comparisons—takes a comparable budget of $\approx$ 128 GPUs $\times$ 6 days per run. Small-scale SFT ablation runs use $\approx$ 16 GPUs for 12h on 3B models ($\approx$ 192 GPU-hours per run) and $\approx$ 16 GPUs for 24h on 8B models ($\approx$ 384 GPU-hours per run). RL with GRPO uses $\approx$ 128 GPUs for 8 days ($\approx$ 24.6K GPU-hours). Trajectory synthesis with GPT-OSS-120B for the mid-training and SFT pools uses $\approx$ 128 GPUs for 2 days each ($\approx$ 6.1K GPU-hours per pool). Auxiliary LLM passes in the data pipeline (relevance filter, decontamination judge, answer verification) account for a negligible fraction of the total compute relative to the figures above.

B.2 TRAINING HYPER-PARAMETERS

We use separate training configurations for mid-training, supervised fine-tuning (SFT), and reinforcement learning (RL), as summarized in Tables 8, 9, and 10. For mid-training, we use the same optimization setup across all runs and vary only the number of training steps according to the token budget. Specifically, the 100B-token setting is trained for 6,000 steps, while the 400B-token setting is trained for 24,000 steps. For SFT, all experiments are trained for 6 epochs with a cosine learning-rate schedule and 10% warmup. For RL, we adopt GRPO with multiple rollouts per prompt. Following prior work, we remove standard-deviation normalization (Liu et al., 2025c;a) and use MIS (Liu et al., 2025b) to improve training stability.

Table 8: Mid-training hyperparameters.

Params.	Value
Global batch size	512
Max sequence length	32,768
LR	3×10^{-5}
LR schedule	Constant
Optimizer	Adam
Weight decay	0.01

Table 9: SFT hyperparameters.

Params.	Value
Global batch size	1,024
LR schedule	Cosine
Max LR	3×10^{-4}
Warmup ratio	10%
Weight decay	0.01
Epochs	6

Table 10: RL hyperparameters.

Params.	Value
Algorithm	GRPO
Rollout batch size	256
Samples per prompt	16
Global batch size	1,024
LR	1×10^{-6}
Std. normalization	Removed

B.3 TRAINING DATA

In the mid-training baseline, the proportions of the individual sources are shown in Table 11. For the mid-training setup based on Davinci-Math, the QA component can be fully replaced with midtrain part of Davinci-Math.

For the SFT stage, we compare the SFT portions of DaVinci-Math, OpenMathReasoning, Nemotron-Math, and NemotronCascade. For OpenMathReasoning, we use the CoT subset. For Nemotron-Math, we use only the CoT subset rolled out by gpt-oss-high-120b. For NemotronCascade, we use the CoT subset from its math portion. For the full version of each dataset, we report the data statistics in Table 13. For the SFT datasets discussed in §3.3, we report the data statistics in Table 12.

Table 11: Composition of the mid-training data mixture.

Type	Corpus / Dataset Group	Dataset	Weight
Corpus	Nemotron-CC-Math	nemotron-cc-math-v1-4plus	0.101108
		nemotron-cc-math-v1-3	0.155141
		nemotron-cc-math-v1-4plus-MIND	0.140770
	MegaMath	MegaMath-Web-Pro-Max	0.143765
	UltraMath	Ultramath-L3	0.159217
	Subtotal	–	**0.70000**
QA	NemotronMath	NemotronMath	0.150000
	OpenMathReasoning	OpenMathReasoning	0.150000
	Subtotal	–	**0.300000**
Total	–	–	**1.00000**

Table 12: Matched-budget small-scale SFT dataset statistics. All datasets are aligned to an SFT token budget of approximately 1.892B tokens.

Dataset	Samples	Tokens
daVinci-Math-SFT	300,000	1,892,057,306
OpenMathReasoning	241,437	1,892,064,897
NemotronCascade	102,136	1,892,059,255
NemotronMath	158,401	1,892,077,325

Table 13: Full-scale SFT dataset statistics for the four datasets compared in our experiments.

Dataset	Samples	Tokens
daVinci-Math-SFT	3,875,078	24,551,007,931
OpenMathReasoning	3,201,061	25,133,023,903
NemotronCascade	2,142,332	39,740,745,044
NemotronMath	2,336,251	28,004,435,236

B.4 Evaluation Parameters

For all evaluations we use a sampling temperature of 0.6 and top-p of 0.95, and verify correctness with the DeepScaleR verifier. Per-benchmark sampling budgets are summarized in Table 14.

Table 14: Per-benchmark sampling budgets used throughout the paper.

Benchmark	Used in	Metric
AIME 25, AIME 26, HMMT Feb 25, HMMT Nov 25	In-domain math (§3.2–§3.3)	avg@32
MATH-500	In-domain math (§3.2–§3.3)	avg@1
GPQA-D, SciBench, PIQA, FinQA	OOD generalization (§3.4)	avg@8

C Detailed Dataset Statistics

C.1 Per-source curation breakdown

Two patterns in Table 15 are worth noting. First, dedup compression varies dramatically by source—*Llama-Nemotron-math_v1* drops from 19.8M raw to 134K canonical (99.3% removal) due to heavy internal duplication, while small RL-grade sources like *deepscaler* retain 11.0% of their raw size. Second, the routing distribution reflects source character: distillation-heavy sources (*OpenMath-Reasoning*, *Nemotron-Math-v2*) yield a high share of post-training-eligible records, while general-purpose corpora (*Ring-lite-sft*, *dolphin-r1*, *stepfun_sft*) are dominated by the discard bucket because most of their content is not mathematical.

Table 15: Per-source breakdown. We report the size of each source after our format conversion and light filtering (Raw), the number of canonical problems surviving exact and two-round n-gram deduplication (Dedup'd), and the absolute count routed to each destination by the classification pipeline (§2.2), with percentages of the deduplicated set in gray subscripts. Upstream provenance and licenses are listed separately in Table 6.

Dataset	Raw	Dedup'd	Discard	Mid-train	Post-train
Ring-lite-sft	6,238,992	876,653	491,485 56.1%	236,550 27.0%	148,618 17.0%
dolphin-r1	814,334	429,561	379,142 88.3%	50,419 11.7%	0 0.0%
stepfun_sft	1,293,523	317,819	295,364 92.9%	19,893 6.3%	2,562 0.8%
Nemotron-Math-v2	324,185	200,910	16,182 8.1%	120,149 59.8%	64,579 32.1%
numinamath	859,594	185,711	62,986 33.9%	76,202 41.0%	46,523 25.1%
OpenMathReasoning	5,678,317	154,864	12,612 8.1%	64,784 41.8%	77,468 50.0%
Llama-Nemotron-math_v1	19,840,970	133,541	47,829 35.8%	28,328 21.2%	57,384 43.0%
Nemotron-Cascade-2-SFT-Data	1,044,114	78,791	12,822 16.3%	53,626 68.1%	12,343 15.7%
OpenR1-math	450,258	66,660	22,147 33.2%	17,879 26.8%	26,634 40.0%
DeepMath-103K	103,022	51,895	4,690 9.0%	23,798 45.9%	23,407 45.1%
Llama-Nemotron-math_v1.1	2,225,427	39,077	5,997 15.3%	14,284 36.6%	18,796 48.1%
OpenThoughts-114k	113,957	29,508	20,120 68.2%	7,911 26.8%	1,477 5.0%
OpenThoughts-3-1.2M	850,000	29,223	4,832 16.5%	19,533 66.8%	4,858 16.6%
Skywork-OR1-RL-Data	105,055	23,191	7,007 30.2%	7,710 33.2%	8,474 36.5%
SYNTHETIC-2	224,558	19,723	19,569 99.2%	150 0.8%	4 0.0%
openreasoning_57k_72k	129,322	18,347	5,210 28.4%	1,899 10.4%	11,238 61.3%
MiroMind-m1-62k	62,118	16,810	4,637 27.6%	4,920 29.3%	7,253 43.1%
polaris-data-53K	341,373	9,615	2,113 22.0%	3,398 35.3%	4,104 42.7%
Dolci-Instruct-RL	63,998	6,441	1,364 21.2%	1,167 18.1%	3,910 60.7%
deepscaler	40,315	4,435	850 19.2%	971 21.9%	2,614 58.9%
dapo-math-17k	1,791,700	3,972	538 13.5%	1,926 48.5%	1,508 38.0%
STILL-3-RL	29,925	3,097	616 19.9%	803 25.9%	1,678 54.2%
openreasoning_13k_hard	13,451	2,102	619 29.4%	452 21.5%	1,031 49.0%
Dolci-Think-RL-7B	30,180	2,077	687 33.1%	424 20.4%	966 46.5%
omega	52,218	2,042	639 31.3%	49 2.4%	1,354 66.3%
Dolci-Think-RL-32B	30,182	2,015	625 31.0%	369 18.3%	1,021 50.7%
Dolci-RL-Zero-Math-7B	13,314	1,346	204 15.2%	164 12.2%	978 72.7%
KlearReasoner	30,000	675	230 34.1%	25 3.7%	420 62.2%
Total	**42,794,402**	**2,710,101**	**1,421,116 52.4%**	**757,783 28.0%**	**531,202 19.6%**

C.2 PER-SOURCE ROLLOUT STATISTICS

Table 16 reports per-source rollout statistics on the 531,202 post-training queries that survived classification, routing, and answer verification. Each query was sampled $n = 8$ times from GPT-OSS-120B, yielding 4.2M rollout trajectories in total. The columns separate per-query length (averaged within a query, then aggregated) from per-sample length (aggregated across all rollouts), so that sources whose individual rollouts vary widely in length can be distinguished from sources with uniformly short or long generations. Three patterns are worth highlighting:

- **Length distribution spans an order of magnitude across sources.** Per-sample mean trajectory length ranges from ∼1,250 tokens for *Llama-Nemotron-math_v1* (short, calculator-style problems) to ∼16,500 tokens for *Dolci-Think-RL-7B* (multi-step competition problems). The P95 statistics reach 53,401 tokens for *OpenMathReasoning*, exposing the long tails that dominate context-length budgeting and rollout cost.
- **Pass ratio varies systematically with source character.** The average fraction of the 8 rollouts that match ground-truth after the hybrid verification cascade (Appendix A.5) ranges from 0.798 (*Nemotron-Math-v2*) to 0.980 (*omega*). Low-pass-ratio sources concentrate either on formally hard problems or on noisy ground-truth conventions; RL-curated sources tend to cluster above 0.94, consistent with their tighter answer schemas.
- **Volume is heavily long-tailed.** The top three sources (*Ring-lite-sft*, *OpenMathReasoning*, *Nemotron-Math-v2*) supply more than half of all post-training samples, while the long tail of small RL-grade sources (*KlearReasoner*, *deepscaler*, the *Dolci-** family, etc.) contributes under 5% in volume but is over-represented in the verifiable RL subset of §2.2.3.

Table 16: Per-source rollout statistics for post-training queries. For each source we report the number of post-training queries (Queries), the total number of GPT-OSS-120B rollout samples (n=8 per query, Samples), the mean rollout length in tokens averaged over all rollouts (Mean), the subset of samples whose answers match the ground-truth and are therefore retained for SFT (Correct), the mean length of these correct samples (Mean$_c$), and the per-query pass ratio (average fraction of the 8 rollouts matching the ground-truth).

Dataset	Queries	Samples	Mean	Correct	Mean$_c$	Pass ratio
Ring-lite-sft	148,581	1,188,648	4,406	1,112,620	4,049	0.916
OpenMathReasoning	77,466	619,728	13,135	552,110	11,033	0.871
Nemotron-Math-v2	64,573	516,584	12,435	404,152	11,284	0.798
Llama-Nemotron-math_v1	57,384	459,072	1,557	446,723	1,391	0.938
NuminaMath	46,523	372,184	3,452	352,777	3,025	0.938
OpenR1-math	26,634	213,072	4,813	202,241	4,571	0.945
DeepMath-103K	23,407	187,256	6,125	175,616	5,816	0.905
Llama-Nemotron-math_v1.1	18,796	150,368	9,693	139,488	8,839	0.910
Nemotron-Cascade-2-SFT-Data	12,343	98,744	16,270	84,380	15,031	0.855
openreasoning_57k_72k	11,238	89,904	5,518	87,283	5,183	0.947
Skywork-OR1-RL-Data	8,472	67,776	10,637	63,507	9,267	0.947
MiroMind-m1-62k	7,253	58,024	8,853	53,651	8,100	0.906
OpenThoughts-3-1.2M	4,858	38,864	14,736	33,186	13,083	0.831
polaris-data-53K	4,102	32,816	13,066	30,749	12,135	0.914
Dolci-Instruct-RL	3,910	31,280	7,717	30,277	7,129	0.949
DeepScaleR	2,613	20,904	7,494	20,045	6,748	0.934
stepfun-sft	2,562	20,496	3,882	19,351	3,782	0.929
STILL-3-RL	1,678	13,424	6,849	12,831	6,252	0.928
dapo-math-17k	1,507	12,056	8,293	11,807	7,870	0.968
OpenThoughts-114k	1,477	11,816	6,353	10,974	5,665	0.920
Omega	1,354	10,832	4,726	10,725	4,326	0.980
openreasoning_13k_hard	1,030	8,240	10,257	7,798	9,497	0.928
Dolci-Think-RL-32B	1,021	8,168	15,300	7,534	14,108	0.894
Dolci-RL-Zero-Math-7B	978	7,824	10,542	7,534	9,705	0.944
Dolci-Think-RL-7B	965	7,720	15,386	7,096	14,203	0.902
KlearReasoner	420	3,360	4,979	3,296	4,899	0.968
Total	**531,202**	**4,249,616**	**7,294**	**3,875,078**	**6,341**	**0.903**

C.3 ROUTING-LABEL AND VERIFICATION STATISTICS

We complement the per-source breakdown with two pipeline-level views of the routing stage in §2.2. Table 15 reports how many deduplicated records carry each quality label and are therefore routed to

mid-training. Table 16 reports the outcome distribution of the subsequent answer-verification stage on the residual problems that pass the quality filters.

Table 17: Complete inventory of filter labels produced by the pipeline. Labels in the *posttrain* block correspond to records that pass all quality checks (with `equivalent_rescued` also routed to posttrain after the LLM judge accepts mathematically-equivalent answers); the *midtrain* block lists problems that survive but cannot be used for rule-verifiable RL; the *drop* block lists records removed entirely. Counts are over the full 2,710,101-record canonical pool.

Label	Origin	Records	%
Routed to `posttrain`			
`(none)`	rule	259,327	9.57
`equivalent_rescued`	LLM	275,287	10.16
Routed to `midtrain`			
`non_verifiable_answer`	rule + LLM	236,014	8.71
`answer_mismatch`	LLM	152,524	5.63
`boolean_and_mcq`	rule	109,869	4.05
`empty_ground_truth`	rule	107,586	3.97
`answer_leakage`	rule + LLM	69,410	2.56
`chinese`	rule	48,794	1.80
`gt_unclear`	LLM	16,706	0.62
`meta_task`	rule + LLM	4,033	0.15
`low_confidence`	rule	2,269	0.08
`answer_uncertain`	rule	2,044	0.08
Dropped			
`non_math_signal`	rule + LLM	909,285	33.55
`broken_format`	rule + LLM	231,537	8.54
`missing_context`	rule + LLM	197,824	7.30
`trivial_content`	LLM	28,276	1.04
`other_non_english`	rule	22,974	0.85
`short_question`	rule	20,231	0.75
`bare_expression`	rule	7,828	0.29
`prompt_contamination`	rule + LLM	5,122	0.19
`contaminated`	decontam	2,564	0.09
`phantom_reference`	rule	538	0.02
`broken_reference`	rule	59	0.00

D FULL EVALUATION RESULTS

D.1 FULL MAIN RESULTS

This section complements the summary results in Table 2 and the controlled comparisons in §3.3. While the main paper emphasizes average performance to highlight the central trends, the tables below report the full per-benchmark breakdown for the midtraining comparison, the matched-budget SFT comparison, and the combined midtraining–SFT study. These detailed results help verify that the improvements attributed to DAVINCI-MATH are broadly consistent across AIME, HMMT, and MATH-500 rather than being driven by a single benchmark.

Table 18: Detailed midtraining comparison under matched downstream SFT evaluation.

Midtraining budget	Midtraining data	AIME25	AIME26	HMMT Feb	HMMT Nov	MATH-500	AVG
100B	NemoMath-OMR	29.48	32.50	21.98	22.81	81.20	37.59
	DAVINCI-MATH	31.25	36.77	24.17	23.85	85.60	**40.33**
400B	NemoMath-OMR	36.25	40.52	26.77	26.98	87.80	43.66
	DAVINCI-MATH	41.98	48.13	29.27	31.67	87.20	**47.65**

Table 19: Detailed matched-budget small-scale SFT comparison across base models. All SFT datasets are matched to approximately 1.892B tokens. AVG is the unweighted mean across the five benchmarks.

Base model	SFT dataset	AIME25	AIME26	HMMT Feb	HMMT Nov	MATH-500	AVG
DAVINCI-LLM-3B	DAVINCI-MATH	21.67	26.77	16.15	16.56	81.00	**32.43**
	Nemotron-Cascade-2	14.58	15.42	10.94	10.21	77.00	25.63
	Nemotron-Math	18.23	19.27	10.52	9.90	74.80	26.54
	OpenMathReasoning	22.50	22.92	16.15	15.10	82.10	31.76
QWEN2.5-7B	DAVINCI-MATH	52.71	58.33	40.10	41.35	90.80	**56.66**
	Nemotron-Cascade-2	47.92	48.96	34.48	35.00	90.40	51.35
	Nemotron-Math	36.25	45.52	26.15	27.81	85.20	44.19
	OpenMathReasoning	37.40	41.56	23.75	30.63	90.60	44.79
QWEN3-8B-BASE	DAVINCI-MATH	64.27	67.71	50.83	48.65	93.20	**64.93**
	Nemotron-Cascade-2	64.38	62.29	52.19	50.31	94.60	64.75
	Nemotron-Math	63.23	65.10	46.67	46.98	92.60	62.92
	OpenMathReasoning	43.85	52.08	28.96	36.67	92.80	50.87

Table 20: Detailed midtraining + SFT comparison on daVinci-LLM-3B.

Midtraining data	SFT dataset	AIME25	AIME26	HMMT Feb	HMMT Nov	MATH-500	AVG
DAVINCI-MATH	DAVINCI-MATH	40.73	47.60	28.54	32.19	90.40	**47.89**
DAVINCI-MATH	Nemotron-Cascade-2	40.52	41.56	29.27	28.12	87.20	45.34
DAVINCI-MATH	Nemotron-CoT	37.60	40.62	26.04	29.48	86.20	43.99
DAVINCI-MATH	OpenMathReasoning	29.48	32.50	23.12	22.50	86.40	38.80
NemoMath-OMR	DAVINCI-MATH	39.06	45.94	29.48	27.71	88.00	46.04
NemoMath-OMR	Nemotron-Cascade-2	34.48	38.54	22.71	23.65	85.60	41.00
NemoMath-OMR	NemotronMath	30.10	34.79	21.46	21.15	84.00	38.30
NemoMath-OMR	OpenMathReasoning	29.58	33.54	19.58	20.62	87.00	38.07

D.2 FULL ABLATION RESULTS

This subsection provides additional ablation results. Figure 6 and Table 21 report the stage-decomposition analysis referenced in §3.2, showing that each of the three training stages contributes a substantial gain. Table 22 reports the complete three-seed factorized ablation results (all five routing variants and both pipeline-component variants) discussed in §4.1 and §4.2, together with the two additional benchmarks (AIME 26, HMMT Nov) not shown in the main-text tables. Together, these results show that the main-text conclusions are not driven by a single benchmark, but remain broadly consistent across all four competition benchmarks.

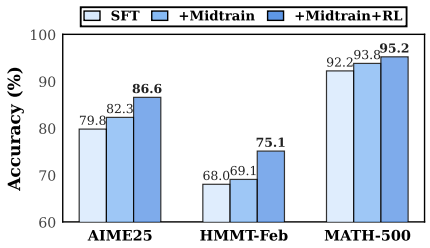


Figure 6: Effect of progressively adding training stages on top of the same DAVINCI-LLM-3B. Each stage yields a substantial gain, and the full pipeline achieves the strongest result.

Table 21: Full evaluation results for the progressive stage-addition study.

Setting	AIME25	AIME26	HMMT Feb	HMMT Nov	MATH-500	avg5
SFT only	79.79	80.31	68.02	63.65	92.20	76.79
Midtraining + SFT	82.29	83.48	69.06	64.90	93.80	78.71
Midtraining + SFT + RL	86.56	85.94	75.10	69.38	95.20	82.44

Table 22: Full factorized ablation results under the reduced-scale protocol, over three independent training seeds. Each entry reports mean ± standard error; Avg. is the mean over the four listed benchmarks.

Setting	AIME 25	AIME 26	HMMT Feb	HMMT Nov	Avg.
DAVINCI-MATH	**36.15 ± 1.11**	**42.40 ± 1.21**	**24.69 ± 1.00**	**23.65 ± 0.68**	**31.72 ± 0.50**
Routing (§4.1)					
w/o low-challenge routing	33.23 ± 0.72	38.23 ± 1.40	21.98 ± 0.78	22.08 ± 0.94	28.88 ± 0.54
distribution-matched random routing	32.40 ± 0.98	35.83 ± 1.18	22.29 ± 0.77	22.29 ± 1.12	28.20 ± 0.59
w/o routing	29.69 ± 1.01	38.02 ± 1.15	22.71 ± 0.84	21.98 ± 1.09	28.10 ± 0.52
w/o hard-to-verify routing	31.25 ± 1.17	37.40 ± 1.21	20.73 ± 0.82	22.92 ± 0.89	28.07 ± 0.46
swap midtrain ↔ SFT	32.81 ± 1.02	35.52 ± 0.79	19.69 ± 1.02	20.94 ± 1.01	27.24 ± 0.44
Pipeline components (§4.2)					
w/o LLM classification	32.60 ± 1.12	38.65 ± 1.28	22.29 ± 1.01	23.54 ± 0.88	29.27 ± 0.45
w/o answer verification	31.46 ± 0.99	37.71 ± 1.24	20.00 ± 0.85	21.35 ± 1.01	27.63 ± 0.50

E Additional Robustness Analyses

E.1 Mid-Training Topic Diversity

To quantify whether daVinci-Math's mid-training corpus covers a broader span of mathematical content than existing open corpora, rather than simply being a larger sample of the same distribution, we compare topic diversity across four mid-training datasets. From each dataset we draw a fixed-seed sample of 10,000 unique problems, classify every problem against the same 52-topic taxonomy (He et al., 2025b), and score all four datasets with the same judge (Qwen2.5-32B). We summarize each dataset's coverage with the effective number of domains and effective number of topics (the exponential of the Shannon entropy over the judge's topic assignments), which penalizes concentration on a few dominant topics more than a raw topic count would.

Table 23: Topic diversity of mid-training corpora, measured on a fixed-seed 10,000-problem sample per dataset under a shared 52-topic taxonomy and a common Qwen2.5-32B judge.

Dataset	Effective domains	Effective topics
daVinci-Math	7.54	34.49
Nemotron-Math-v2 CoT	7.55	31.92
Nemotron-Cascade-2 no-tool	7.58	31.79
OpenMathReasoning	6.35	26.92

daVinci-Math attains the highest effective-topic count despite domain coverage comparable to the two Nemotron variants, indicating that its problems are more evenly spread across the taxonomy rather than concentrated in a handful of dominant topics.

E.2 Controlled Comparisons with Standard Errors

Table 24 reports standard errors for a representative subset of the controlled comparisons in §3.3 (SFT across base models, large-scale SFT, and mid-training at 100B/400B tokens), alongside the strongest open-data alternative for each setting. Accuracy here is averaged over four benchmarks only (AIME25, AIME26, HMMT Feb, HMMT Nov), excluding MATH-500; consequently these numbers are not directly comparable to the five-benchmark averages reported for the same comparisons in §3.3 and Appendix D.1, and should be read as an independent uncertainty estimate rather than a corrected version of those results.

Table 24: Representative controlled data comparisons, accuracy $\pm$ standard error (%). "Strongest alternative" is the best-performing open dataset baseline for each comparison.

Comparison	daVinci-Math	Strongest alternative
SFT on daVinci-LLM-3B	22.42 ± 0.43	OMR: 19.61 ± 0.35
SFT on Qwen3-8B	65.21 ± 0.65	Nemotron-Cascade-2: 63.88 ± 0.41
Large-scale SFT	72.01 ± 0.48	Nemotron-Cascade-2: 64.40 ± 0.62
Mid-training, 100B + mini-SFT	34.01 ± 0.48	NemoMath-OMR: 30.57 ± 0.66
Mid-training, 400B + mini-SFT	42.40 ± 0.47	NemoMath-OMR: 37.73 ± 0.48

E.3 Output-Token-Length Comparisons

A natural concern with any data-curation method that improves accuracy is that the gain is bought by longer, more verbose generations at inference time rather than genuine reasoning improvement. We report average output length (generated tokens per response) alongside accuracy for the same controlled comparisons as Appendix E.2, plus the routing-ablation settings of §4, to check whether daVinci-Math's accuracy gains come with a token-efficiency cost. As in Appendix E.2, accuracy and output length are both averaged over the same four benchmarks (AIME25, AIME26, HMMT Feb, HMMT Nov), excluding MATH-500, so the accuracy figures here are likewise not directly comparable to the five-benchmark averages reported elsewhere in the paper for the same comparisons.

Table 25: SFT data comparison across base models. Each cell reports accuracy (%) / average generated tokens per response.

Base model	DAVINCI-MATH	Nemotron-Cascade-2	Nemotron-Math-v2	OMR
DAVINCI-LLM-3B	22.42 / 24,502	14.82 / 27,307	15.42 / 26,119	19.61 / 14,715
Qwen2.5-7B	55.36 / 20,126	47.71 / 22,953	36.95 / 23,299	34.09 / 13,939
Qwen3-8B	65.21 / 18,473	63.88 / 20,376	60.89 / 19,654	42.06 / 13,139

Table 26: SFT data comparison on mid-trained checkpoints. Performance (%) and average output tokens.

Mid-trained checkpoint	SFT data	Performance	Output tokens
DAVINCI-MATH	DAVINCI-MATH	44.45	21,836
DAVINCI-MATH	Nemotron-Cascade-2	40.39	23,769
DAVINCI-MATH	Nemotron-Math-v2	35.65	23,154
DAVINCI-MATH	OMR	27.66	14,368
NemoMath-OMR	DAVINCI-MATH	40.08	22,513
NemoMath-OMR	Nemotron-Cascade-2	31.90	24,791
NemoMath-OMR	Nemotron-Math-v2	30.10	24,137
NemoMath-OMR	OMR	26.25	14,238

As a direct check against token-count inflation in the routing ablations of §4, Table 29 reports performance and output length for the same ablation settings, evaluated as the unweighted average over AIME25, AIME26, HMMT Feb 2025, and HMMT Nov 2025. The full DAVINCI-MATH pipeline ("Baseline") achieves both the highest accuracy and the shortest average output among these settings, indicating that removing any single pipeline component both hurts accuracy *and* increases verbosity rather than trading one for the other.

E.4 PAIRED BOOTSTRAP CONFIDENCE INTERVAL

To quantify the uncertainty of the headline comparison between DAVINCI-MATH-3B and Qwen3.5-4B, we compute paired, question-level bootstrap confidence intervals under the original 80K-token evaluation protocol, both per benchmark and stratified across the full suite (Table 30). Because each benchmark contributes only 30 questions, the per-benchmark intervals are individually wide and one (HMMT Nov 2025) crosses zero; the stratified overall interval, which pools questions across all four benchmarks, is tighter and remains above zero ($+3.39$ points, 95% CI $[0.18, 6.64]$). We do not additionally re-run the full mid-training–SFT–RL pipeline across multiple random seeds to estimate training-side variance—doing so at the 400B-token scale would require disproportionate compute ($\sim$20K H800 GPU-hours per additional seed)—so this interval quantifies evaluation-time sampling uncertainty for a fixed pair of trained checkpoints, not end-to-end training variance.

E.5 EVALUATION-PROTOCOL ROBUSTNESS CHECK

The main results in Table 2 use a $81{,}920$-token (80K) generation cap. Long chain-of-thought responses on benchmarks with high token budgets (e.g., HMMT) can occasionally be truncated under this cap. To check whether this affects the ranking of results, we additionally evaluate a subset of models under a 256K-token cap with Math-Verify grading. This is reported as an independent robustness check on the subset of models for which both protocols are available, not as a replacement for Table 2.

The relative ranking and the direction of DAVINCI-MATH's advantage over Qwen3.5-4B are unchanged under the longer-context protocol.

Table 27: Large-scale SFT comparison. Performance (%) and average output tokens.

Experiment	Performance	Output tokens
DAVINCI-MATH-V2 large-scale SFT	72.01	17,099
Nemotron-Cascade-2 full-set SFT	64.40	20,056
Nemotron-Math-v2 large-scale SFT	26.77	24,532
OMR-3.2M large-scale SFT	45.73	12,165

Table 28: Mid-training comparison. Performance (%) and average output tokens.

Experiment	Performance	Output tokens
DAVINCI-MATH MT 100B + mini-SFT	34.01	23,508
DAVINCI-MATH MT 400B + mini-SFT	42.40	22,391
NemoMath-OMR MT 100B + mini-SFT	30.57	24,022
NemoMath-OMR MT 400B + mini-SFT	37.73	22,984

Table 29: Performance and output length for the reduced-scale ablation checkpoints. Performance is the unweighted average over AIME25, AIME26, HMMT Feb 2025, and HMMT Nov 2025; output tokens are averaged over the same four benchmarks.

Ablation setting	Performance	Output tokens
w/o answer verification	27.63	23,437
w/o LLM classification	29.27	23,313
swap midtrain $\leftrightarrow$ SFT	27.24	23,675
w/o routing	28.10	23,593
Baseline	**31.72**	**22,945**

Table 30: Paired, question-level bootstrap confidence intervals for DAVINCI-MATH-3B minus Qwen3.5-4B, under the 80K-token protocol.

Benchmark	Difference	Approx. paired 95% CI
AIME 2025	$+6.66$	$[-0.94, 14.37]$
AIME 2026	$+3.13$	$[0.94, 5.73]$
HMMT Feb 2025	$+5.52$	$[-0.42, 12.60]$
HMMT Nov 2025	-1.77	$[-9.37, 6.67]$
Stratified overall	$\mathbf{+3.39}$	$[\mathbf{0.18}, \mathbf{6.64}]$

Table 31: Context-length robustness check: average accuracy (%) under the main 80K-token protocol vs. a 256K-token protocol with Math-Verify, for the subset of models evaluated under both.

Model	80K avg	256K avg
Qwen3.5-4B	79.65	80.44
DAVINCI-MATH-3B	82.44	86.89