Big-Math: A Large-Scale, High-Quality Math Dataset for Reinforcement Learning in Language Models

Alon Albalak[1], **Duy Phung**[1], **Nathan Lile**[1], **Rafael Rafailov**[†2], **Kanishk Gandhi**[2], **Louis Castricato**[1], **Anikait Singh**[2], **Chase Blagden**[1], **Violet Xiang**[2], **Dakota Mahan**[1] **and Nick Haber**[2]
[1]SynthLabs, [2]Stanford University

Increasing interest in reasoning models has led math to become a prominent testing ground for algorithmic and methodological improvements. However, existing open math datasets either contain a small collection of high-quality, human-written problems or a large corpus of machine-generated problems of uncertain quality, forcing researchers to choose between quality and quantity. In this work, we present Big-Math, a dataset of over 250,000 high-quality math questions with verifiable answers, purposefully made for reinforcement learning (RL). To create Big-Math, we rigorously filter, clean, and curate openly available datasets, extracting questions that satisfy our three desiderata: (1) problems with uniquely verifiable solutions, (2) problems that are open-ended, (3) and problems with a closed-form solution. To ensure the quality of Big-Math, we manually verify each step in our filtering process and iteratively improve our filters over multiple rounds. Based on the findings from our filtering process, we introduce 47,000 new questions with verified answers, Big-Math-Reformulated: closed-ended questions (i.e. multiple choice questions) that have been reformulated as open-ended questions through a systematic reformulation algorithm. Compared to the most commonly used existing open-source datasets for math reasoning, GSM8k and MATH, Big-Math is an order of magnitude larger (250,000 questions vs. 8,000 questions in GSM8k and 12,000 in MATH), while our rigorous filtering ensures that we maintain the questions most suitable for RL. We also provide a rigorous analysis of the dataset, finding that Big-Math contains a high degree of diversity across problem domains, and incorporates a wide range of problem difficulties, enabling a wide range of downstream uses for models of varying capabilities and training requirements. In conclusion, this work presents our new dataset, Big-Math, the largest open dataset of math problems suitable for RL training. By bridging the gap between data quality and quantity, Big-Math establish a robust foundation for advancing reasoning in LLMs. Big-Math and Big-Math-Reformulated are available at https://huggingface.co/datasets/SynthLabsAI/Big-Math-RL-Verified.

1. Introduction

In the past few years, mathematics has emerged as a critical testing ground for the development and evaluation of advanced reasoning techniques used in Large Language Models (LLMs) [Cobbe et al., 2021, Wei et al., 2023, Schaeffer et al., 2023, Mirzadeh et al., 2024, OpenAI et al., 2024, DeepSeek-AI et al., 2025, inter alia]. Following the release of strong reasoning models, such as OpenAI's o1 [OpenAI et al., 2024] and DeepSeek's R1 [DeepSeek-AI et al., 2025], a plethora of supervised fine-tuning (SFT) datasets have been released with the aim to distill reasoning capabilities into other models, or to bootstrap a model prior to RL [Liu et al., 2024, Li et al., 2025, AI, 2025, Labs, 2025, Muennighoff et al., 2025]. However, knowledge distillation has its limitations. For example, research has shown that while SFT distillation enables models to memorize reasoning patterns and solutions, the resultant models do not generalize well [Chu et al., 2025]. On the other hand, RL

Filtering and reformulation code available at https://github.com/SynthLabsAI/Big-Math.
Prompts for all filters and reformulation can be found in Appendix D.
[†]Work done while at SynthLabs.

training algorithms have been shown to yield models with better generalization to novel problems by emphasizing the exploration and refinement of reasoning strategies. Furthermore, [Wang et al., 2025] find that current reasoning models (across model sizes and families) frequently achieve the correct answer with incorrect reasoning steps, suggesting that distilled reasoning models will also suffer from incorrect reasoning. Additionally, OpenAI et al. [2025] show that scaling RL training is very effective in improving reasoning and other capabilities. Therefore, *while distillation through SFT datasets is clearly a helpful method for improving model performance on some evaluations, it does not address the greater needs at the forefront of reasoning capabilities which can be addressed by RL: learning how to reason* [Xiang et al., 2025, Yeo et al., 2025, Kim et al., 2025].

A significant bottleneck in RL-based reasoning research is the lack of high-quality datasets tailored for reinforcement learning. RL training methods all assume access to a dataset with verifiable answers [DeepSeek-AI et al., 2025, Team et al., 2025]; either to train a reward model [Lightman et al., 2023b, Havrilla et al., 2024b, Zhang et al., 2024, Mahan et al., 2024] or to directly evaluate the correctness of a generated solution and, as has been demonstrated through the course of deep learning's existence, scaling up data quantity and quality is a crucial step towards success [Havrilla et al., 2024a]. However, *a central issue with the existing math datasets is that they are either (1) human written, but limited in quantity, or (2) machine generated and large, but of unknown quality, forcing researchers to choose datasets with either quality or quantity.* For example, prior works have mostly used the GSM8k [Cobbe et al., 2021] and MATH [Hendrycks et al., 2021] datasets. While they contain human-written questions and answers, both datasets are quite limited in quantity, with 8,000 and 12,000 problems, respectively. On the other hand, large-scale datasets, such as NuminaMath [Li et al., 2024b], exhibit quality issues, including many duplicate problems and incomplete solutions, hindering their utility in RL training. Furthermore, many existing math datasets contain problems which are not well-suited for training a reasoning model with RL. For instance, they sometimes have a high proportion of multiple choice questions. While multiple choice questions may be useful for SFT, they are less effective for RL-based training. Recent works [Xiang et al., 2025], suggest that the goal of RL for reasoning is not just to train the model to answer correctly, but *more importantly*, to reason correctly. With this in mind, questions in multiple choice formats are problematic for RL; even though the correct answers may be difficult to deduce, the model can simply guess the correct answer without performing the correct reasoning. Problems like these highlight the biggest issues with curating a dataset for RL.

To address these challenges, we present Big-Math, a dataset of over 250,000 high-quality math problems and solutions, curated with three core desiderata:

1. **Uniquely verifiable solutions:** problems must admit a single correct answer that can be reliably verified;
2. **Open-ended problem formulations:** problems that cannot be easily solved by guessing (as might occur in multiple choice formats) and instead require nontrivial reasoning steps; and
3. **Closed-form solutions:** answers must be expressible in a closed form (e.g., a scalar or formula, not a proof), thereby enabling automated evaluation.

With these desiderata in mind, we design and develop a process of cleaning and curating datasets by applying a strict set of filters. To ensure the quality of our filtration process, we apply a human-in-the-loop algorithm, iteratively improving each filter through multiple rounds of manual verification until the filter achieves suitable levels of precision and recall (minimum of 90%). Then, we apply our filtration process to three openly available datasets, extracting a high-quality and large-scale subset of over 200,000 uniquely verifiable, open-ended problems with closed-form solutions. Additionally, we introduce Big-Math-Reformulated, a novel subset of 47,000 problems derived by reformulating multiple-choice questions into an open-ended format, while preserving their integrity and complexity

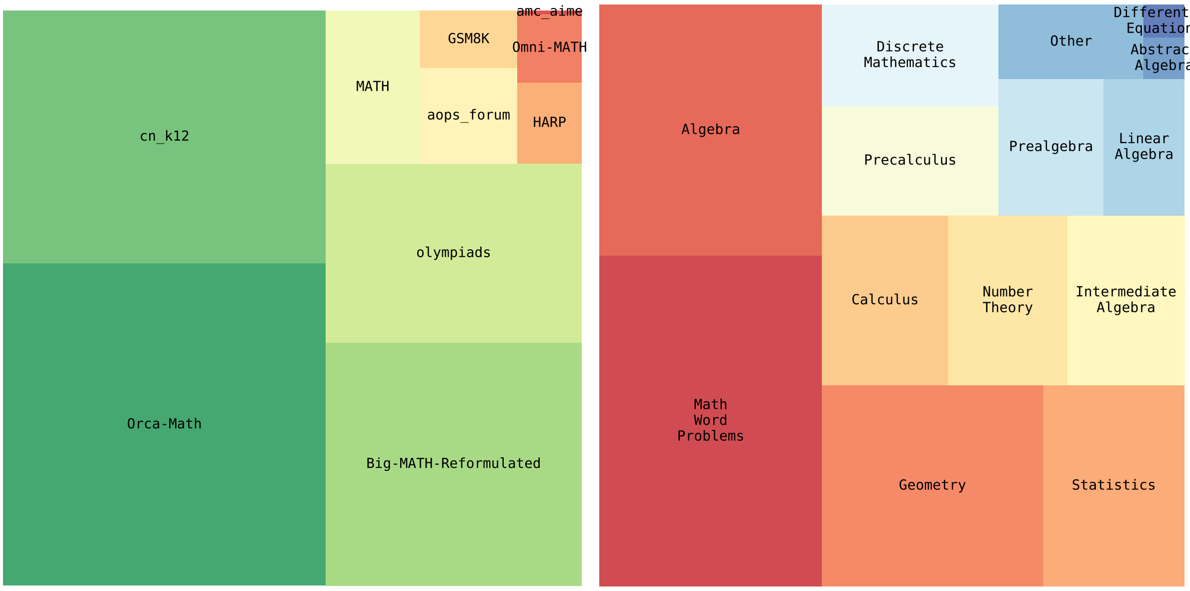


Figure 1: The sources (left) and domains (right) of the BIG-MATH dataset, scaled by size.

and aligning with RL training requirements. In combination, our filtered datasets and reformulated problems create BIG-MATH, a dataset of over 250,000 questions with extracted answers, specifically designed for RL training.

In summary, this work provides:

- BIG-MATH, the largest open source dataset of math problems, designed specifically for RL, with over 250,000 problems.
- A rigorous filtering and cleaning pipeline, involving humans-in-the-loop, and achieving over 90% precision and recall.
- BIG-MATH-REFORMULATED, a dataset of 47,000 novel problems, reformulated from existing multiple choice problems, and the full reformulation pipeline.
- Empirical analyses highlighting the diversity, difficulty, and suitability of BIG-MATH for advancing reasoning in LLMs.

2. BIG-MATH

In this section, we describe the technical details of our data collection, cleaning, filtering, and reformulation processes.

2.1. Dataset Collection

We considered a number of openly available math datasets and selected 3 well established mathematical problem datasets that are commonly used in recent literature: HARP [Yue et al., 2024], Omni-MATH [Gao et al., 2024], and NuminaMath [Li et al., 2024b]. The goal of this work is to find a large set of high-quality math problems that satisfy our three desiderata. Human-written problems are generally associated as being high-quality, and the datasets that we select provide a large quantity of human-written problems to start with ($> 75\%$). Additionally, we choose to incorporate synthetically generated data, but limit ourselves to a single, well established source Table 1 shows a breakdown of the sources, including the original quantity of data and the amount after filtering.

Data Source	**Original**	**Big-Math**
Orca-Math [Mitra et al., 2024]	153,314	83,215
cn_k12 [Li et al., 2024b]	276,554	63,609
olympiads [Li et al., 2024b]	150,563	33,485
MATH [Hendrycks et al., 2021]	12,000	9,257
aops_forum [Li et al., 2024b]	30,192	5,740
GSM8k [Cobbe et al., 2021]	7,473	3,254
HARP [Yue et al., 2024]	4,780	2,996
Omni-MATH [Gao et al., 2024]	4,428	2,478
amc_aime [Li et al., 2024b]	4,070	78
Big-Math-Reformulated (Section 2.3)	–	47,010
Total	643,374	251,122

Table 1: **Comparison of problems by data source** with the original quantity and the quantity included in Big-Math. Big-Math-Reformulated is a new problem set introduced in this work and does not have an "original" size.

HARP First, we include the Human Annotated Reasoning Problems (HARP) dataset [Yue et al., 2024], which contains 4,780 short answer problems collected from U.S. national math competitions. We specifically take only the "short answer" subset, which should not contain any questions requiring proofs, or multiple choice questions.

Omni-MATH Next, we incorporate the Omni-MATH dataset [Gao et al., 2024], contributing almost 4,500 olympiad-level problems curated from 39 different competition websites. Notably, this dataset employs professional annotators and verifiers to maintain solution-answer quality.

NuminaMath Finally, we use the NuminaMath [Li et al., 2024b] dataset, which is composed of roughly 860,000 problems divided into 9 sources (6 of which we retain). First, there are three sources that we do not use from Numina: the synthetic_math, synthetic_amc, and the MATH subsets. We choose not to include either of the synthetic data subsets as they was never evaluated independently from the remainder of the dataset, so we cannot know their quality. Additionally, while NuminaMath uses the original split of the MATH dataset (7,000 training problems, 5,500 test problems), we choose to use version with 12,000 training problems and 500 test problems, as proposed by [Lightman et al., 2023a]. The largest subset from Numina is the cn_k12 subset of data, composed of $\sim 275,000$ math problems scraped from chinese math exams. The next largest subset from NuminaMath is the Orca-Math [Mitra et al., 2024] subset ($\sim 150,000$ problems), which is a synthetically generated dataset of grade school math problems. Although this data was synthetically generated, Mitra et al. [2024] demonstrate that as the lone source of SFT training data, this subset leads to impressive math performance, even for small language models. Next, we include the olympiads subset, consisting of 150,000 problems collected from online sources of international-level math competitions. Next, Li et al. [2024b] scrape the Art of Problem Solving forum for additional math competition problems, selecting to keep only problems with high quantities of LaTeX and at least one `\boxed` or $\blacksquare$ symbol, leading to an additional 30,000 problems. Additionally, we use the GSM8k [Cobbe et al., 2021] subset, providing another $\sim 8,000$ problems. Finally, we also include the amc_aime subset, consisting of 4,000 more math competition problems, with solutions scraped from online sources.

2.2. Dataset Cleaning and Filtering

The collection of all the above datasets leads us to a combined dataset of over 640,000 problems. However, this dataset likely has many duplicated questions, undesirable content, and data that does not satisfy our desiderata. To achieve a dataset of the highest quality, and appropriate for RL training, we next clean and filter the data from each source using a combination of bespoke and common strategies [Albalak et al., 2024, Soldaini et al., 2024, Li et al., 2024a]. We ensure the quality of our filters by applying human-in-the-loop methodology, iteratively refining the filters through human verification and annotation of positive and negative examples. By the end of the iterative process our filters achieve over 90% F1 score, oftentimes reaching much higher than 90%. After filtering, the data should contain only problems that closely follow our three desired properties: open-ended, verifiable, closed-form problem-solution pairs.

2.2.1. Source-specific Filtering and Cleaning

The first step in our cleaning and filtering process is to observe a sampling of data from each dataset, and to design bespoke filters to be utilized on each source separately based on their unique idiosyncrasies.

HARP For the HARP dataset [Yue et al., 2024], we find many problems that contain figures in the Asymptote[1] vector graphics language, which we identify by string matching for "[asy]". We err on the side of caution and remove 625 such problems (13% of the dataset), assuming that the model would need to see the rendered image to solve the question.

Omni-MATH When exploring Omni-MATH [Gao et al., 2024], we found a number of problems containing author attributions (e.g. a person's name in parenthesis, or in the format "[i] Name [/i]") and manually revise each problem to remove the attribution (45 in total). Next, we found that some problems contain information about their scores in the competition and remove this text from the problem. Specifically, we found the phrase "If the correct answer is `X` and your answer is `Y`" in a number of problems. Additionally, we found and removed 2 problems with the following solution: "The problem provided does not contain a solution. Therefore, no final answer can be extracted.", likely a parsing error when extracting problem-solution pairs from online sources.

NuminaMath Within NuminaMath [Li et al., 2024b], we find some unique characteristics of each subset. First, because some of the subsets are quite large, we deduplicate problems within each subset with a MinHashLSH filter. We use 128 hashing functions, and through a few rounds of experiments, empirically determine that a similarity threshold of 0.6 or 0.7 (depending on the subset) is the strictest threshold we can set without the filter quality degrading. Next, the NuminaMath dataset does not explicitly contain answers to each problem, so we extract answers by searching for boxed solutions ("\boxed{}" in LaTeX). Any problem whose solution does not contain exactly 1 boxed answer is filtered out. Finally, for the aops_forum subset, we find 2535 problems containing unnecessary information, including problem attribution (e.g. "proposed by"), year of submission, point scoring (e.g "(1 point)") and remove these strings with regular expressions.

At the end of source-specific filtering, we find that we have 463,426 remaining problems. However, these problems are not yet fit for use, as this data still contains problems not suitable for RL training.

[1] https://asymptote.sourceforge.io/

Subset	Multiple Choice	True/False	Yes/No	Multi-Part	Proof	Semantic Duplicates	Total Matching
Orca-Math	1,277	5	12	18,375	5	8,286	26,487
cn_k12	96,804	420	384	38,625	5,233	2,098	123,809
olympiads	12,688	311	4,615	22,116	30,357	1,362	53,102
MATH	1,124	0	3	2,885	31	7,694	9,714
aops_forum	1,665	85	126	3,162	4,608	262	7,853
GSM8k	46	0	0	792	0	7,664	7,999
HARP	303	2	7	288	2	155	695
Omni-MATH	149	126	206	467	207	378	1,182
amc_aime	2,962	0	22	1,274	66	53	3,066
Total Matching	117,018	949	5,375	87,984	40,509	27,952	233,907

Table 2: **The quantity of data matched by each filter.** Some data matches multiple filters, so column totals are a summation, but row totals are not. We see some interesting trends here: the cn_k12 subset contains a significant amount more multiple choice questions than any other subset, olympiads contains many more proof questions, but also yes/no questions than other subsets. Interestingly, but unsurprisingly, we also see that nearly the entire MATH and GSM8k datasets contain semantic duplicates with other subsets.

2.2.2. Source-agnostic Filtering

After running each of the described filters over the individual subsets, we perform 11 filtering operations across the full collection. These filters are specially designed to convert the raw dataset of question-answer pairs into a dataset that is suitable for training a math reasoning model with RL. Just as for the source-specific filters, we also manually verify the results of each source-agnostic filter, and iteratively improve their performance until they achieve suitable results (where appropriate). The high-level results of our filtering process are found in Table 2, with more detailed results in Appendix A. Prompts for model-based filters are found in Appendix D.1.

Deduplication and Decontamination To prevent a model from unknowingly being trained on the same problem too frequently (and risk overfitting), we need to remove duplicates. To handle this, we run a very simple and efficient deduplication step, calculating duplicate problem by string matching (not including whitespace), removing all but one copy of the duplicated problem. Next, we also want to ensure that the final dataset has a diverse set of problems, without focusing on any specific problem types. To do so, we remove semantic duplicates (problems with similar meaning, e.g. the same problem with numbers changed) with the SemDeDup algorithm [Abbas et al., 2023]. To embed the problems, we use the model at sentence-transformers/all-MiniLM-L6-v2 and remove problems with a cosine similarity over 0.5. For semantic deduplication, we tested thresholds between 0.95 and 0.2, finding that even the strictest thresholds can cluster dissimilar problems (likely caused by short problems). We want to ensure that no duplicated data passes through, so we purposefully select a threshold where our manual verification determines that no duplicates remain. Finally, the MATH [Hendrycks et al., 2021] and Omni-MATH [Gao et al., 2024] test sets are prime candidates for evaluating a model trained on our dataset, so we need to ensure that the problems in those test sets do not exist in our training set. To decontaminate our dataset with any of the 500 MATH test set problems and the 500 Omni-MATH test set problems we find and remove all contaminated data by a string matching algorithm on the problems, the same as for deduplication. We were surprised to find minimal duplication ($< 1\%$), removing only 4,229 problems, with the majority of duplicates being between the Orca-Math (3,916), GSM8k (3,875), and cn_k12 (405) datasets.

Ensuring that problems are solvable For a number of reasons, there are problems in existing math datasets which are not solvable.

(Language Filter) First, in this work, we focus on English-only models and require English math problems. So, we use a FastText language identifier [Joulin et al., 2016b,a, Grave et al., 2018] and remove any problems where English is not the primary language (only 101 non-English problems were detected).[2] Through our iterative improvement process, we found that it was important to remove LaTeX, along with most special characters (e.g. numbers, math symbols, "()[]{}!@#$%ˆ&", etc.) in order to achieve a high level of accuracy with the language detection model. Additionally, problems which were very short (< 10 characters) were often classified as non-English (even if they were entirely numbers), so we simply include all problems with fewer than 10 non-LaTeX, non-special characters.

(Hyperlink Detection) Next, we remove problems containing a hyperlink using a simple regular expression, as the existence of hyperlinks suggests that a model may not have the full resources required to solve the problem (e.g. hyperlinks that point to a website containing a theorem). While this likely removes problems that are solvable (hyperlinks may also link to the website source of the problem), we prefer to err on the side of caution and remove all problems containing hyperlinks. This over-filtering can be addressed in the future through more complex filters.

(Model Solve Rate) Finally, while it is not feasible to manually ensure the correctness of each problem-answer pair, we develop a heuristic for correctness using language models. For each problem, we generate 64 solutions from Llama-3.1-8B ($\sim 30,000,000$ rollouts) and 5-8 solutions from Llama-3.1-405B ($\sim 1,100,000$ rollouts, generated on a pre-filtered subset) [Dubey et al., 2024]. If either model answers the question with the ground truth answer, then we determine that the question-answer pair may be valid. We do not apply this filter to HARP, Omni-Math, MATH, or GSM8k as these datasets include pre-parsed answers.

Of course, this method does not guarantee that the given answer is correct, as it is possible that the answers fall under a commonly made mistake, or that the models have seen the data during pre- or post-training. Furthermore, this filter does not guarantee that removed data has an incorrect answer, as it is very likely that the models we use cannot solve the most difficult math problems. One method for improving this filter would be to use a stronger, math-specific model.

Ensuring that problems are open-ended An important aspect of reinforcement learning is that the training signal should appropriately attribute the good actions with high rewards and poor actions with low rewards. Therefore, problems with multiple choice answers pose a problem: the model can inadvertently respond with the correct answer option (generally between a 25-50% chance of guessing correctly) without providing the correct intermediate reasoning steps, leading to a poor learning signal. For this reason, we choose to remove any problems that are multiple choice, True/False, and Yes/No. To detect and remove all three types of questions, we develop both a regular expression-based filter, and a model-based filter.

(Regular Expression Filters) For multiple choice questions, we use a simple regular expression filter that searches for either alphabetic options (A, B, C, D) or numerical options (1, 2, 3, 4), occurring in order. To ensure that we do not incidentally remove questions referring to shapes (e.g. "rectangle ABCD...") or numbers (e.g. 1234), we first remove those strings from the question, prior to the regular expression search. Next, for the True/False questions, we search for either "true" or "false" in the answer or, when available, in the final line of the solution. Then, for Yes/No questions, we perform the same check as True/False questions, searching in the answer or solution for exact phrases.

[2] We use the model at https://huggingface.co/facebook/fasttext-language-identification.

Additionally, we search the final line of the question for specific phrases that imply a Yes/No question: “is”, “are”, “do”, “does”, and “can”. While these regular expression filters are guaranteed to remove all data of this exact form, they are not flexible enough to catch many exceptions.

(Model-based Filters) Therefore, for each question type, we design a model-based filter by iteratively developing a prompt to use with Llama-3.1-70B [Dubey et al., 2024]. For example, to develop the prompt for multiple choice questions, we first manually find examples of multiple choice problems and open-ended problems and include these as in-context examples. Next, we run our filter over the dataset and inspect 100 problems classified as positive examples (multiple choice) and 100 problems classified as negative examples (open-ended). We iteratively add the difficult incorrectly classified problems into the prompt, mostly selecting problems following a previously unseen pattern. For these filters we prioritize a high recall, ensuring that we remove as much undesirable data as possible. In this iterative process of filter development, we continue until achieving over 98% recall, requiring between 5-8 rounds of manual verification.

Ensuring that problems are uniquely verifiable A critical aspect of training reasoning models with reinforcement learning is the existence of verifiable answers, generally in the form of a ground truth to be compared against the model response.

(Answer Filter) As a simple first step, we remove all examples where the final answer did not previously exist, or could not be extracted from the solution (e.g. if there was no “\boxed{}” element in the solution).

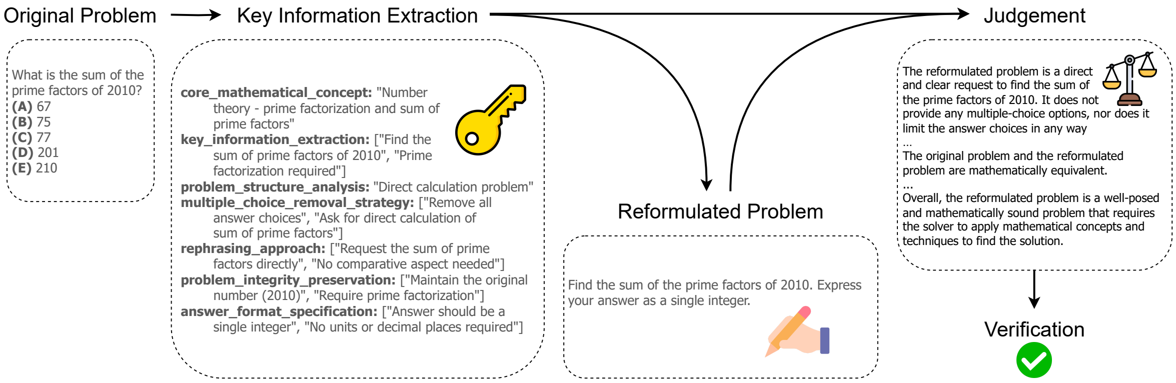
(Multi-part Question Filter) Next, we find a large proportion of questions with multiple parts requiring multiple corresponding answers. There are surely methods with which to handle partial correctness in multi-part questions, but this is still an open research question requiring further study, so we leave these problems for future, more difficult, versions of the dataset. To develop our multi-part question filter, we follow a similar pattern to the multiple choice questions filter: a dual-filter approach including a regular expression filter and a model-based filter. For the regular expressions, we iteratively improve the filter through manual search of the data and improvements to the regular expressions. The final version of our filter searches for commonly found signals: ordered roman numerals (e.g. I, II), multiple numbered parts in parentheses (e.g. (1) ... (2)), multiple numbered parts with a period (e.g. 1. ... 2.), as well as numbered special characters (e.g. ① ... ②). For the model-based filter, we use the same iterative process as for the multiple choice question, using a Llama-3.1-70B [Dubey et al., 2024], inspecting 100 examples of positively and negatively classified examples at each iteration, ending only once we achieve over 98% recall.

(Proof Filter) While proofs can be verified, there can be many correct variations, and how to quickly verify proofs written in natural language is unclear at the moment (other than converting to a theorem proving language, which can incur additional parsing errors). Therefore we also elect to remove proofs from our dataset, leaving them for a future version which is more difficult. We develop our proof filter with the same dual-method approach previously discussed. The regular expression filter simply searches for either of the following phrases in the problem: “prove that” or “a proof”. We again implement the model-based filter through an iterative improvement process with Llama-3.1-70B, inspecting 100 examples of predicted positive and negative examples, completing the process only once 98% recall is achieved.

2.3. Big-Math-Reformulated

Based on findings from the previous sections we introduce a new subset of 47,000 questions and answers, Big-Math-Reformulated, which we describe here.

During the development of our filters, we found a staggering number of multiple choice questions



What is the sum of the prime factors of 2010?
(A) 67
(B) 75
(C) 77
(D) 201
(E) 210

core_mathematical_concept: "Number theory - prime factorization and sum of prime factors"
key_information_extraction: ["Find the sum of prime factors of 2010", "Prime factorization required"]
problem_structure_analysis: "Direct calculation problem"
multiple_choice_removal_strategy: ["Remove all answer choices", "Ask for direct calculation of sum of prime factors"]
rephrasing_approach: ["Request the sum of prime factors directly", "No comparative aspect needed"]
problem_integrity_preservation: ["Maintain the original number (2010)", "Require prime factorization"]
answer_format_specification: ["Answer should be a single integer", "No units or decimal places required"]

Find the sum of the prime factors of 2010. Express your answer as a single integer.

The reformulated problem is a direct and clear request to find the sum of the prime factors of 2010. It does not provide any multiple-choice options, nor does it limit the answer choices in any way
...
The original problem and the reformulated problem are mathematically equivalent.
...
Overall, the reformulated problem is a well-posed and mathematically sound problem that requires the solver to apply mathematical concepts and techniques to find the solution.

Figure 2: **Our reformulation strategy** first extracts key information, then reformulates the problem as open-ended, followed by a judgement of the reformulation and a final verification. The full process is detailed in Section 2.3.1, with successful and failed examples of reformulation in Appendix C.

($> 117,000$), which are removed during the filtering process for BIG-MATH. In particular, we are concerned with the loss of significant quantities of data from human-written and high-quality sources: olympiads, amc_aime, and aops_forum. The inherent structure of multiple choice questions presents a challenge for RL algorithms. Specifically, the multiple-choice format increases the probability of answering correctly regardless of the correctness of a reasoning chain. This can lead the algorithm to assign high rewards to incorrect reasoning, reducing the utility of such questions in an RL environment. To address this issue, we propose a novel approach to reintroduce these valuable questions by reformulating the multiple choice questions into open-ended questions with a series of carefully devised and detailed steps enacted by large language models. Our reformulation process is developed to be systematic and rigorous, in order to ensure that the integrity and complexity of the original questions are maintained, while making them amenable for RL. We include examples of a successful reformulation and a failed reformulation in Figure 9.

2.3.1. Reformulation strategy

We develop a 4-step process to reformulate multiple choice problems as open-ended problems. All prompts can be found in Appendix D.2, and we use Llama-3.1-405B [Dubey et al., 2024] to create BIG-MATH-REFORMULATED.

(Key Information Extraction) We begin the process by identifying a few core pieces of information about each question. We determine whether the question is in the multiple choice format, extract the core mathematical concepts (e.g. geometry), and identify key problem details (e.g. distances, goal of the question). In the first step, we also ask the language model to develop a strategy to convert the question into an open-ended format, as well as strategies for rephrasing and ensuring that the integrity of the original problem is maintained. Finally, we extract a plan for what the format of the final answer should be (e.g. the answer should be expressed in cm^3).

Source	**Number of Problems**
cn_k12	35,211
olympiads	9,019
amc_aime	1,878
aops_forum	853
MATH	48
Orca-Math	1

Table 3: **Composition of problems in BIG-MATH-REFORMULATED**, reformulated from problems in the source column.

(Reformulation) Next we reformulate the multiple choice question into an open-ended question by conditioning on the key extracted information extracted.

(Judgement) Following the reformulation, a critical evaluation along multiple dimensions is conducted to evaluate whether the reformulation succeeded. This includes determining whether the new question is multiple choice, or if it even has a limited number of possible answers which would still be considered multiple choice. Additionally, we evaluate whether the reformulated problem is free from any information distortion: no existing information was changed, no critical information was omitted, and no new information was added. Finally, we determine whether the expected answer format is clear and if the problem can stand alone, without reference to the original options.

(Verification) Finally, we perform a last verification that the entire process succeeded by checking that the expected information is present (e.g. expected answer format, rephrasing strategy, etc.).

2.3.2. Reformulation Post-Processing

The outcome of our reformulation is 88,983 questions that have passed the judgement and verification steps. However, we still need to ensure that the questions in Big-Math satisfy the same criteria: uniquely verifiable, open-ended problems with closed-form solutions. Therefore, we next examine the solvability of each reformulated problem by evaluating Llama-3.1-8B (8 rollouts) and Llama-3.1-405B (3 rollouts) on them. We filter the 88,983 problems down to only 48,698 by keeping only problems that were solved at least once by either model, but not solved 100% by Llama-3.1-8B (to remove questions that may be too simple or obvious). In instances where neither model produces a solution that matches the reformulated answer, it is difficult to determine whether the issue lies with the model's performance, or with the reformulated answer. To mitigate such uncertainty, we exclude these problems from the dataset.

Finally, Big-Math-Reformulated undergoes the same comprehensive filters as the rest of the datasets, as outlined in Section 2.2.2, yielding a final set of 47,010 reformulated and filtered problems, with the composition of source data found in Table 3. We find that this process has successfully reintroduced high-quality questions that were previously removed. Specifically, 72.7% of the amc_aime subset was found to be multiple choice, but the reformulation process successfully reintroduces 63.4% of the multiple choice problems.

3. Analysis and Discussion

In this section we discuss and analyze the Big-Math dataset. We consider the dataset difficulty, diversity, and outcomes of the filters that we proposed. Throughout the analysis, we include discussion points to aid the downstream uses of Big-Math.

3.1. Dataset Difficulty

We calculate the difficulty of problems in Big-Math based on rollouts from the Llama-3.1-8B model [Dubey et al., 2024], which provide a benchmark for understanding problem complexity. For each problem in the dataset we generate 64 rollouts and calculate the success rate per problem. The distribution of success rates, split by source and domain is found in Figures 3 and 4.

First, Figure 3 shows that the majority of the easiest data (highest solve rate) comes from the Orca-Math, cnk_12, and MATH datasets, while the most difficult data is divided more evenly across the datasets. In particular, we find that nearly all of Omni-MATH and HARP are unsolveable by Llama-3.1-8B. Thus, in order to apply RL for Llama-3.1-8B on these difficult subsets, this particular model would need to be either be supervised fine-tuned on these datasets, or using an RL training algorithm that makes use of a process reward model. For example, Reinforcement Learning with Verifiable Rewards (RLVR) [Lambert et al., 2024] would be unlikely to work effectively on Omni-Math

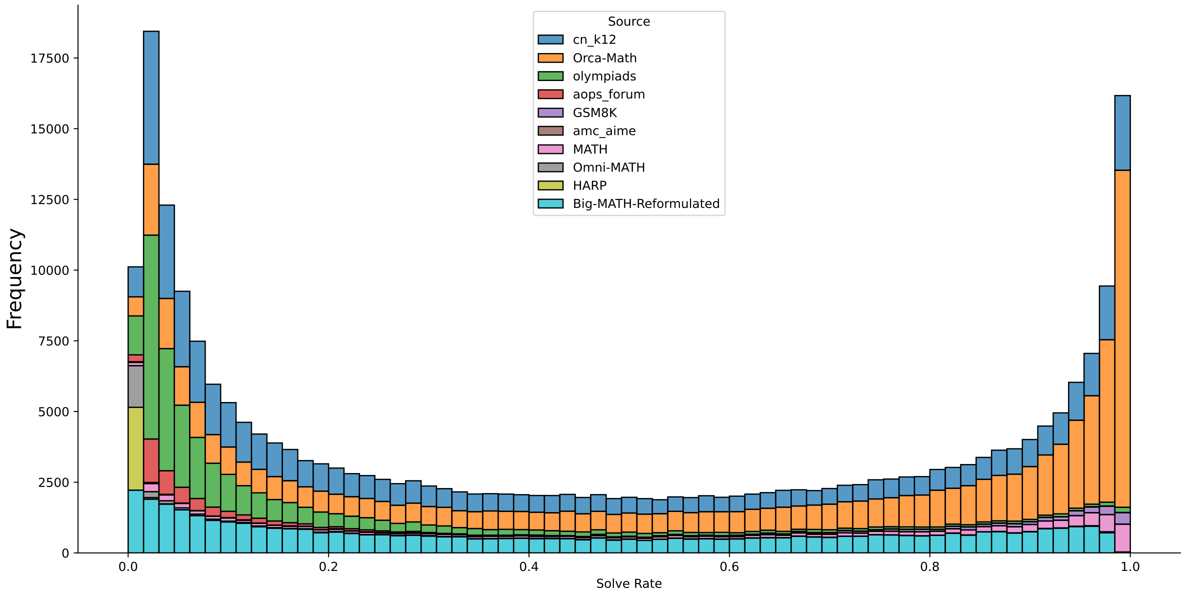


Figure 3: Distribution of solve rates on each subset of Big-Math, calculated with Llama-3.1-8B.

and HARP as the models responses would produce no training signal.

Next, we group problems into difficulty quintiles, with the hardest quintile being problems that have a success rate less than 20% and the easiest quintile with a success rate over 80%. We find that, from easiest to hardest, the quintiles have 71,926 (28.64%), 30,533 (12.16%), 25,763 (10.26%), 31,249 (12.44%), and 91,647 problems (36.50% of the total problems). An obvious question now is: **how should practitioners use these dataset difficulties for their own purposes?** In general, those training less capable, or smaller, models may want to remove the most difficult problems as it is unlikely that model rollouts will lead to a correct answer. This leads to inefficiency in the learning process because most RL methods used for LLMs (except those with a process reward model) will have 0 signal if the model can never come to the correct answer. On the other hand, for those training a larger, or math-specific, model will find many of the easy questions redundant, and training on such data will be inefficient. Therefore, for practitioners training strong models it would be sensible to keep only the harder problems. Supposing that the hardest two quintiles of data are retained, there is still $> 120,000$ problems, *10 times more problems than the next closest RL-suitable dataset.*

Next, we look at the difficulty of our novel Big-Math-Reformulated subset. We see that our subset follows a similar solve rate distribution as the rest of the dataset; it has slightly more density around the low- and high-ends of the difficulty distribution. However, Big-Math-Reformulated is skewed towards more difficult problems. Specifically, we find that 34.44% of Big-Math-Reformulated is in the hardest quintile, with an additional 16.42% in the second hardest quintile, combining to greater than 50% of the new data.

Finally, we look into the distribution of solve rates by each problem domain, shown in Figure 4. We find that the most difficult problems come from the differential equations, discrete mathematics, and abstract algebra domains, while the prealgebra domain is the easiest by a wide margin. Interestingly, the remaining domains have a very wide distribution of difficulties, suggesting that within each domain there are likely problems requiring varying levels of expertise. Surprisingly, linear algebra was found to be one of the easier domains, while geometry was one of the most difficult domains, however, this may either be an artifact of the domain classification process, or of the specific training data for Llama-3.1-8B.

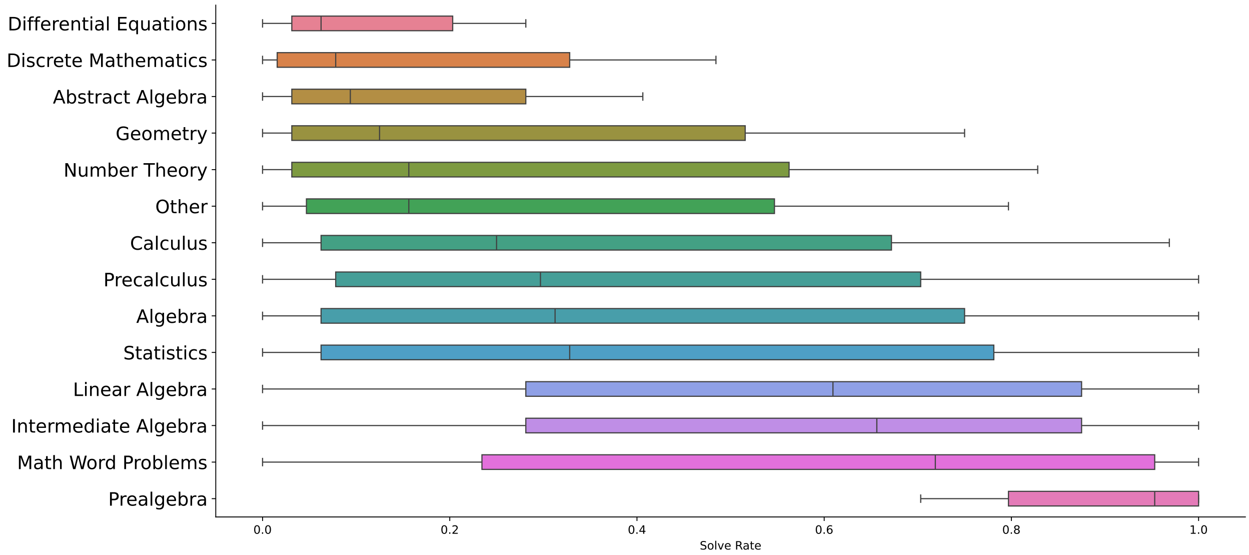


Figure 4: **Distribution of solve rates by domain**, calculated with Llama-3.1-8B.

3.2. Dataset Diversity

To better understand our dataset and the diversity of problems it provides, we next study the domain composition and distribution of our dataset.

We consider the distribution of problems sorted into mathematics domains according to two ontologies. The first ontology we consider is that proposed by Gao et al. [2024], and the second is the ontology defined by the American Mathematical Society in their 2020 Mathematics Subject Classification (MSC[3]). We use the same prompt and procedure as Gao et al. [2024] to classify questions into their domains. Similar to Muennighoff et al. [2025], we classify problems into one of the 63 domains defined by the MSC using GPT-4o-mini. The domains of Gao et al. [2024] contain high-level topics ranging from grade-school to college-level, while the MSC domains are more fine-grained and include more detailed domains on math-adjacent topics. For the domains defined by Gao et al. [2024], we present examples in Figure 6 and a chart with the number of problems per domain in Figure 5, as well as a detailed breakdown of domains by source in Section B in the Appendix. For the MSC domains, Figure 5 shows the distribution of all domains with 10 or more problems, and Figure 8 shows example problems for the top-5 most common domains.

We see in Figures 5 and 7 that the distributions for both ontologies have long tails. The largest domain from Gao et al. [2024] is the Math Word Problems, which we find to come disproportionately from Orca-Math, which contains $> 66,000$ such problems. Surprisingly, we find that the largest MSC domain was operations research, and we inspected the data under this category to better understand where such a large quantity came from. We found that these problems are generally an application of algebra, geometry, or statistics to real world domains and could just as easily have been classified into another category. We found that the distribution of MSC domains follows a nearly log-linear relation, with large quantities of problems in college-level topics, including $\sim 10,000$ problems on ordinary differential equations, field theory, and optimal control. We also found smaller quantities of applications in the sciences in the dataset, with problems in electromagnetic theory, thermodynamics, and fluid mechanics, showing that the collected dataset contains problems from a wide variety of mathematics domains.

[3] https://mathscinet.ams.org/mathscinet/msc/msc2020.html

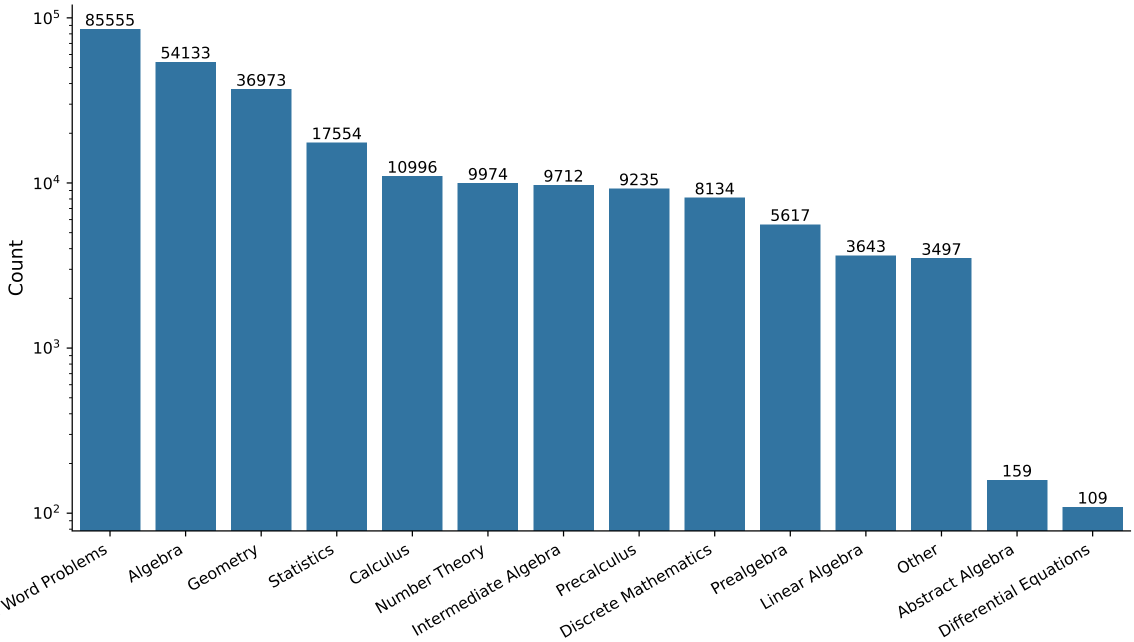


Figure 5: **Number of problems per domain**, as classified by gpt-4o-mini using math domains as defined by Gao et al. [2024]. Note that the y-axis is log-scale.

Example Problem: Math Word Problem

Question: A shopper buys a 100 dollar coat on sale for 20% off. An additional 5 dollars are taken off the sale price by using a discount coupon. A sales tax of 8% is paid on the final selling price. Calculate the total amount the shopper pays for the coat. Express your answer in dollars, rounded to two decimal places.
Answer: 81.00

Example Problem: Algebra

Question: Real numbers x and y satisfy $4x^2 - 5xy + 4y^2 = 5$. If $s = x^2 + y^2$, then $\frac{1}{s_{\max}} + \frac{1}{s_{\min}} = ?$
Answer: $\frac{8}{5}$

Example Problem: Geometry

Question: Let $ABCD$ be a cyclic quadrilateral, and suppose that $BC = CD = 2$. Let I be the incenter of triangle ABD . If $AI = 2$ as well, find the minimum value of the length of diagonal BD .
Answer: $2\sqrt{3}$

Example Problem: Statistics

Question: A three-digit natural number abc is termed a "convex number" if and only if the digits a, b, and c (representing the hundreds, tens, and units place, respectively) satisfy $a < b$ and $c < b$. Given that a, b, and c belong to the set $\{5, 6, 7, 8, 9\}$ and are distinct from one another, find the probability that a randomly chosen three-digit number abc is a "convex number".
Answer: $\frac{1}{3}$

Example Problem: Calculus

Question: Find the sum $\frac{1}{2!} + \frac{2}{3!} + \frac{3}{4!} + \ldots + \frac{n}{(n+1)!}$ and compute its limit as $n \to \infty$.
Answer: 1

Figure 6: **Example problems from the top-5 most common domains defined by Gao et al. [2024].**

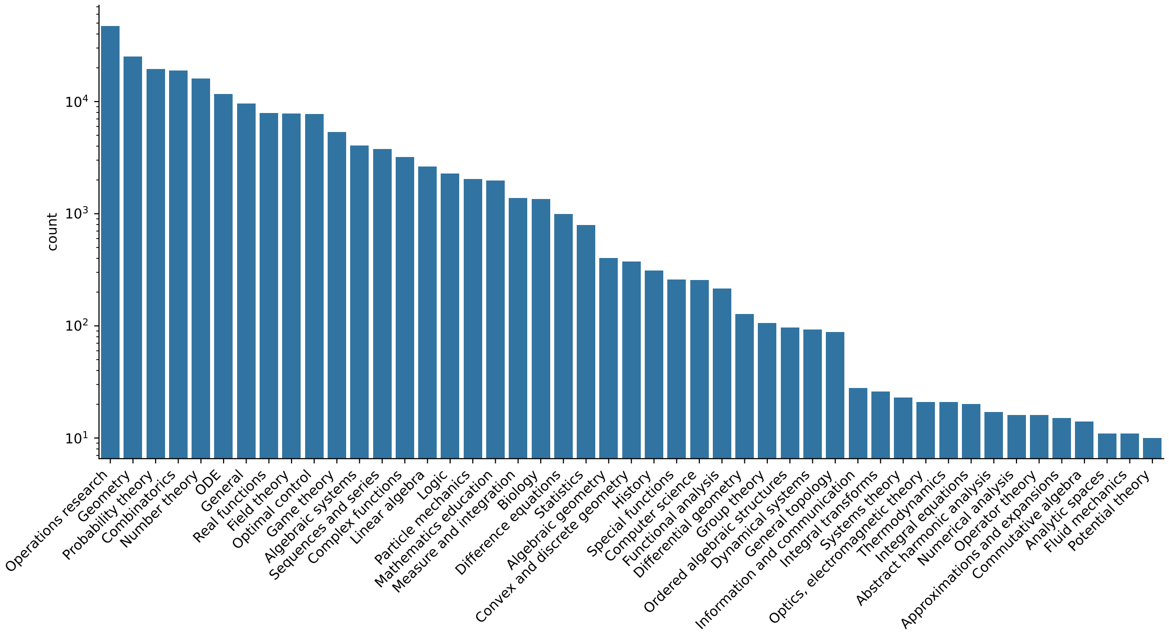


Figure 7: Number of problems per domain, as classified by gpt-4o-mini using math domains as defined by the 2020 Mathematics Subject Classification. Note that the y-axis is log-scale.

Example Problem: Operations Research

Question: A book was sold at a profit of 12%. Had it been sold for \$18 more, a profit of 18% would have been gained. What is the cost price of the book?
Answer: \$300

Example Problem: Geometry

Question: Find the angles at which the parabola $y = x^2 - x$ intersects the x-axis.
Answer: $\frac{3\pi}{4}$ and $\frac{\pi}{4}$

Example Problem: Probability Theory

Question: Ash and Gary independently come up with their own lineups of 15 fire, grass, and water monsters. Then, the first monster of both lineups will fight, with fire beating grass, grass beating water, and water beating fire. The defeated monster is then substituted with the next one from their team's lineup; if there is a draw, both monsters get defeated. Gary completes his lineup randomly, with each monster being equally likely to be any of the three types. Without seeing Gary's lineup, Ash chooses a lineup that maximizes the probability p that his monsters are the last ones standing. Compute p.
Answer: $1 - \frac{2^{15}}{3^{15}}$

Example Problem: Combinatorics

Question: Let $x_1 \leq x_2 \leq \cdots \leq x_{100}$ be real numbers such that $|x_1| + |x_2| + \cdots + |x_{100}| = 1$ and $x_1 + x_2 + \cdots + x_{100} = 0$. Among all such 100-tuples of numbers, the greatest value that $x_{76} - x_{16}$ can achieve is $\frac{m}{n}$, where m and n are relatively prime positive integers. Find $m + n$.
Answer: 841

Example Problem: Number Theory

Question: Suppose that (u_n) is a sequence of real numbers satisfying $u_{n+2} = 2u_{n+1} + u_n$ and that $u_3 = 9$ and $u_6 = 128$. What is u_5?
Answer: 53

Figure 8: Example problems from the top-5 most common domains defined by the 2020 Mathematics Subject Classification.

3.3. Filter Analysis

In this section we provide an analysis of the filtering process we designed to curate Big-Math. See Table 2 for the main results and Appendix A for detailed results. Before looking closely at the filter outputs, it is important to note that our filtering process is intentionally designed to be more strict than is necessary. This conservative approach ensures that Big-Math contains only the types of data that we desire. However, this leaves room for improvements to our filtering process, and expansion to the Big-Math dataset. The filtering set-up proposed in this work serves only as a first step, allowing us to identify areas for improvement to maximize the utility of the source data and develop more precise filters.

In order to best identify where to spend future efforts, we start by identifying the filters which remove the most data: multiple-choice, multi-part, and proof filters.

3.3.1. Multiple Choice Filter

The largest portion of removed data falls under the multiple choice filter ($\sim 18\%$), with over 80% of the multiple choice questions being sourced from the cn_k12 subset. Since this filter affects such a large number of problems, we see this as promising area for improvement in the future, where the recall of filters can be improved to reintroduce some of the open-ended questions that are mistakenly classified as multiple choice. Additionally, we find large discrepancies in the classifications of the regular expression filter compared with the model-based filter, especially for the olympiad and MATH subsets, where the regular expression-based filter removes significantly more data than the model-based filter. A closer look at these data, which are classified as positive by one filter but negative by the other filter, are a prime area for further inspection. Exploration in this area can further improve the recall of the whole filtering system, and may lead to 1,000s of reintroduced problems.

3.3.2. Multi-Part Filter

Similarly, the multi-part filter shows potential for further refinement. Again, we see that the regular expression-based filter consistently removes a larger portion of data than the model-based filter. Specifically, we find that the regular expression-based filter removes 14,000 more questions from Orca-Math, 6,500 more from Olympiads, and 2,700 more from the MATH subset, compared with the model-based filter. Given these large differences, there is a clear opportunity to further improve this filtering step.

3.3.3. Proof Filter

Interestingly, the proof filter has the opposite outcome from the multiple choice and multi-part filters: the model-based filter removes 10,000 more problems than the regular expression-based filter. This is again an area for further investigation to determine how many of those problems can be reintroduced to the dataset, while still fulfilling our desiderata.

3.3.4. Asymptote Filter

One area that would be interesting to study further is a model's ability to use the Asymptote vector graphics language. In our work, we found 625 problems from the HARP subset using this language, and removed them all. However, this is not an insignificant portion of human-written problems, and may be beneficial to include these problems. In contrast to our work, Muennighoff et al. [2025] evaluate a model on AIME24 [of America, 2024] which contains images in the asymptote language. Presumably, training on such problems would be beneficial for this setting, however, they do not train a model on any asymptote images and do not analyze the performance of their model on such questions. Therefore, whether including such problems in the training set is beneficial is still unclear.

4. Future Directions

While this work primarily focuses on the creation and curation of the Big-Mathand Big-Math-Reformulateddatasets, this only represents the starting point for further research. Below we outline several promising directions for exploration.

Using this dataset:

- **Scaling** - Investigate data scaling laws on RL training. In particular, there are a number of training algorithms, and new methods continually developing, and the efficiency of each is unknown [Team et al., 2025]. For example, in very recent studies, Hou et al. [2025] demonstrate that training data and inference time compute scale in a complementary way (increasing inference is most useful when also scaling training data), and Setlur et al. [2025] demonstrate that the gap between methods trained with verifier-based methods and verifier-free methods increases as the quantity of training data increases.
- **Balancing** - Future studies can analyze how to define and balance the quality, diversity, and complexity of a dataset [Havrilla et al., 2024a].
- **Leveraging difficulty** - Our difficulty classifications can be used studies on subset selection for RL, similar to the method proposed by Muennighoff et al. [2025]. In particular, it will be interesting to explore how these difficulty scores translate to other models or model families. Additionally, a potentially fruitful avenue of research is to explore if difficulty scores can be used to form an effective curriculum, where curriculum learning has been effectively applied to RL in the past [Dennis et al., 2021, Jiang et al., 2021, inter alia], but has only successfully been applied to training language models recently [Albalak et al., 2023, Wang et al., 2024]
- **Distillation** - As has been done with other datasets [Liu et al., 2024, Guan et al., 2025, inter alia], Big-Math an also be used in concert with a strong reasoning model to generate a dataset for SFT distillation.

Improving future datasets:

- **Filter and verifier improvements** - As described throughout the paper, our filters are overly strict. This allows for further improvements on our filters, enabling the inclusion of more complex problems. Additionally, throughout this paper, we have assumed the use of a very simple verifier and only retained data that can be extracted based on simple string matching. However, the use of improved answer extraction and verifiers would allow for a much broader coverage of data, including proofs and multi-part questions.
- **Model-based filtering** - Refine and improve the process of designing model-based filters (e.g. prompting approaches). Integrating methods such as LLM-as-a-judge, or similar quality checks, to the model-based filtering can reduce the reliance on human oversight. Overall, the goal would be to develop advanced filters with fewer human-in-the-loop requirements. For instance, can RL training be used to train a filtering model, which in turn improves the RL training for a reasoning model? Probably! There exists huge quantities of positive and negative examples for the filters that we use here. The question then becomes: can the RL training teach a model to filter for new requirements? For example, training the model to identify data where the answer is not consistent with the model's solution. With methods such as Meta-RL or continual learning, this is a very exciting direction.

- **Domain expansion** - Some of the methodology in this work is specific to mathematics (e.g. proof filter), but much of the work is applicable to other domains (e.g. multiple choice and multi-part question filters). Extending the methodologies developed here to other domains is a very meaningful direction of study. A very recent example of generating verifiable question-answer pairs is that from Yuan et al. [2025], which can be used as inspiration.

5. Conclusion

There has been an incredible rise in the number of proposed methodologies on RL to train reasoning models [OpenAI et al., 2024, DeepSeek-AI et al., 2025, Liu et al., 2024, Yuan et al., 2024, Guan et al., 2025], where each method requires a large-scale corpus of high-quality, diverse prompts with verifiable answers. While these recent works have proposed differing methodologies, many of them achieve similar results, suggesting that algorithms may not be the limiting factor in performance but, as has been proposed, data is the limiting factor on performance [Xiang et al., 2025, Yeo et al., 2025].

This work aims to address these issues by developing a large-scale corpus of high-quality, diverse math problems, with verifiable answers, Big-Math. Through our human-in-the-loop filtering process, we bridge the gap between data quality and quantity, leading to a dataset of over 250,000 problems satisfying our three key desiderata: uniquely verifiable solutions, open-ended formulations, and closed-form answers. Furthermore, we introduce Big-Math-Reformulated, demonstrating the potential to reclaim valuable data lost during filtering, contributing over 47,000 high quality problems to the dataset. Through an in-depth analysis, we explored the diversity and difficulty of Big-Math, highlighting it's utility in advancing the state of the art in LLM reasoning capabilities, and pointing out promising directions for the future development of filters.

By releasing Big-Math to the research community, we aim to catalyze progress in RL methodologies and establish a shared benchmark for future advancements in mathematical reasoning with LLMs. This dataset not only supports the development of more generalizable reasoning models but also lays the groundwork for exploring scaling laws, training dynamics, and new RL algorithms in the context of mathematical problem-solving.

References

Amro Abbas, Kushal Tirumala, Dániel Simig, Surya Ganguli, and Ari S. Morcos. Semdedup: Data-efficient learning at web-scale through semantic deduplication, 2023. URL https://arxiv.org/abs/2303.09540.

ServiceNow AI. Slam lab - r1-distill-sft dataset, 2025. URL https://hf.co/ServiceNow-AI/R1-Distill-SFT. Accessed: 2025-02-04.

Alon Albalak, Liangming Pan, Colin Raffel, and William Yang Wang. Efficient online data mixing for language model pre-training, 2023. URL https://arxiv.org/abs/2312.02406.

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. A survey on data selection for language models, 2024. URL https://arxiv.org/abs/2402.16827.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V. Le, Sergey Levine, and Yi Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training, 2025. URL https://arxiv.org/abs/2501.17161.

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

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang.

Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design, 2021. URL https://arxiv.org/abs/2012.02096.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. Omni-math: A universal olympiad level mathematic benchmark for large language models, 2024. URL https://arxiv.org/abs/2410.07985.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. rstar-math: Small llms can master math reasoning with self-evolved deep thinking, 2025. URL https://arxiv.org/abs/2501.04519.

Alex Havrilla, Andrew Dai, Laura O'Mahony, Koen Oostermeijer, Vera Zisler, Alon Albalak, Fabrizio Milo, Sharath Chandra Raparthy, Kanishk Gandhi, Baber Abbasi, Duy Phung, Maia Iyer, Dakota Mahan, Chase Blagden, Srishti Gureja, Mohammed Hamdy, Wen-Ding Li, Giovanni Paolini, Pawan Sasanka Ammanamanchi, and Elliot Meyerson. Surveying the effects of quality, diversity, and complexity in synthetic data from large language models, 2024a. URL https://arxiv.org/abs/2412.02980.

Alex Havrilla, Sharath Raparthy, Christoforus Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, and Roberta Raileanu. Glore: When, where, and how to improve llm reasoning via global and local refinements, 2024b. URL https://arxiv.org/abs/2402.10963.

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, 2021. URL https://arxiv.org/abs/2103.03874.

Zhenyu Hou, Xin Lv, Rui Lu, Jiajie Zhang, Yujiang Li, Zijun Yao, Juanzi Li, Jie Tang, and Yuxiao Dong. Advancing language model reasoning through reinforcement learning and inference scaling, 2025. URL https://arxiv.org/abs/2501.11651.

Minqi Jiang, Edward Grefenstette, and Tim Rocktäschel. Prioritized level replay, 2021. URL https://arxiv.org/abs/2010.03934.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, H'erve J'egou, and Tomas Mikolov. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016a.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016b.

Juno Kim, Denny Wu, Jason Lee, and Taiji Suzuki. Metastable dynamics of chain-of-thought reasoning: Provable benefits of search, rl and distillation, 2025. URL https://arxiv.org/abs/2502.01694.

Bespoke Labs. Bespoke-stratos: The unreasonable effectiveness of reasoning distillation, 2025. URL https://hf.co/bespokelabs/Bespoke-Stratos-32B. Accessed: 2025-01-22.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training, 2024. URL https://arxiv.org/abs/2411.15124.

Dacheng Li, Shiyi Cao, Tyler Griggs, Shu Liu, Xiangxi Mo, Shishir G. Patil, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Llms can easily learn to reason from demonstrations structure, not content, is what matters!, 2025. URL https://arxiv.org/abs/2502.07374.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etash Guha, Sedrick Keh, Kushal Arora, Saurabh Garg, Rui Xin, Niklas Muennighoff, Reinhard Heckel, Jean Mercat, Mayee Chen, Suchin Gururangan, Mitchell Wortsman, Alon Albalak, Yonatan Bitton, Marianna Nezhurina, Amro Abbas, Cheng-Yu Hsieh, Dhruba Ghosh, Josh Gardner, Maciej Kilian, Hanlin Zhang, Rulin Shao, Sarah Pratt, Sunny Sanyal, Gabriel Ilharco, Giannis Daras, Kalyani Marathe, Aaron Gokaslan, Jieyu Zhang, Khyathi Chandu, Thao Nguyen, Igor Vasiljevic, Sham Kakade, Shuran Song, Sujay Sanghavi, Fartash Faghri, Sewoong Oh, Luke Zettlemoyer, Kyle Lo, Alaaeldin El-Nouby, Hadi Pouransari, Alexander Toshev, Stephanie Wang, Dirk Groeneveld, Luca Soldaini, Pang Wei Koh, Jenia Jitsev, Thomas Kollar, Alexandros G. Dimakis, Yair Carmon, Achal Dave, Ludwig Schmidt, and Vaishaal Shankar. Datacomp-lm: In search of the next generation of training sets for language models, 2024a. URL https://arxiv.org/abs/2406.11794.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. Numinamath. [https://huggingface.co/AI-MO/NuminaMath-CoT](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf), 2024b.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023a.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023b. URL https://arxiv.org/abs/2305.20050.

Zihan Liu, Yang Chen, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Acemath: Advancing frontier math reasoning with post-training and reward modeling, 2024. URL https://arxiv.org/abs/2412.15084.

Dakota Mahan, Duy Van Phung, Rafael Rafailov, Chase Blagden, Nathan Lile, Louis Castricato, Jan-Philipp Fränken, Chelsea Finn, and Alon Albalak. Generative reward models, 2024. URL https://arxiv.org/abs/2410.12832.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models, 2024. URL https://arxiv.org/abs/2410.05229.

Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. Orca-math: Unlocking the potential of slms in grade school math, 2024. URL https://arxiv.org/abs/2402.14830.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025. URL https://arxiv.org/abs/2501.19393.

Mathematical Association of America. Aime, February 2024. URL https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions/.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin

Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card, 2024. URL https://arxiv.org/abs/2412.16720.

OpenAI, :, Ahmed El-Kishky, Alexander Wei, Andre Saraiva, Borys Minaev, Daniel Selsam, David Dohan, Francis Song, Hunter Lightman, Ignasi Clavera, Jakub Pachocki, Jerry Tworek, Lorenz Kuhn, Lukasz Kaiser, Mark Chen, Max Schwarzer, Mostafa Rohaninejad, Nat McAleese, o3 contributors, Oleg Mürk, Rhythm Garg, Rui Shu, Szymon Sidor, Vineet Kosaraju, and Wenda Zhou. Competitive programming with large reasoning models, 2025. URL https://arxiv.org/abs/2502.06807.

Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage?, 2023. URL https://arxiv.org/abs/2304.15004.

Amrith Setlur, Nived Rajaraman, Sergey Levine, and Aviral Kumar. Scaling test-time compute without verification or rl is suboptimal, 2025. URL https://arxiv.org/abs/2502.12118.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Harsh

Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew E. Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Pete Walsh, Luke Zettlemoyer, Noah A. Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. Dolma: an open corpus of three trillion tokens for language model pretraining research, 2024. URL https://arxiv.org/abs/2402.00159.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, Haiqing Guo, Han Zhu, Hao Ding, Hao Hu, Hao Yang, Hao Zhang, Haotian Yao, Haotian Zhao, Haoyu Lu, Haoze Li, Haozhen Yu, Hongcheng Gao, Huabin Zheng, Huan Yuan, Jia Chen, Jianhang Guo, Jianlin Su, Jianzhou Wang, Jie Zhao, Jin Zhang, Jingyuan Liu, Junjie Yan, Junyan Wu, Lidong Shi, Ling Ye, Longhui Yu, Mengnan Dong, Neo Zhang, Ningchen Ma, Qiwei Pan, Qucheng Gong, Shaowei Liu, Shengling Ma, Shupeng Wei, Sihan Cao, Siying Huang, Tao Jiang, Weihao Gao, Weimin Xiong, Weiran He, Weixiao Huang, Wenhao Wu, Wenyang He, Xianghui Wei, Xianqing Jia, Xingzhe Wu, Xinran Xu, Xinxing Zu, Xinyu Zhou, Xuehai Pan, Y. Charles, Yang Li, Yangyang Hu, Yangyang Liu, Yanru Chen, Yejie Wang, Yibo Liu, Yidao Qin, Yifeng Liu, Ying Yang, Yiping Bao, Yulun Du, Yuxin Wu, Yuzhi Wang, Zaida Zhou, Zhaoji Wang, Zhaowei Li, Zhen Zhu, Zheng Zhang, Zhexu Wang, Zhilin Yang, Zhiqi Huang, Zihao Huang, Ziyao Xu, and Zonghan Yang. Kimi k1.5: Scaling reinforcement learning with llms, 2025. URL https://arxiv.org/abs/2501.12599.

Jiachen (Tianhao) Wang, Tong Wu, Dawn Song, Prateek Mittal, and Ruoxi Jia. Greats: Online selection of high-quality data for llm training in every iteration. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 131197–131223. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ed165f2ff227cf36c7e3ef88957dadd9-Paper-Conference.pdf.

Yu Wang, Nan Yang, Liang Wang, and Furu Wei. Examining false positives under inference scaling for mathematical reasoning, 2025. URL https://arxiv.org/abs/2502.06217.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

Violet Xiang, Charlie Snell, Kanishk Gandhi, Alon Albalak, Anikait Singh, Chase Blagden, Duy Phung, Rafael Rafailov, Nathan Lile, Dakota Mahan, Louis Castricato, Jan-Philipp Franken, Nick Haber, and Chelsea Finn. Towards system 2 reasoning in llms: Learning how to think with meta chain-of-thought, 2025. URL https://arxiv.org/abs/2501.04682.

Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning in llms, 2025. URL https://arxiv.org/abs/2502.03373.

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. Free process rewards without process labels, 2024. URL https://arxiv.org/abs/2412.01981.

Weizhe Yuan, Jane Yu, Song Jiang, Karthik Padthe, Yang Li, Dong Wang, Ilia Kulikov, Kyunghyun Cho, Yuandong Tian, Jason E Weston, and Xian Li. Naturalreasoning: Reasoning in the wild with 2.8m challenging questions, 2025. URL https://arxiv.org/abs/2502.13124.

Albert S. Yue, Lovish Madaan, Ted Moskovitz, DJ Strouse, and Aaditya K. Singh. HARP: A challenging human-annotated math reasoning benchmark, 2024. URL https://github.com/aadityasingh/HARP.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction, 2024. URL https://arxiv.org/abs/2408.15240.

A. Filter Statistics

	Multiple Choice		True/False		Yes/No		Multi-Part		Proof		Semantic	Total
Subset	(R)	(L)	(R)	(L)	(R)	(L)	(R)	(L)	(R)	(L)	**Deduplication**	
Orca-Math	1174	111	5	0	12	0	16345	2298	2	5	8286	26,487
cn_k12	94888	94932	420	0	384	0	20358	19428	3969	5185	2098	123,809
olympiads	12123	9507	311	0	4615	0	13509	6981	23210	30315	1362	53,102
MATH	1046	386	0	0	0	3	2734	50	0	31	7694	9,714
aops_forum	1588	1162	85	0	126	0	1508	1488	2282	4605	262	7,853
GSM8k	45	1	0	0	0	0	782	3	0	0	7664	7,999
HARP	92	215	2	0	7	0	253	44	0	2	155	695
Omni-MATH	125	29	7	122	191	169	283	212	3	207	378	1,182
amc_aime	2955	2938	0	0	22	0	1018	7	36	63	53	3,066
Total	114,036	109,281	830	122	5,357	172	49,932	30,511	29,502	40,413	27,952	233,907

Statistics of the individual filters used to create Big-Math. (R) columns are regular expression filters, while (L) columns are the result of filtering with Llama-3.1-70B.

B. Domains By Source

Domain Source	Abstract Algebra	Algebra	Calculus	Differential Equations	Discrete Mathematics	Geometry	Intermediate Algebra	Linear Algebra	Math Word Problems	Number Theory	Other	Prealgebra	Precalculus	Statistics
amc_aime	0	12	0	0	12	18	2	0	14	8	3	0	0	10
aops_forum	42	1263	212	3	848	1218	27	19	495	1065	174	4	39	458
Big-Math-Reformulated	17	14362	2725	2	1499	8207	3971	1600	4534	1219	939	1059	2629	5186
cn_k12	17	23282	6169	2	1757	12813	4324	1563	3033	774	1039	970	5022	4998
GSM8k	0	17	0	0	0	12	0	0	1810	0	0	11	0	15
HARP	1	677	21	0	123	690	101	4	716	250	36	50	28	340
MATH	12	2752	175	0	102	1247	532	282	992	1145	113	258	431	1018
olympiads	59	6458	1603	101	3347	9117	456	156	5744	2859	780	116	1029	2324
Omni-MATH	11	547	59	0	334	479	38	7	259	355	78	42	24	301
Orca-Math	0	4689	29	1	111	3102	241	3	66563	2285	333	3091	11	2843

Breakdown of problems by source and domains (as defined by Gao et al. [2024]).

C. Reformulation Examples

Successful Reformulation

Original Problem: What is the sum of the prime factors of 2010?
(A) 67
(B) 75
(C) 77
(D) 201
(E) 210

Key Information Extraction:

core_mathematical_concept: "Number theory - prime factorization and sum of prime factors"
key_information_extraction: ["Find the sum of prime factors of 2010", "Prime factorization required"]
problem_structure_analysis: "Direct calculation problem"
multiple_choice_removal_strategy: ["Remove all answer choices", "Ask for direct calculation of sum of prime factors"]
rephrasing_approach: ["Request the sum of prime factors directly", "No comparative aspect needed"]
problem_integrity_preservation: ["Maintain the original number (2010)", "Require prime factorization"]
answer_format_specification: ["Answer should be a single integer", "No units or decimal places required"]
is_multiple_choice: true

Reformulation: Find the sum of the prime factors of 2010. Express your answer as a single integer.

Judgement: The reformulated problem is a direct and clear request to find the sum of the prime factors of 2010. It does not provide any multiple-choice options, nor does it limit the answer choices in any way. The problem requires the solver to find the prime factorization of 2010 and then sum the prime factors, which is a well-defined mathematical task.

The original problem and the reformulated problem are mathematically equivalent, and the solution to the original problem is still applicable to the reformulated problem. The reformulated problem does not introduce any new mathematical concepts or difficulties, and it does not provide any additional information that would make the problem easier to solve.

The answer format specification is clear and unambiguous, requiring the solver to express the answer as a single integer. This is a suitable format for the problem, as the sum of the prime factors is a well-defined integer value.

Overall, the reformulated problem is a well-posed and mathematically sound problem that requires the solver to apply mathematical concepts and techniques to find the solution.

Failed Reformulation

Original Problem: Let x_1, x_2 be distinct positive real numbers and a be a real number in the interval $(0, 1)$. Define $y_1 = \frac{x_1}{1+a} + \frac{ax_2}{1+a}$ and $y_2 = \frac{x_2}{1+a} + \frac{ax_1}{1+a}$. Determine the relationship between x_1x_2 and y_1y_2:
(A) $x_1 \cdot x_2 > y_1y_2$
(B) $x_1x_2 = y_1y_2$
(C) $x_1x_2 < y_1y_2$
(D) Cannot be determined, it depends on the value of a

Key Information Extraction:

core_mathematical_concept: "Inequalities involving real numbers and their products"
key_information_extraction: ["Distinct positive real numbers x1, x2", "Real number a in the interval (0,1)", "Expressions for y1 and y2 in terms of x1, x2, and a", "Need to compare the product x1*x2 with y1*y2"]
problem_structure_analysis: "Comparison problem requiring algebraic manipulation and inequality analysis"
multiple_choice_removal_strategy: ["Remove all comparisons and answer choices", "Ask for direct determination of the relationship between x1*x2 and y1*y2"]
rephrasing_approach: ["Keep the expressions for y1 and y2 intact", "Request the determination of the relationship between x1*x2 and y1*y2", "Specify the possible relationships (>, <, =)"]
problem_integrity_preservation: ["Maintain all original expressions and conditions", "Remove comparative aspect entirely"]
answer_format_specification: ["Answer should be expressed as one of the following: >, <, or =", "Include the relationship between x1*x2 and y1*y2 in the answer"]
is_multiple_choice: true

Reformulation: Given distinct positive real numbers x_1, x_2 and a real number a in the interval (0,1), define $y_1 = x_1/(1 + a) + a * x_2/(1 + a)$ and $y_2 = x_2/(1 + a) + a * x_1/(1 + a)$. Determine the relationship between $x_1 * x_2$ and $y_1 * y_2$. Express your answer as one of the following: $>$, $<$, or $=$.

Judgement: The reformulated problem still implies a multiple-choice format with the options $>$, $<$, or $=$, which may limit the answer choices and does not fully open the problem to exploration.

Figure 9: **Examples of the reformulation process** starting from the original problem, we extract key information, and then use that information to reformulate the problem as an open-ended question. Finally, a critical evaluation of the reformulation determines whether the reformulation succeeded. We include an example of both a successful reformulation (top) and failed reformulation (bottom).

D. Prompts

D.1. Llama-3.1-70B Filter prompts

Multiple Choice

Given this question: {problem}
Is this a multiple choice question (a question that provides specific options to choose from, typically labeled as A, B, C, D or 1, 2, 3, 4)? Return only "yes" or "no" without any additional explanation.

D.2. Reformulation Prompts

Proof

Given this question: {problem}
Is this a mathematical proof question (a question that asks to prove a statement, theorem, or property...)? Examples of proof indicators:
- "Prove that..."
- "Show that..."
- "Demonstrate why..."
- "Justify your answer..."
- "Explain why..." Here are examples of proof questions:

Example 1: Given positive integers a and b such that $b > a > 1$, and a does not divide b, and a given sequence of positive integers $\{b_n\}_{n=1}^{\infty}$ satisfying $b_{n+1} \geq 2b_n$ for all positive integers n. Does there always exist a sequence of positive integers $\{a_n\}_{n=1}^{\infty}$ such that for all positive integers n, $a_{n+1} - a_n \in \{a, b\}$, and for all positive integers m and l (which can be the same), $a_m + a_l \notin \{b_n\}$ for all n?

Example 2: Let $f(x) = x^n, x \in D, n \in \mathbf{N}^+$. Determine whether $f(x)$ is a solution to the functional inequality

$$f(x) + f(1-x) > 1$$

If so, find the domain D; if not, provide an explanation.

Example 3: In a right angled-triangle ABC, $\angle ACB = 90^o$. Its incircle O meets BC, AC, AB at D,E,F respectively. AD cuts O at P. If $\angle BPC = 90^o$, prove $AE + AP = PD$.

Example 4: A(x,y), B(x,y), and C(x,y) are three homogeneous real-coefficient polynomials of x and y with degree 2, 3, and 4 respectively. we know that there is a real-coefficient polinimial R(x,y) such that $B(x,y)^2 - 4A(x,y)C(x,y) = -R(x,y)^2$. Show that there exist 2 polynomials F(x,y,z) and G(x,y,z) such that $F(x,y,z)^2 + G(x,y,z)^2 = A(x,y)z^2 + B(x,y)z + C(x,y)$ if for any x, y, z real numbers $A(x,y)z^2 + B(x,y)z + C(x,y) \geq 0$

Example 5: Prove

$$\frac{1}{\cos 0^\circ \cos 1^\circ} + \frac{1}{\cos 1^\circ \cos 2^\circ} + \cdots + \frac{1}{\cos 88^\circ \cos 89^\circ} = \frac{\cos 1^\circ}{\sin^2 1^\circ}.$$

Here are examples of non-proof questions:

Example 1: In a 100×25 rectangular table, each cell is filled with a non-negative real number. The number in the i-th row and j-th column is denoted by $x_{i,j}$ $(i = 1, 2, \ldots, 100; j = 1, 2, \ldots, 25)$ (Table 1). The numbers in each column of Table 1 are then reordered in descending order to create Table 2 such that $x'_{1,j} \geq x'_{2,j} \geq \cdots \geq x'_{100,j}$ $(j = 1, 2, \ldots, 25)$. Find the smallest natural number k such that if the numbers in Table 1 satisfy $\sum_{j=1}^{25} x_{i,j} \leq 1$ $(i = 1, 2, \ldots, 100)$, then for $i \geq k$, Table 2 satisfies $\sum_{j=1}^{25} x'_{i,j} \leq 1$ $(i = 1, 2, \ldots, 100)$.

Example 2: We are given $2n$ natural numbers

$$1, 1, 2, 2, 3, 3, \ldots, n-1, n-1, n, n.$$

Find all n for which these numbers can be arranged in a row such that for each $k \leq n$, there are exactly k numbers between the two numbers k.

Example 3: Determine all positive integers n, $n \geq 2$, such that the following statement is true: If $(a_1, a_2, ..., a_n)$ is a sequence of positive integers with $a_1 + a_2 + \cdots + a_n = 2n - 1$, then there is block of (at least two) consecutive terms in the sequence with their (arithmetic) mean being an integer.

Example 4: Turbo the snail sits on a point on a circle with circumference 1. Given an infinite sequence of positive real numbers $c_1, c_2, c_3, \ldots$, Turbo successively crawls distances $c_1, c_2, c_3, \ldots$ around the circle, each time choosing to crawl either clockwise or counterclockwise. Determine the largest constant $C > 0$ with the following property: for every sequence of positive real numbers $c_1, c_2, c_3, \ldots$ with $c_i < C$ for all i, Turbo can (after studying the sequence) ensure that there is some point on the circle that it will never visit or crawl across.

Example 5: For an even integer positive integer n Kevin has a tape of length $4n$ with marks at $-2n, -2n + 1, \ldots, 2n-1, 2n$. He then randomly picks n points in the set $-n, -n+1, -n+2, \ldots, n-1, n$, and places a stone on each of these points. We call a stone 'stuck' if it is on $2n$ or $-2n$, or either all the points to the right, or all the points to the left, all contain stones. Then, every minute, Kevin shifts the unstuck stones in the following manner: He picks an unstuck stone uniformly at random and then flips a fair coin. If the coin came up heads, he then moves that stone and every stone in the largest contiguous set containing that stone one point to the left. If the coin came up tails, he moves every stone in that set one point right instead. He repeats until all the stones are stuck. Let p_k be the probability that at the end of the process there are exactly k stones in the right half. Evaluate $\frac{p_{n-1} - p_{n-2} + p_{n-3} - \ldots + p_3 - p_2 + p_1}{p_{n-1} + p_{n-2} + p_{n-3} + \ldots + p_3 + p_2 + p_1}$ in terms of n.

Example 6: A 0-1 sequence of length 2^k is given. Alice can pick a member from the sequence, and reveal it (its place and its value) to Bob. Find the largest number s for which Bob can always pick s members of the sequence, and guess all their values correctly. Alice and Bob can discuss a strategy before the game with the aim of maximizing the number of correct guesses of Bob. The only information Bob has is the length of the sequence and the member of the sequence picked by Alice.

Return only "yes" or "no" without any additional explanation.

Yes/No

Given this question: {problem} Is this a yes/no question (a question that asks to choose between two options, typically labeled as yes or no)? Return only "yes" or "no" without any additional explanation.

True/False

Given this question: {problem} Is this a true/false question (a question that asks to choose between two options, typically labeled as true or false)? Return only "true" or "false" without any additional explanation.

Multiple Part

Your task is to determine if the given question contains multiple sub-questions, sub-parts, or sub-tasks. A multi-part question requires separate answers for different components, rather than a single comprehensive answer. Besides that, if the question is multiple choice and only requires to select one option, it is not a multi-part question.
Here are examples of multi-part questions that require multiple distinct answers:
Example 1: Given the set $M = \{0, 1\}$, $A = \{(x, y)|x \in M, y \in M\}$, $B = \{(x, y)|y = -x + 1\}$. 1. Please list the elements of set A. 2. Find $A \cap B$ and list all subsets of $A \cap B$.
Example 2: In the Cartesian coordinate system xOy, the parametric equation of curve C_1 is $\begin{cases} x = \cos\theta \\ y = 1 + \sin\theta \end{cases}$ (where θ is the parameter), and the equation of curve C_2 is $\frac{x^2}{1} + \frac{y^2}{2} = 1$. With O as the pole and the non-negative half-axis of x as the polar axis, a polar coordinate system is established with the same unit of length as the Cartesian coordinate system xOy. (1) Find the polar equations of curves C_1 and C_2; (2) The ray $\theta = \frac{\pi}{3}(\rho > 0)$ intersects curve C_1 at point A (other than the pole) and intersects curve C_2 at point B. Find $|AB|$.
Example 3: Given the function $f(x) = |x+2| - |2x-a|$, $(a \in \mathbb{R})$. (I) When $a = 3$, solve the inequality $f(x) > 0$; (II) When $x \in [0, +\infty)$, $f(x) < 3$ always holds, find the range of a.
Example 4: Given an ellipse C_1: $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$ $(a > b > 0)$ with a major axis length of 4 and an eccentricity of $\frac{1}{2}$, where F_1 and F_2 are its left and right foci, respectively. A moving circle passes through point F_2 and is tangent to the line $x = -1$. (I) (i) Find the equation of the ellipse C_1; (ii) Find the equation of the trajectory of the center C of the moving circle; (II) On the curve C, there are two points M and N, and on the ellipse C_1, there are two points P and Q, satisfying that MF_2 and $\overrightarrow{NF_2}$ are collinear, $\overrightarrow{PF_2}$ and $\overrightarrow{QF_2}$ are collinear, and $\overrightarrow{PF_2} \cdot \overrightarrow{MF_2} = 0$, find the minimum value of the area of quadrilateral $PMQN$.
Example 5: In the rectangular coordinate system xOy, a polar coordinate system is established with the coordinate origin as the pole and the positive semi-axis of the x-axis as the polar axis. The polar coordinate equation of circle C is $\rho^2 - 2m\rho\cos\theta + 4\rho\sin\theta = 1 - 2m$. (1) Find the rectangular coordinate equation of C and its radius. (2) When the radius of C is the smallest, the curve $y = \sqrt{3}|x - 1| - 2$ intersects C at points A and B, and point $M(1, -4)$. Find the area of $\triangle MAB$.
Here are examples of single-part questions that require only one answer:
Example 1: Ancient Greek mathematicians from the Pythagorean school studied various polygonal numbers, such as triangular numbers 1, 3, 6, 10, ..., with the n-th triangular number being $\frac{n(n+1)}{2} = \frac{1}{2}n^2 + \frac{1}{2}n$. Let the n-th k-sided polygon number be denoted as $N(n, k)$ $(k \geq 3)$. Below are the expressions for the n-th number of some k-sided polygon numbers:
Triangular numbers $N(n, 3) = \frac{1}{2}n^2 + \frac{1}{2}n$
Square numbers $N(n, 4) = n^2$
Pentagonal numbers $N(n, 5) = \frac{3}{2}n^2 - \frac{1}{2}n$
Hexagonal numbers $N(n, 6) = 2n^2 - n$
...
From this, we can deduce the expression for $N(n, k)$ and calculate $N(8, 12) =$ _________.'
Example 2: Let x be the dividend, y be the divisor, z be the quotient, and r be the remainder. If $y = 3(y_1 + y_2) + 4, z = 2z_1^2 - z_2, r = 3r_1 + 2$, and $x = 2x_1y_1 - x_2 + 10$, find the values of x, y, z, and r, given that $y_1 = 2, y_2 = 3, z_1 = 3, z_2 = 5, r_1 = 1, x_1 = 4$, and $x2 = 6$.
Example 3: If $x_1, x_2, \ldots, x_n$ are positive real numbers with $x_1^2 + x_2^2 + \ldots + x_n^2 = 1$, find the minimum value of $\sum_{i=1}^{n} \frac{x_i^5}{x_1+x_2+\ldots+x_n-x_i}$.
Example 4: Given that the value of the function $f(x) = \frac{1}{(x-2)^2} - 2x + \cos 2\theta - 3\sin\theta + 2$ is always positive for $x \in (-\infty, 2)$, determine the range of the parameter θ within the interval $(0, \pi)$.
Example 5: A transparent, sealed cubic container is exactly half filled with water. When this cube is rotated arbitrarily, the shape of the water surface inside the container can be: (1) triangle; (2) rectangle; (3) square; (4) regular hexagon. Among these, the correct conclusions are _______________.

Llama-3.1-8B Rollout Prompt

You are a math expert. Given the following math problem, provide your solution in Latex format. Always format your final answer in perfect LaTeX \boxed{final_answer} format.

Llama-3.1-405B Rollout Prompt (part 1)

From this moment forward, adopt the persona of a tenured mathematics professor. You receive math problems and you solve them, step by step, following the formatting instructions below.

# Mathematical Solution Formatting Guidelines

## Mathematical Notation Rules

Use $$ $$ for:
- Standalone equations
- Complex mathematical expressions
- Multi-line equations
- Final answers

Use single $ $ for:
- Simple inline expressions
- Variables
- Numbers with mathematical meaning
- Parts of expressions being discussed

For equations with multiple lines:
‘latex
$$\\begin{aligned}
equation1 &= expression1 \\\\
equation2 &= expression2
\\end{aligned}$$
‘

Place the final answer inside \boxed{answer_here}
After solving and verifying your full solution, write the final answer in \\boxed{answer} notation.

## Thought Structure
Each thought must:
1. Begin with active voice ("I" statements or direct observations)
2. Express exactly ONE logical step
3. Integrate mathematical notation naturally within sentences
4. Use appropriate notation based on context
5. No more than 1 sentence + interleaved math notation long

### Example: "I can see that when $x = 2$, the expression $y = x^2 + 3x + 1$ becomes $y = 4 + 6 + 1 = 11$."

## Common Mathematical Patterns
- Use $$ $$ for equations being solved
- Use $ $ for discussing components: "where $m$ is the slope"
- Keep expressions inline when discussing specific parts
- Use block notation ($$) for key steps and results

## Visual Formatting
- Two blank lines between thoughts (\n \n)
- No bullet points or numbered lists
- No section headers within the solution
- Mathematical expressions should flow naturally within sentences
- No explicit step labeling
- Human-style reasoning

—

Llama-3.1-405B Rollout Prompt (part 2)

## Examples

### Problem 1:

A standard die is rolled six times. What is the probability that the product of all six rolls is odd? Express your answer as a common fraction.

### Solution 1:

I need to find the probability by dividing favorable outcomes by total outcomes.

I know that the total number of possible outcomes when rolling a die six times is 6^6.

For the product to be odd, each individual roll must be odd.

Looking at a standard die, the odd numbers are 1, 3, and 5.

Therefore, for each roll, I have only 3 choices to maintain an odd product.

The total number of favorable outcomes is 3^6 since I have 3 choices for each of the 6 rolls.

The probability is thus $\frac{3^6}{6^6}$.
This simplifies to $\left(\frac{1}{2}\right)^6 =$ \\boxed{$\frac{1}{64}$}.

### Problem 2:

Solve over the integers: $2^a + 4^b + 8^c = 328$
### Solution 2:

I notice that all terms are powers of 2, so I can rewrite everything with base 2.
I know that $4 = 2^2$ and $8 = 2^3$, so the equation becomes $2^a + 2^{2b} + 2^{3c} = 328$.
To solve this, I can decompose 328 into powers of 2.
The largest power of 2 less than 328 is 256 (2^8).
Subtracting 256 from 328 leaves 72.
The largest power of 2 less than 72 is 64 (2^6).
Subtracting 64 leaves 8 (2^3).
Therefore, $328 = 2^8 + 2^6 + 2^3$.

Llama-3.1-405B Rollout Prompt (part 3)

Comparing terms: $2^a = 2^8$, $2^{2b} = 2^6$, and $2^{3c} = 2^3$.
Solving these equations: $a = 8$, $b = 3$, and $c = 1$.
My final answer is \\boxed{(a, b, c) = (8, 3, 1)}

### Problem 3:

Find the equation of the circle which passes through $(2, 3)$ and $(4, 5)$ and whose center lies on the straight line $y - 4x + 3 = 0$.

### Solution 3:

I know that the center lies on the line, so I can write its coordinates as $(x, 4x - 3)$.
The radius can be found using the distance from the center to $(2, 3)$.
Using the distance formula, I get $r^2 = (x - 2)^2 + (4x - 3 - 3)^2 = (x - 2)^2 + (4x - 6)^2$.
Expanding and simplifying: $r^2 = 17x^2 - 52x + 40$.
Since the circle also passes through $(4, 5)$, I can write $r^2 = (x - 4)^2 + (4x - 3 - 5)^2$.
This simplifies to $r^2 = 17x^2 - 72x + 80$.
Since both expressions equal r^2, I can write $17x^2 - 52x + 40 = 17x^2 - 72x + 80$.

Llama-3.1-405B Rollout Prompt (part 4)

I can now simplify: $20x = 40$, so $x = 2$.
The center is therefore $(2, 4(2) - 3) = (2, 5)$.
The radius squared is $r^2 = 17(2)^2 - 52(2) + 40 = 16$, so $r = 4$.
The equation of the circle is $(x - 2)^2 + (y - 5)^2 = 16$.
The final answer is the full equation of the circle, so the correct final answer is \\boxed{$(x-2)^2+(y-5)^2 = 16$}

Multiple Choice Reformulation (Part 1)

You are an AI assistant specializing in transforming multiple-choice math problems into open-ended, solvable questions suitable for an automatic grading system that relies on regex pattern matching within ‘$\boxed{}$‘. Your task is to reformulate the given problem while adhering to specific guidelines.

Rules:
- Never turn problems into proofs, all problems should have an answer that is machine-verifiable
- Reformulated problems should have the same final answer as the original problem (just not in the multiple-choice format)
- It must be reasonably clear from the problem what the student is expected to write inside of $\boxed{final\_answer}$. E.g. "What are its key properties and characteristics?" is almost always a bad reformulation because it is ambiguous what the student should write inside of the ‘$\boxed{}$‘. "Express your answer in centimeters." is a good answer format specification because it does NOT give away the answer but does specify a clear format.
- Do NOT reformulate problems that are not actually multiple choice or are impossible to reformulate
- If a problem is not multiple choice, return "N/A" for the reformulated problem and other "N/A" fields

Follow these steps:

1. **Analyze the original problem:**
- Identify the core mathematical concept.
- Note any crucial information, including numerical values, equations, and key terms.
- Determine the structure of the problem (e.g., scenario-based, direct question, multi-step).
- Preserve all mathematical notations, symbols, and formatting as they appear in the original problem.

2. **Remove multiple-choice options:**
- Eliminate all answer choices and their labels (A, B, C, etc.).
- If any options contain information essential to solving the problem, incorporate that information into the main problem statement.

3. **Rephrase the question:**
- Transform the question into an open-ended format that requires a direct numerical or algebraic answer.
- Ensure the rephrased question is clear, unambiguous, and uses language appropriate for the student’s level.

4. **Maintain problem integrity:**
- Keep all original numerical values, equations, figures, and key terms intact.
- Preserve any scenarios, dialogues, or conditional information crucial for solving the problem.
- Do not introduce new information, alter units of measurement, or change the mathematical intent of the problem.
- If the problem references diagrams or figures, ensure that any necessary descriptions are included.

5. **Specify the answer format:**
- Instruct the student to provide their answer using the ‘$\boxed{}$‘ format.
- Do not include placeholders like ‘$\boxed{N}$‘; instead, guide the student to input their calculated answer within the boxed format.
- For example, "Provide your answer in the form ‘$\boxed{\text{your answer here}}$‘."

6. **Final check:**
- Ensure the reformulated problem contains all necessary information for independent solving.
- Verify that the problem hasn’t become easier or harder than the original.
- Check for any common errors, such as unit inconsistencies or typographical mistakes.
- Confirm that no hints, solution methods, or additional explanations have been inadvertently added.

7. **Is actually multiple choice:**
- Some problems are not actually multiple choice and do NOT actually need to be reformulated.
- If the problem is **NOT** multiple choice, do NOT reformulate it! Note here if it is not multiple choice and return "N/A" for the reformulated problem.

Before providing the final reformulated problem, please create a ‘reformulation_process‘ dictionary (for internal use; do not include this dictionary or its content in the final problem). The dictionary should have the following exact keys:

- ‘"core_mathematical_concept"‘: Summarize the core mathematical concept.
- ‘"key_information_extraction"‘: List key information (numerical values, equations, terms).
- ‘"problem_structure_analysis"‘: Identify the problem structure.
- ‘"multiple_choice_removal_strategy"‘: Plan how to remove multiple-choice options.
- ‘"rephrasing_approach"‘: Outline the rephrasing strategy.

Multiple Choice Reformulation (Part 2)

```
- ‘"problem_integrity_preservation"‘: Note how to maintain problem integrity.
- ‘"answer_format_specification"‘: Specify the answer format.
- ‘"is_multiple_choice"‘: Whether the problem is actually multiple choice.

**This structured approach will help ensure accuracy and completeness in your final output.**

**After your reformulation process, present the reformulated problem as plain text, making sure it meets all
the criteria outlined above. Do not include the ‘<reformulation_process>‘ tags or any of their content in the
final problem presented to the student.**

**Examples:**
“‘
<example>
<problem>
Three squares with the same center and corresponding parallel sides are drawn. The distance between the sides
of successive squares is 3 units, and the side length of the largest square is 22 units. What is the perimeter of
the smallest square?
(A) 40
(B) 64
(C) 100
(D) 10
(E) 20
</problem>

<reformulation_process>

"core_mathematical_concept": "Geometry - concentric squares and their dimensions",
"key_information_extraction": [
"Three concentric squares",
"Distance between sides of successive squares: 3 units",
"Side length of the largest square: 22 units",
"Need to find the perimeter of the smallest square"
],
"problem_structure_analysis": "Direct question based on given information",
"multiple_choice_removal_strategy": [
"Eliminate all answer choices and their labels",
"No essential information is lost by removing the options"
],
"rephrasing_approach": [
"Keep the description of the squares intact",
"Rephrase the question to ask for the perimeter in an open-ended format",
"Specify units and number format requirements"
],
"problem_integrity_preservation": [
"Preserve all original numerical values and key terms", "Do not add any new information or hints"
],
"answer_format_specification": [
"Answer should be expressed in units",
"Answer should be given as a whole number",
"No decimal places required based on original options"
],
"is_multiple_choice": true

</reformulation_process>

<reasoning>
Let me think about this step by step:

1. Analyzing the multiple choice aspect:
- The options are all whole numbers: 40, 64, 100, 10, 20
- No units are specified in the options but units are used in the problem
- The options suggest an exact answer is expected, not an approximation

2. Looking at the essential information:
- The geometric relationships are clearly stated
- All measurements are given in units
- No complex transformations or unit conversions needed
```

Multiple Choice Reformulation (Part 3)

3. What needs to be preserved:
- The concentric squares description
- The 3-unit distance between squares
- The 22-unit largest square side length
- The focus on perimeter rather than area

4. What needs to be clarified:
- The answer should include units
- A whole number is expected
- No decimal places needed

5. How to make it open-ended:
- Remove all multiple choice options
- Keep the geometric scenario exactly as is
- Add specific answer format requirements
- Maintain the same level of difficulty

6. Final considerations:
- The reformulated question should be solvable without reference to the original options
- The specifications should ensure an unambiguous answer format
- The mathematical challenge should remain unchanged

</reasoning>

<reformulated_problem>
Three squares with the same center and corresponding parallel sides are drawn. The distance between the sides of each successive square is 3 units, and the side length of the largest square is 22 units. Find the perimeter of the smallest square. Express your answer in units as a whole number.
</reformulated_problem>
</example>
“‘

“‘
<example>
<problem>
The teacher picked 9 out of 13 cards labeled with numbers 1 to 13 and stuck them on the foreheads of 9 students. Each student can see the numbers on the other 8 students’ foreheads but not their own. The teacher asked, "Those who know the number of divisors of their own number, please raise your hand." Two students raised their hands. After they put their hands down, three students had the following conversation.
- A: "I know my number now."
- B: "I still don’t know what my number is, but I have figured out whether it is odd or even."
- C: "My number is 2 less than B’s number and 1 more than A’s number."

What is the sum of the numbers on the four cards that were not picked?
</problem>

<reformulation_process>

"core_mathematical_concept": "Number theory and logical deduction with combinatorics",
"key_information_extraction": [
"13 cards numbered from 1 to 13",
"9 cards picked and assigned to students",
"Students can see others’ numbers but not their own",
"Two students know their number based on number of divisors",
"Conversation between students A, B, and C revealing relationships",
"Need to find sum of the 4 unpicked cards"
],
"problem_structure_analysis": "Complex logical deduction problem combining number theory with information from multiple stages",
"multiple_choice_removal_strategy": "Not applicable as this is not a multiple choice question",
"rephrasing_approach": "Not needed as problem is already in appropriate format",
"problem_integrity_preservation": "No modifications needed",
"answer_format_specification": "Answer should be expressed as a single integer (sum of the unpicked cards)",
"is_multiple_choice": false

</reformulation_process>

Multiple Choice Reformulation (Part 4)

<reasoning>
Let me think about this step by step:

1. First, assess if reformulation is needed:
- This is not a multiple choice question
- It already asks for a specific numerical answer
- The format is clear and unambiguous

2. Analyzing the problem structure:
- It’s a complex logic puzzle
- All information provided is necessary
- The sequence of events matters
- The dialogue provides crucial information

3. Consider what might need clarification:
- Answer format is implicitly clear (a sum)
- No unit specification needed
- No decimal places involved (working with integers 1-13)
- No ambiguity in what constitutes an answer

4. Check for potential improvements:
- Current format is optimal for the problem type
- Adding structure might make it harder to follow
- The narrative flow is important for understanding

5. Final decision:
- No reformulation needed
- Problem is already in ideal open-ended format
- Answer requirements are clear (single integer sum)
</reasoning>

<reformulated_problem>
N/A
</reformulated_problem>
</example>
“‘

“‘
<example>
<problem>
In the cube shown, L and K are midpoints of adjacent edges AD and AB, respectively. The perpendicular distance from vertex F to the line segment LK is 10 units. Which of the following measurements is closest to the volume of the cube after subtracting 3 units?

(A) 200 in^3
(B) 500 cm^3
(C) 0.01039 m^3
(D) Tie
</problem>

<reformulation_process>
{
"core_mathematical_concept": "3D geometry involving cube volume calculation",

"key_information_extraction": [
"Points L and K are midpoints of adjacent edges AD and AB",
"Perpendicular distance from F to LK is 10 units",
"Need to subtract 3 units from final volume"
],

"problem_structure_analysis": "Geometry problem requiring volume calculation",

"multiple_choice_removal_strategy": [
"Remove all comparisons",
"Ask for direct volume calculation"
],

Multiple Choice Reformulation (Part 5)

"rephrasing_approach": [
"Keep geometric relationships intact",
"Request exact volume calculation",
"Specify units and precision"
],

"problem_integrity_preservation": [
"Maintain all numerical values and geometric relationships",
"Remove comparative aspect entirely"
],

"answer_format_specification": [
"Answer should be expressed in cubic meters (m^3)",
"Round to 5 decimal places",
"Include units in answer"
],

"is_multiple_choice": true
}
</reformulation_process>
<reasoning>
Let me think about this step by step:

1. First, I need to identify what makes this multiple choice:

- The core task is actually calculating a specific volume
- The multiple choice aspect is just about comparing to different unit measurements
- The comparison isn't fundamental to the mathematical concept being tested

2. Looking at the original answer choices:

- They use different units (in^3, cm^3, m^3)
- The m^3 value has 5 decimal places (0.01039)
- This suggests m^3 with 5 decimal places is an appropriate precision level

3. How to remove the multiple choice aspect:

- Instead of asking which measurement is closest
- Ask for the exact calculation
- Need to specify one consistent unit to avoid ambiguity

4. What to preserve:

- All geometric relationships
- The 3-unit subtraction
- The original scenario and values

5. What needs to be added:

- Specific unit requirement (m^3)
- Decimal place requirement (5 places)
- Clear instruction about including units

6. The reformulated question should:

- Be completely open-ended
- Have a unique, calculable answer
- Remove all traces of comparison
- Maintain the same difficulty level

</reasoning>
<reformulated_problem>
In the cube shown, L and K are midpoints of adjacent edges AD and AB, respectively. The perpendicular distance from vertex F to the line segment LK is 10 units. Calculate the volume of the cube after subtracting 3 cubic units. Express your answer in cubic meters (m^3), rounded to 5 decimal places.
</reformulated_problem>
</example>
“‘

Multiple Choice Reformulation Judge (Part 1)

You are a IMO judge, specialized in evaluating the quality of IMO math problem reformulations. Your task is to critically analyze whether a reformulated problem truly meets the criteria for being open-ended, challenging, and maintains the original problem's integrity.

Important:
- Problems you review are being transformed from multiple choice problems into open-ended, solvable questions suitable for an automatic grading system that relies on regex pattern matching within ‘$\boxed{}$‘. It is VERY important that even though the problem is no longer multiple choice, it still has a single, definite answer.
- It must be reasonably clear from the problem what the student is expected to write inside of $\boxed{final\_answer}$. E.g. "What are its key properties and characteristics?" is almost always a bad reformulation because it is ambiguous what the student should write inside of the ‘$\boxed{}$‘. "Express your answer in centimeters." is a good answer format specification because it does NOT give away the answer but does specify a clear format.

Key aspects to evaluate:

1. Hidden Multiple Choice
- Check if the reformulation still effectively presents multiple choice options by:
- Embedding a limited/fixed set of choices (e.g. 4 or 5 options) within the problem text that limits the answer choices to one of those options
- Asking to compare with specific values
- Limiting answers to specific options
- Flag any reformulations that are just disguised multiple choice questions

Example:

<problem>
For real numbers $t \neq 0$, the point

$$(x, y) = \left(\frac{t+1}{t}, \frac{t-1}{t}\right)$$

is plotted. All the plotted points lie on what kind of curve? (A) Line (B) Circle (C) Parabola (D) Ellipse (E) Hyperbola Enter the letter of the correct option.
</problem>

Reformulated failed example (all multiple choice aspect still present):
<bad_reformulated_problem>
For real numbers $t \neq 0$, the point

$$(x, y) = \left(\frac{t+1}{t}, \frac{t-1}{t}\right)$$

is plotted. What type of curve do all the plotted points lie on? Provide your answer as a specific curve type (e.g., line, circle, parabola, ellipse, hyperbola).
</bad_reformulated_problem>

Reformulated successful example (multiple choice aspect removed):
<reformulated_problem>
For real numbers $t \neq 0$, the point

$$(x, y) = \left(\frac{t+1}{t}, \frac{t-1}{t}\right)$$

is plotted. Determine the type of coordinate geometry curve on which all the plotted points lie.
</reformulated_problem>

2. Mathematical Integrity
- Verify that the mathematical difficulty remains unchanged
- Ensure no accidental hints or simplifications were introduced
- Check that all necessary information was preserved
- Confirm no extraneous information was added

Example 1:
This problem was not actually multiple choice, but it does imply options that are clearly missing/omitted.
<problem>
A resident wants to renovate their house and buys several strips of wood, each with a length of 0.7 meters and 0.8 meters. By connecting some of these wood strips, many different lengths of wood can be obtained. For example, $0.7 + 0.7 = 1.4$ meters, $0.7 + 0.8 = 1.5$ meters, etc. From the options below, what length of wood strip cannot be obtained by connecting these wood strips?
</problem>

Multiple Choice Reformulation Judge (Part 2)

Example 2:
<problem>
$\triangle ABC$ is inscribed in a semicircle of radius r so that its base AB coincides with diameter AB. Point C does not coincide with either A or B. Let $s = AC + BC$. Then, for all permissible positions of C:
(A) $s^2 \leq 8r^2$ **(B)** $s^2 = 8r^2$ **(C)** $s^2 \geq 8r^2$
(D) $s^2 \leq 4r^2$ **(E)** $s^2 = 4r^2$
</problem>

The reformulation is flawed because it prematurely focuses on a single inequality ($s^2 \leq 8r^2$) rather than inviting exploration of the entire range of s^2. It biases the solver, limits generality, and reduces the problem's open-ended nature.
<bad_reformulated_problem>
$\triangle ABC$ is inscribed in a semicircle of radius r so that its base AB coincides with diameter AB. Point C does not coincide with either A or B. Let $s = AC + BC$. Prove or disprove the inequality $s^2 \leq 8r^2$ for all permissible positions of C.
</bad_reformulated_problem>

An open-ended revision that requires the same analysis and leads to the same conclusion
<reformulated_problem>
Let triangle ABC be inscribed in a semicircle of radius r, with its base AB coinciding with the diameter AB. Point C lies on the semicircle but does not coincide with A or B. Let $s = AC + BC$. Determine the maximum possible value of s^2 in terms of r, and prove that $s^2 \leq 8r^2$ for all permissible positions of C.
</reformulated_problem>

Example 3:
<problem>
For real numbers t, the point

$$(x, y) = \left(\frac{1-t^2}{1+t^2}, \frac{2t}{1+t^2}\right)$$

is plotted. All the plotted points lie on what kind of curve? (A) Line (B) Circle (C) Parabola (D) Ellipse (E) Hyperbola Enter the letter of the correct option.
</problem>

Although it removes the multiple-choice format, it does not truly open the problem to exploration. The answer asks for a description, which introduces too much variability in responses <bad_reformulated_problem>
For real numbers t, the point

$$(x, y) = \left(\frac{1-t^2}{1+t^2}, \frac{2t}{1+t^2}\right)$$

is plotted. What kind of curve do all the plotted points lie on? Provide a brief description of the curve.
</bad_reformulated_problem>

This reformulation is clear, concise, and ensures consistent answers while remaining open-ended for exploration.
<reformulated_problem>
For real numbers t, consider the point

$$(x, y) = \left(\frac{1-t^2}{1+t^2}, \frac{2t}{1+t^2}\right).$$

Determine the type of curve on which all such points (x, y) lie. State your answer as the name of the curve.
</reformulated_problem>

Example 4:

<problem>
Define * as an operation on ordered pairs of real numbers, such that $(a, b) * (c, d) = (ac + bd, ad + bc)$. If $(a, b) * (x, y) = (a, b)$, then which of the following is (x, y)? (A) $(0, 0)$. (B) $(0, 1)$. (C) $(1, 0)$. (D) $(1, 1)$. (E) $(1, -1)$.
</problem>

Ambiguous about whether the equation holds for all (a, b) or specific values, lacks context to ensure a unique solution, and removes guidance provided by the multiple-choice format. Becomes unbounded.
<bad_reformulated_problem>
Define * as an operation on ordered pairs of real numbers, such that $(a, b) * (c, d) = (ac + bd, ad + bc)$. If $(a, b) * (x, y) = (a, b)$, then calculate the ordered pair (x, y). Express your answer in the format (x, y), including parentheses and a comma.
</bad_reformulated_problem>

Multiple Choice Reformulation Judge (Part 3)

Ideal because it clarifies the scope (applies to all (a, b)), ensures a unique solution, and balances open-ended exploration with clear grading criteria
<reformulated__problem>
Define $*$ as an operation on ordered pairs of real numbers, such that

$$(a, b) * (c, d) = (ac + bd, ad + bc).$$

Find the ordered pair (x, y) that satisfies

$$(a, b) * (x, y) = (a, b)$$

for **all** real numbers a and b. Express your answer as (x, y).
</reformulated__problem>

3. Answer Format Clarity
- Evaluate if the answer format specification is:
- Clear and unambiguous
- Appropriate for the mathematical concept
- Not overly unbounded or restrictive in a way that creates an open-ended problem out of it
- Check if unit/precision requirements make sense for the problem

4. Problem Independence
- Verify the reformulated problem can stand alone
- Ensure it doesn't rely on knowledge of the original options
- Check that answer requirements aren't derived solely from original choices

Example:
<problem>
Which of the following is a root of the equation $x^2 - x - 6 = 0$? (A) -3 (B) -2 (C) 2 (D) 3 (E) 6
</problem>

Relies on original options, failing to stand independently and limiting exploration.
<bad__reformulated__problem>
Find a root of the equation $x^2 - x - 6 = 0$. Your answer must be one of the following: $-3, -2, 2, 3, 6$.
</bad__reformulated__problem>

Ideal because it ensures the solver identifies all roots without being constrained by the original options.
<reformulated__problem>
Solve the quadratic equation $x^2 - x - 6 = 0$ and find all real roots. Provide your answers in increasing order.
</reformulated__problem>

Remember: A truly open-ended reformulation should allow for calculation and expression of the answer without any reference to or knowledge of the original multiple choice options.