

Guiding Through Complexity: What Makes Good Supervision for Hard Reasoning Tasks?

Xuan He*♣ Da Yin*♠ Nanyun Peng♠

♣ Tsinghua University

♠ University of California, Los Angeles

hexuan21@mails.tsinghua.edu.cn, {da.yin,violetpeng}@cs.ucla.edu

Abstract

How can “weak teacher models” (Bowman et al., 2022) such as average human annotators or existing AI systems, effectively supervise LLMs to improve performance on hard reasoning tasks, especially those that challenge and requires expertise or daily practice from the teacher models? In this paper, we seek for empirical answers to this question by investigating various data-driven strategies that offer supervision data at different quality levels upon tasks of varying complexity. Two intuitive strategies emerge for teacher models to provide supervision during alignment training: 1) using lower-quality supervision from hard full tasks that match the difficulty of the target reasoning tasks, and 2) leveraging higher-quality supervision from easier subtasks that are less challenging. Interestingly, we find that even when the outcome error rate for hard task supervision is high (e.g., 90%), training on such data can outperform perfectly correct supervision on easier subtasks on multiple hard math benchmarks. We further identify a more critical factor influencing training performance: step-wise error rates, which indicate the severity of errors in solutions. Specifically, training on hard task supervision with the same outcome error rates but disparate step-wise error rates can lead to a 30% accuracy gap on MATH benchmark. Our results also reveal that supplementing hard task supervision with the corresponding subtask supervision can yield notable performance improvements than simply combining rephrased hard full task supervision, suggesting new avenues for data augmentation. Data and code will be released at <https://github.com/hexuan21/Weak-to-Strong>.

1 Introduction

The rapid advancements in Large Language Models (LLMs) (OpenAI, 2024b; Anthropic, 2024;

* These two authors contributed equally. The order of authorship decided by the flip of a coin.

Reid et al., 2024; Dubey et al., 2024) indicate that AI systems will soon be capable of progressively solving hard reasoning tasks that challenge both humans and current LLMs. To enhance the performance of today’s AI systems (Bowman et al., 2022; Burns et al., 2023) on such hard reasoning tasks, especially those that exceed the reasoning abilities of the average human or current LLMs, what strategies should we adopt to consistently and effectively supervise the AI systems?

We are concerned what supervision data those “weak teacher models”, humans and current LLMs, should annotate or synthesize for better hard downstream task performance. We mainly consider two possible data-driven supervision strategies – **Strategy 1**: Sampling or annotating solutions directly on the hard tasks that both human annotators and AI models typically struggle with; **Strategy 2**: Sampling or annotating solutions on hard tasks’ corresponding subtasks, which are more manageable and where humans or AI models are more likely to succeed (Gülçehre and Bengio, 2016; Glasmachers, 2017; Wies et al., 2023; Jothimurugan et al., 2023). For Strategy 1, while the tasks used for supervision match the reasoning difficulty of our studied hard downstream tasks, the quality of supervision may be compromised due to the difficulty that challenges the human and AI teacher models. In contrast, Strategy 2 involves easier tasks that do not match the difficulty level of the downstream tasks, but the supervision quality tends to be higher because these tasks are simpler for annotators or models to handle. Should we obtain low-quality supervision for hard tasks, or employ high-quality supervision for decomposed subtasks for fine-tuning?

To fairly compare the two supervision strategies, which involve task difficulty and supervision quality, we introduce a simulated setting: The supervision quality is controlled by introducing varying outcome error rates, which represent the percentage

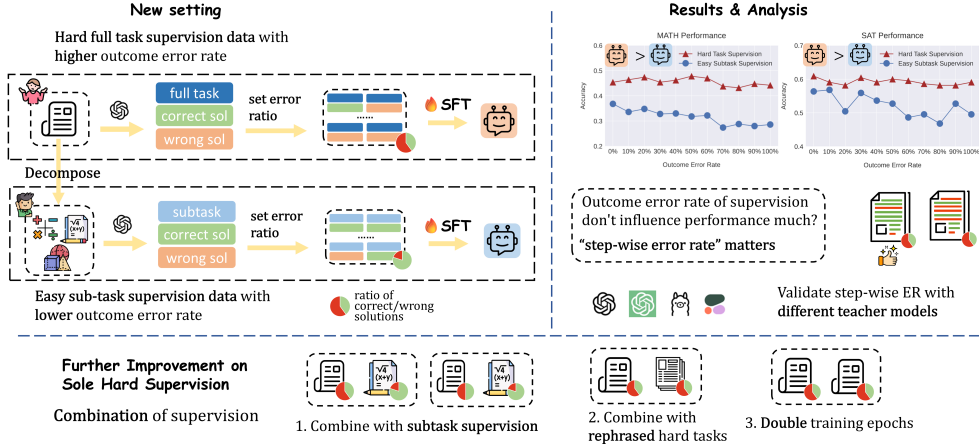


Figure 1: Overview of our empirical study on two contrasting supervision strategies and further analysis.

of noisy supervision where solutions with wrong final answers replace the ground-truth solutions. It is designed for simulating the varying supervision quality from “weak” teacher models with different capacity. After acquiring full task and subtask supervision with varying outcome error rates, and following Bansal et al. (2024), we apply supervised fine-tuning (SFT) using the acquired supervision to LLMs for solving hard reasoning tasks.

Across 5 hard math reasoning tasks, including college entrance exams, high-school competitions and college-level math tests, we find hard task supervision consistently outperforms subtask supervision, regardless of outcome error rates. Specifically, we observe that: 1) even if outcome error rate of hard supervision is 90%, it can still outperform perfectly correct subtask supervision; 2) for easy subtask and hard full task supervision, as the outcome error rate increases from 0% to 100%, the performance does not decrease monotonically.

Rather than the outcome error rate, what factors highly impact the hard test task performance? One key factor we identify is the *step-wise error rate*, i.e., average ratio of erroneous steps (§5) to the total steps in the supervision set. Specifically, we observe that when different teacher models exhibit similar outcome error rates, an increase in step-wise error rate leads to lower performance on hard reasoning tasks. We test a batch of LLMs, including GPT-4o-mini (OpenAI, 2024a), GPT-3.5-turbo (OpenAI, 2022), Llama-3-70B-Instruct, and Command-R (Cohere, 2024), as teacher models for providing hard task supervision. We then compare the performance of the rest models’ supervision against that of GPT-4o-mini at a similar outcome error rate. Notably, with a lower step-wise error rate, GPT-4o-mini’s sampling supervision always leads

to higher results. When both GPT-4o-mini and Command-R have around 90% outcome error rate in their generated supervision, Command-R underperforms by 12% accuracy on MATH (Hendrycks et al., 2021b) and is even worse than training with subtask supervision.

Can we further improve the hard downstream task performance without relying on more hard task annotations? We find that supplementing hard task supervision with subtask supervision, i.e., the supervision of hard tasks’ corresponding subtasks, can lead to more performance gains. Specifically, we combine the hard task supervision with an outcome error rate of A and the subtask supervision with an outcome error rate of B with three distinct combinations of A and B . We observe that adding subtask supervision improves performance across most of the 5 downstream tasks. When $A = 50\%$ and $B = 10\%$, we achieve the best results on benchmarks like MATH and SAT-Math. We also find that this simple supervision combination method is more effective than simply doubling the training epochs on the mere hard task supervision or mixing rephrased hard task supervision.

Overall, we summarize our main contributions for enhancing LLM’s hard reasoning task performance as follows:

- We offer new insights into how complexity and quality of weak supervision collectively impact hard task performance, distinct from traditional weak supervision and weak-to-strong generalization literature.
- We conduct one of the most comprehensive evaluations across a range of hard reasoning tasks in weak-to-strong generalization field, surpassing in both scope and task difficulty. We also introduce

a fairer comparison by ensuring that the supervision from easy subtasks and hard full tasks is kept within a similar range.

- We provide comprehensive and novel insights which contribute data-driven guidance for researchers and practitioners aiming to improve LLMs in future reasoning-heavy applications.

2 Formulation of Supervision Strategy Analysis

In this section, we present the framework for our analysis of supervision strategies aimed at enhancing performance on hard downstream reasoning tasks. We outline the motivation behind two supervision strategies involving both easy and hard tasks, introduce the key notations, and discuss the design principles of the analysis formulation for fair and comprehensive evaluation.

2.1 Overview of Supervision Strategies

Future AI systems need to solve hard reasoning tasks that “weak teacher models” (e.g., average humans or current AI systems) currently struggle with. To achieve better performance on such hard tasks, what types of training supervision should these teacher offer for training LLMs? We propose the two following supervision strategies, each with distinct characteristics that could contribute to this improvement:

Strategy 1: Use teacher models to sample or annotate solutions for **hard full tasks** that human annotators or AI models typically struggle with. Although this supervision aligns with the reasoning difficulty of the target downstream tasks, the supervision may be **lower-quality** due to the task complexity for obtaining correct final answer.

Strategy 2: Use teacher models to sample or annotate solutions for **easy subtasks**. These tasks, while not matching the complexity of the downstream tasks, offer **higher-quality** supervision because they are simpler to handle.

We denote the two types of supervision as $\mathcal{D}_{\text{Hard}}$ and $\mathcal{D}_{\text{Subtask}}$. We also define supervision quality of $\mathcal{D}_{\text{Hard}}$ and $\mathcal{D}_{\text{Subtask}}$ in terms of outcome error rates, i.e., the proportion of incorrect solutions with wrong final answers relative to the total supervision instances. Higher outcome error rates indicate lower supervision quality. Following Bansal et al. (2024), we employ supervised fine-tuning (SFT) with the supervision from the above strategies to

train LLMs for solving hard tasks and determine the more effective strategy.

In summary, we formally examine **whether hard task supervision $\mathcal{D}_{\text{Hard}}$ with outcome error rate ϵ_{Hard} yields better performance on hard reasoning tasks compared to the easy subtask supervision $\mathcal{D}_{\text{Subtask}}$ with error rate $\epsilon_{\text{Subtask}}$.**

2.2 Design Principles of Analysis Formulation

To ensure the fairness and comprehensiveness of our analysis, we address the following aspects:

Comparison Fairness. *We ensure that subtasks must be less difficult than hard full tasks and avoid introducing any extra tasks that go beyond the scope of hard task supervision and might help solve the test tasks (§3.2).* It was neglected in prior related work which involves preparing easy and hard task supervision (Sun et al., 2024; Hase et al., 2024; Bansal et al., 2024).

Evaluation with Varying Supervision Quality. *Even if some teacher models like human annotators are weak, they still have varying capabilities.* For instance, some may fail 60% of hard full tasks, while others might fail up to 90%. To simulate varying supervision quality from those teacher models with different capacity, we first collect a substantial set of hard full tasks and subtasks, each containing both ground-truth and incorrect solutions (§3.3). To create supervision with $A\%$ outcome error rate, we can simply replace $A\%$ of the original ground-truth solutions with incorrect ones.

3 Hard Full Task and Easy Subtask Supervision Synthesis

In this section, we detail the process of collecting easy subtask and hard full task supervision, $\mathcal{D}_{\text{Subtask}}$ and $\mathcal{D}_{\text{Hard}}$, based on the design principles of our supervision strategy analysis framework.

3.1 Overview

As outlined in §2.2, we need to decompose hard tasks into subtasks, ensuring that each decomposed subtask remains within the scope of the original hard task’s solutions. Additionally, we introduce an outcome error rate metric as an indicator of supervision quality, allowing us to simulate varying levels of quality based on the capabilities of different teacher models. This enables us to analyze performance trends in relation to changes in supervision quality and facilitates direct comparisons between hard full task and subtask supervision across

different quality levels. To this end, our supervision synthesis pipeline comprises two main stages: **Stage 1: Hard Task Decomposition** and **Stage 2: Outcome Error Rate Adjustment**.

Our synthesis pipeline utilizes large-scale high-quality math training annotations (See §3.4) as the foundations. These resources provide many hard reasoning tasks with chain-of-thought solutions, which are fundamental for task decomposition, and supervision quality adjustment and validation.

3.2 Stage 1: Hard Task Decomposition

Concretely, we leverage the hard full tasks and their solutions from the large-scale high-quality math resource, such as NuminaMath-CoT, to construct $\mathcal{D}_{\text{Hard}}$ (as detailed §3.4). Then, with the hard task decomposition prompt shown in Appendix I, we employ GPT-4o-mini to break down each hard task and its complete solution into 2-3 subtasks along with their corresponding solutions derived directly from the hard full task solutions.

To maintain the quality of subtasks, we filter out unqualified tasks with ill-defined problems and overly-simple solutions (Appendix D.2). After filtering, the product of Stage 1 is a collection of hard full tasks and their subtasks, each paired with a ground-truth solution and final answer. They serve as the foundation for generating supervision $\mathcal{D}_{\text{Hard}}$ and $\mathcal{D}_{\text{Subtask}}$ with varying outcome error rates and the filtering process also bridges the distribution gap between the text style of hard full and easy subtask instructions (we explore further about the potential discrepancies between $\mathcal{D}_{\text{Hard}}$ and $\mathcal{D}_{\text{Subtask}}$ in Appendix F). In a small sample set, the solutions to hard tasks have an average of 13.01 steps while for subtasks, it's 5.96.

3.3 Stage 2: Outcome Error Rate Adjustment

To introduce supervision quality change and conduct comparison at different quality levels, we swap a percentage of ground-truth solutions with the incorrect ones to adjust supervision quality. For example, to create hard task supervision with a 20% outcome error rate, 20% of ground-truth solutions are replaced with incorrect ones.

However, Stage 1 only offers ground-truth solutions and this process requires incorrect task solutions. How can we obtain the incorrect solutions for error-introducing replacement? Next we outline the process to ensure each task has both ground-truth and incorrect solutions.

Sampling Multiple Solutions. We first use LLMs to perform multi-sampling, generating **three solutions** for each hard full task and subtask. To increase the likelihood of obtaining incorrect solutions for the tasks with different complexities, we leverage LLMs with different reasoning capabilities. Concretely, we employ GPT-4o for hard tasks and GPT-4o-mini for subtasks. After generating the three candidate solutions, we assess their correctness by comparing each one against the ground-truth final answers.

Gathering and Filtering Tasks with Ground-Truth and Incorrect Solutions. With three sampled solutions and the ground-truth, we attempt to formulate solution pairs for each task, with one correct and one incorrect. Details and examples are shown in Appendix D.3. It may occur that all decomposed subtasks corresponding are filtered out, or vice versa, where a subtask's corresponding hard full task is filtered out. To ensure that the hard full tasks in $\mathcal{D}_{\text{Hard}}$ encompass the scope of the subtasks in $\mathcal{D}_{\text{Subtask}}$ for comparison fairness discussed in §2.2, we enforce that every hard full task in $\mathcal{D}_{\text{Hard}}$ must have at least one corresponding subtask in $\mathcal{D}_{\text{Subtask}}$, and each subtask in $\mathcal{D}_{\text{Subtask}}$ must correspond to a hard task in $\mathcal{D}_{\text{Hard}}$.

Controlling Supervision Error Rate by Adjusting the Proportion of Noisy Supervision. Once we ensure that all hard full tasks and easy subtasks have both ground-truth and incorrect solutions, we adjust the supervision quality by substituting a specified percentage of the original ground-truth solutions with incorrect ones. For example, to create hard task supervision with a 20% error rate, we replace 20% of the ground-truth solutions with their corresponding incorrect versions.

Overall, after Stage 2, we obtain the final supervision datasets $\mathcal{D}_{\text{Hard}}$ and $\mathcal{D}_{\text{Subtask}}$ with outcome error rates ranging from 0% to 100% in 10% increments, simulating varying supervision quality from different teacher models. As outlined in §2.1, following (Bansal et al., 2024), we apply SFT using the two types of supervision to train LLMs on hard reasoning tasks, followed by a comprehensive analysis of their performance.

3.4 Supervision Sources and Quality Check

We focus on the math domain and apply the pipeline to various math sources, including three subsets from NuminaMath-CoT (Li et al., 2024b), PRM800K (Lightman et al., 2023), along with the

difficulty levels 4-5 from MATH, all featuring **high school competition, college entrance exam and college-level exam difficulties** (See details in Appendix B). Table 4 summarizes the quantity of hard full task and the subtask supervision from each data source. In total, $\mathcal{D}_{\text{Hard}}$ contains 42,274 ground-truth hard task solutions, while $\mathcal{D}_{\text{Subtask}}$ includes 67,328 ground-truth subtask solutions. We then introduce varying error rates, ϵ_{Hard} and $\epsilon_{\text{Subtask}}$, for both $\mathcal{D}_{\text{Hard}}$ and $\mathcal{D}_{\text{Subtask}}$. We present a thorough analysis and detailed case studies on the supervision synthesis process, spanning Appendix D.1 to D.3, to illustrate the effectiveness of our carefully simulated settings for fair comparison. We conduct human evaluation on 60 samples to further verify the data quality, as described in Appendix E.

4 Which Supervision Strategy is Better?

4.1 Setup

We compare the hard math task performance after training with final supervision $\mathcal{D}_{\text{Hard}}$ and $\mathcal{D}_{\text{Subtask}}$ obtained from the two supervision strategies. Our test data of hard tasks encompass a range of difficulty levels, spanning from high school competition to college-level exams. This includes challenges from the Olympic-Arena (Huang et al., 2024), MATH, and JEE-Bench (Arora et al., 2023a) (as detailed in Appendix C). As outlined in §3.3, we introduce varying error rates from 0% to 100% in 10% increments to simulate different quality levels for both hard and subtask supervision.

Inspired by Sun et al. (2024), we first establish a base model with fundamental instruction-following and mathematical reasoning abilities. To this end, we fine-tune Llama-3-8B using elementary school and middle school-level math data from Orca-Math (Mitra et al., 2024) subset of NuminaMath-CoT, as well as the MATH dataset (level 1-3). It serves as the base model that we will train upon using $\mathcal{D}_{\text{Hard}}$ and $\mathcal{D}_{\text{Subtask}}$. Additional training and inference hyper-parameters are provided in Appendix A.

4.2 Observations

Figure 2 reveals several intriguing insights regarding the comparison between hard task supervision and subtask supervision, as well as the resulting trends in performance changes.

Hard task supervision consistently outperforms subtask supervision, even with higher outcome error rates. Despite the higher error rate of hard task supervision $\mathcal{D}_{\text{Hard}}$, models trained on $\mathcal{D}_{\text{Hard}}$

still demonstrate superior performance across all five test sets. Remarkably, $\mathcal{D}_{\text{Hard}}$ with a 90% error rate outperforms $\mathcal{D}_{\text{Subtask}}$ with a 10% error rate by approximately 11% on the MATH and 16.9% on Gaokao-Mathcloze. This suggests that, even with a high error rate, adopting hard task supervision is *seemingly* a more beneficial choice than leveraging more accurate subtask supervision.

Performance does not consistently degrade with increasing outcome error rates. Across 5 test benchmarks, accuracy remains stable even as the error rate increases. For example, on JEE-Bench, accuracy fluctuates with rising error rates, while on SAT and MATH, performance remains steady, showing marginal changes with variations in outcome error rates. Interestingly, hard task supervision with a 100% error rate is more effective than with a 10% error rate on JEE-Bench. This finding is different from previous studies in weak supervision literature (Zhang et al., 2017), which generally suggest that supervision quality significantly impacts training performance.

Changes in outcome error rates have a greater impact on subtask supervision than on hard task supervision. As shown in Figure 2, increasing the outcome error rates in subtask supervision leads to a more significant drop in performance compared to hard task supervision. This is evident in the performance gap between the highest and lowest results achieved among varying error rates. For instance, on MATH, the performance drop for subtask supervision is 9.4%, whereas for hard full task supervision, it is only 4.6%. This suggests that the SFT process is more robust to outcome error rate variations when using hard full task supervision compared to subtask supervision.

In summary, it appears that irrespective of the quality of hard full task supervision, it consistently outperforms subtask supervision, even if the latter is more precise in practice. Also, training seems to be more robust across varying quality levels when using hard full task supervision. Why does this occur? Should we always rely on hard full task supervision with high outcome error rates to achieve strong performance on hard tasks? What other factors should we take into account? We explore these questions further in §5.

5 Severity of Erroneous Solutions Matters

Observations in §4 suggest that in domains requiring long-form reasoning, such as math, **outcome**

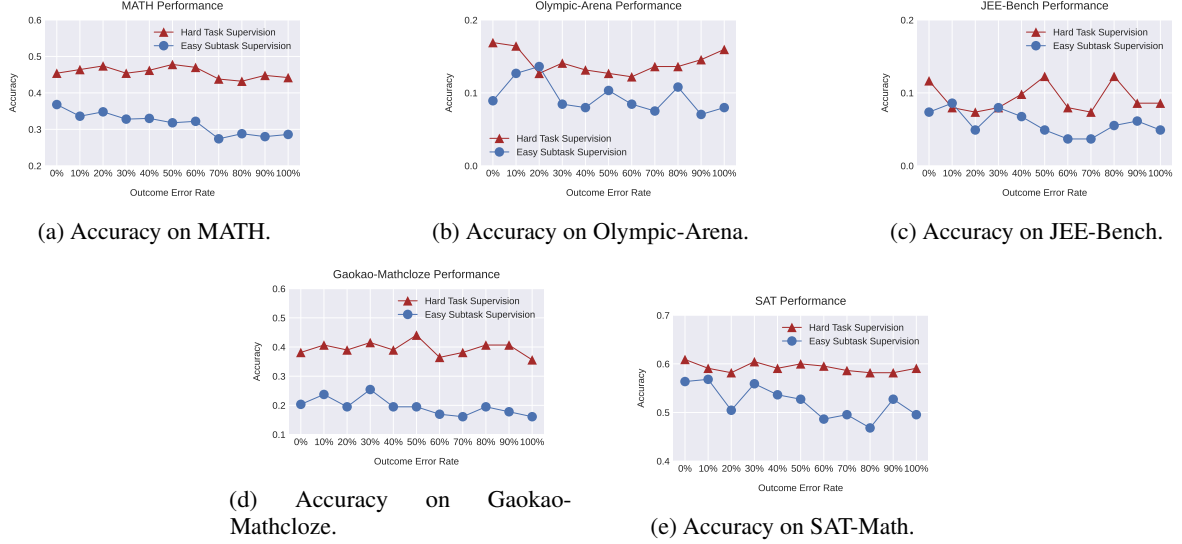


Figure 2: Comparison of easy and hard task supervision with varying outcome ER on 5 hard reasoning benchmarks.

error rate MAY NOT be a reliable indicator of quality. Earlier analysis define “supervision quality” based on a binary match between the final predicted result and ground truth, ignoring the correctness of intermediate steps. Two incorrect solutions may differ greatly in quality: one with a minor computational error but coherent, and another with fundamentally flawed logic. The latter represents a more severe error, yet this distinction is missed by a simple binary evaluation.

To evaluate error severity in solutions, we introduce the concept of **erroneous step**, defined as a step that is logically incorrect or irrelevant to preceding steps. Importantly, we do not classify a step as erroneous if it is logically correct but only relies on incorrect intermediate values from prior mistakes. Given similar outcome error rates, it is intuitive to consider that the proportion of erroneous steps affects the quality of supervision. To validate this on a larger scale, we first examine the impact of using different teacher models with similar outcome error rates and then analyze whether the severity of step-wise errors correlates with the observed performance gap.

5.1 Setup

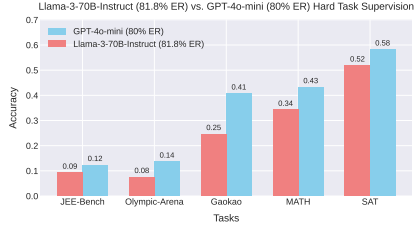
In addition to using the solutions previously sampled from GPT-4o-mini, we include GPT-3.5-turbo, Llama-3-70B-Instruct, Command-R-2024-08, and Command-R-2024-03 as **teacher models with varying capabilities** to generate solutions for constructing new hard full task supervision, denoted as $\mathcal{D}'_{\text{Hard}}$. Concretely, we sample one solution per hard full task in $\mathcal{D}_{\text{Hard}}$ and use these to create

$\mathcal{D}'_{\text{Hard}}$ for SFT. For each teacher model, we assess the outcome error rates of $\mathcal{D}'_{\text{Hard}}$ by comparing the predicted final results against the ground-truth answers of all the hard training tasks.

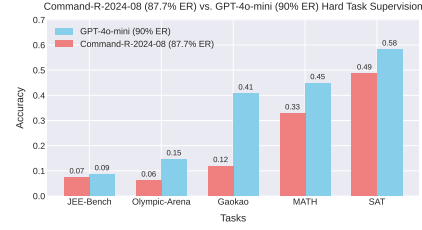
After obtaining the outcome error rates of $\mathcal{D}'_{\text{Hard}}$ generated by different teacher models, we identify the model variant $\mathcal{M}_{\text{Closest}}$ trained with GPT-4o-mini supervision with an outcome error rate closest to the new teacher model for comparison. We then compare $\mathcal{M}_{\text{Closest}}$ with the model trained on $\mathcal{D}'_{\text{Hard}}$ across downstream benchmarks. For example, if the supervision sampled from Llama-3-70B-Instruct has an outcome error rate of 81.8% on the hard tasks, we compare Llama-3-70B-Instruct sampled supervision with the corresponding model in §4.2 trained by GPT-4o-mini sampled supervision with an 80% outcome error rate.

5.2 Performance of Supervision Under Similar Outcome Error Rates

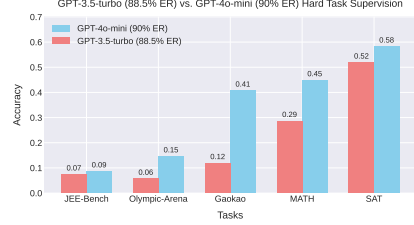
As shown in Figure 3, even with similar outcome error rates, models trained with supervision from diverse teacher models behave differently. For instance, a model trained on solutions from Command-R-2024-08, which has an outcome error rate of 87.7%, achieves only 32.8% accuracy on MATH. In contrast, the model trained on data from GPT-4o-mini, despite a higher error rate of 90%, achieves significantly better accuracy at 44.8%. Additionally, the model trained with supervision from Llama-3-70B-Instruct (81.8% error rate) performs worse than one trained with 100% incorrect data from GPT-4o-mini. Notably, Command-R-2024-03’s sampled hard task supervision is less effective



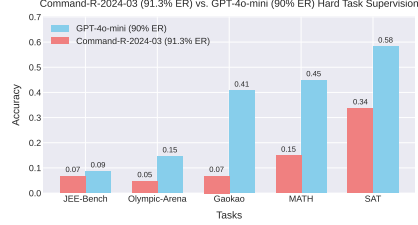
(a) Llama-3-70B-Instruct (81.7% ER) vs. GPT-4o-mini (80% ER) hard full task supervision.



(b) Command-R-2024-08 (87.7% ER) vs. GPT-4o-mini (90% ER) hard full task supervision.



(c) GPT-3.5-turbo (88.5% ER) vs. GPT-4o-mini (90% ER) hard full task supervision.



(d) Command-R-2024-03 (91.3% ER) vs. GPT-4o-mini (90% ER) hard full task supervision.

Figure 3: Performance of the supervision synthesized by different teacher models under similar outcome error rates (a.k.a., ER).

Teacher models	Step-wise Error Ratio (%)
GPT-4o-mini	11.7
Llama-3-70B-Instruct	30.2
Command-R-2024-08	40.0
Command-R-2024-03	62.6
GPT-3.5-turbo	34.7

Table 1: Step-wise error rates (%) for teacher models.

than using the subtask supervision from GPT-4o-mini in 4 out of 5 test hard tasks. This suggests the key factor influencing performance may lie in the quality of the solution steps embedded within the supervision from different teacher models.

5.3 Correlation Between Performance Disparity and Step-Wise Error Rates

To further validate our hypothesis about the severity of incorrect solutions generated by different teacher model models, human evaluation is required. We first collect 60 tasks along with their incorrect solutions from the supervision sampled by each of the four studied teacher models discussed in §5.1, including Command-R-2024-08 and Llama-3-70B-Instruct. We then conduct a manual review of these incorrect solutions. Specifically, we count the total steps and erroneous steps in each solution and calculate **step-wise error rates** by averaging the ratio of erroneous steps to total steps across all the solutions of sampled 60 examples (See Appendix H for comprehensive case studies). As shown in Table 1, the step-wise error rates vary among these teacher

models, with Command-R-2024-03 exhibiting the highest ratio at 62.6% and GPT-4o-mini having the lowest at 11.7%. From Figure 3, we observe a strong correlation between the step-wise error rates and downstream performance: lower rates correspond to higher downstream performance, aligning more closely with results from supervision synthesized by GPT-4o-mini. This indicates that the step-wise error rate is a strong indicator of supervision quality and its impact on hard task performance. Besides the above observations, we attempt to borrow step-wise error rate metric to explain more phenomenons, detailed in Appendix G.

6 Further Improvement Over Sole Hard Task Supervision

As outlined previously, hard task supervision with lower step-wise error rates is highly beneficial. But can performance be further improved using existing hard task supervision without introducing new ones? A natural approach is to integrate easy subtask supervision into existing supervision sources. Since these subtasks are decomposed from the hard ones, this strategy boosts task diversity without adding new information, offering a potential avenue for further performance gains.

What happens after we combine them? Due to the substantial resources and time required to explore all possible combinations (a total of 11×11 combinations) of different outcome error rates for hard and easy data, we select several combinations:

(20%, 10%), (20%, 0%), (50%, 10%), (50%, 0%), (80%, 40%), (80%, 10%) and (80%, 0%), where the two values represent the outcome error rates for hard full task and easy subtask supervision. In each combination, the outcome error rate for hard data is higher, reflecting the reality that annotations for hard tasks are often less accurate.

From another perspective, since the information in hard tasks essentially covers that of the subtasks, one might argue that combining easy subtask and hard full task supervision merely makes LLMs learn the same information roughly twice. We design a baseline denoted as “doubled epochs”, where we train the model on hard task supervision for 4 epochs, which double the training epochs of all previous experiments. We also merge the existing hard task supervision along with its rephrased version which increases more diversity.

Combined Supervision vs. Doubled Epochs vs. Merge Rephrased. Table 2 shows that in most cases, combining easy subtask and hard task supervision enhances performance. For instance, combining hard task supervision with a 50% outcome error rate and subtask supervision with a 10% outcome error rate leads to the highest accuracy on the MATH, Gaokao-Mathcloze and SAT-Math among all previously trained baselines. It also outperforms doubling the training epochs or mixing rephrased data with original SFT supervision, highlighting that integrating subtask supervision can push performance limits beyond repetitive supervision.

Effects of Step-Wise Error Rates of Hard Task Supervision after Merging. We also combined hard task supervision from other different teacher models (Llama-3-70B-Instruct and Command-R-2024-03) and the original easy subtask supervision. We then compared these new combinations with the original ones to further assess the effect of step-wise error rate in hard task supervision, as discussed in §5. The results align with our earlier findings: when a teacher model exhibits a lower step-wise error rate, the performance of the combined hard and easy supervision improves. As shown in Table 1, Command-R-2024-03, which has the highest step-wise error rate, yields the lowest accuracy on the test data. This comparison reinforces the importance of step-wise error rate in determining the quality of math reasoning supervision data for hard downstream tasks.

7 Concluded Empirical Advice

Based on the findings above, we offer some practical advice for optimizing annotation efforts of weak teacher models to enhance on hard tasks:

Prioritize hard task supervision for better performance on challenging reasoning tasks. We reveal in §4 that the best performance is consistently achieved with hard full task supervision. It significantly surpasses the results obtained using only subtask supervision.

Step-wise error rates of teacher models is more critical than outcome error rates on hard reasoning task performance. In §5, we find that, even with identical outcome error rates, supervision with higher step-wise error rates leads to more evident performance drop than what we observe when the outcome error rates change.

To further enhance performance, consider sampling or annotating hard tasks and their associated subtasks, then use the combined supervision to train LLMs. As shown in §6, given a fixed task scope covered in the hard task supervision, integrating hard task supervision and their corresponding subtask supervision boosts or even achieves the best performance across benchmarks. This combination is also more effective than simply increasing the training epochs or further merging the rephrased version of the original supervision.

8 Related Works

Weak Supervision. Weak supervision has traditionally been applied to text classification tasks, typically following an iterative framework that generates pseudo-labels using heuristics and expands the training set (Agichtein and Gravano, 2000; Meng et al., 2018). Recent works have employed LLM-based teacher models to generate training data for fine-tuning smaller models (Bosselut et al., 2019; West et al., 2022; Wang et al., 2023; Hsieh et al., 2023; Brahman et al., 2023), enabling student models to follow instructions and perform reasoning. This approach assumes that a strong teacher LLM offers high-quality supervision. However, our focus shifts to challenging reasoning tasks that even a current strong LLM struggles to solve. We analyze how supervision complexity and quality jointly impacts performance when a teacher model provides noisy weak supervision for hard tasks.

Easy-to-Hard Generalization. This area (Zhou et al., 2022; Burns et al., 2023) involves training on easy task supervision and then evaluating on

Tasks	Combination 1			Combination 2			Combination 3			
	(20%, 10%)	(20%,0%)	20%	(50%, 10%)	(50%,0%)	50%	(80%, 40%)	(80%, 10%)	(80%, 0%)	80%
MATH	43.2	48.0	47.4	48.6	45.8	47.8	43.8	46.8	47.2	43.2
Olympic-Arena	13.6	16.4	12.7	12.7	18.8	12.7	14.1	19.2	17.8	13.6
JEE-Bench	12.3	10.4	7.4	11.7	9.2	12.3	12.3	9.8	7.4	12.3
Gaokao-Mathcloze	40.7	40.7	39	45.8	43.2	44.1	42.4	37.3	43.2	40.7
SAT-Math	58.2	60.5	58.2	61.0	63.2	60.0	60.0	66.8	62.3	58.2
Average	33.6	35.2	32.9	36.0	36.0	35.4	34.5	36.0	35.6	33.6

(a) Combination of easy and hard task supervision vs. Sole hard task supervision.

Tasks	Combination		
	(50%, 10%)	50% (Doubled Epochs)	50% (Merge Rephrased.)
MATH	48.6	45.0	48.4
Olympic-Arena	12.7	12.7	12.7
JEE-Bench	11.7	9.2	10.4
Gaokao-Mathcloze	45.8	42.4	44.1
SAT-Math	61.0	55.5	57.3

(b) Combined vs. Doubled Epochs vs. Merge Rephrased.

Table 2: Accuracy (%) of different combination methods. (a) combination of hard and easy task supervision with varying error rates, (b) other combinations: "Doubled Epochs" refers to doubling original training epochs, while "Merge Rephrased" denotes the training data that combines original hard tasks with the rephrased versions.

Outcome ER Combinations	Combination		
	GPT-4o-mini (80%, 10%)	Llama-3-70B-Instruct (81.7%, 10%)	Command-R-2024-03 (87.7%, 10%)
Step-Wise ER of Hard Task Supervision	11.7%	30.2%	62.6%
MATH	46.8	31.2	24.6
Olympic-Arena	19.2	12.2	5.2
JEE-Bench	8.7	10.4	5.5
Gaokao-Mathcloze	37.3	23.7	5.1
SAT-Math	61.0	54.5	47.7

Table 3: Combination performance of hard full task supervision from different teacher models and original easy subtask supervision, teacher models include GPT-4o-mini, Llama-3-70B-Instruct and Command-R-2024-03.

more difficult examples. Hase et al. (2024) demonstrate that in-context learning or fine-tuning on easy tasks offers a strong baseline for multiple-choice tasks such as ARC (Clark et al., 2018) and MMLU (Hendrycks et al., 2021a), while Sun et al. (2024) show that an easy-to-hard evaluator enhances a generator’s generalization on hard task benchmark MATH, based on the fact that evaluation is easier to learn than generation. In contrast to prior works, our study uniquely focuses on how two extra correlated aspects, supervision task difficulty and supervision quality, jointly influence final performance on hard tasks. Our experimental setup provides two key advantages: 1) one of the most comprehensive evaluations of hard reasoning tasks, far exceeding prior studies; and 2) a fairer comparison by controlling task scope to keep easy subtasks and hard full task supervision within a similar range.

9 Conclusion

In this paper, we explore two possible supervision strategies that weak teacher models can employ to enhance performance on hard reasoning tasks. We first find that even when the outcome error rate for hard task supervision is high, training on this data can surpass the performance of perfectly correct supervision from easier subtasks across multiple hard math tasks. Then, we pinpoint step-wise error rates as a more critical factor influencing hard task performance, as its variation can result in a more evident drop on the hard tasks. We further observe that if no more hard task supervision is allowed to be added, supplementing existing hard task supervision with corresponding subtask supervision yields greater performance than merely merging rephrased hard task supervision. These insights contribute valuable data-driven guidance for researchers and practitioners aiming to improve LLMs in future reasoning-heavy applications.

Limitations

In this section, we list several limitations of our current work and consider to address them to enhance the paper in the future.

Effects of Step-Wise Error Rates. In §5, we focused on examining the effects of step-wise error rates on the quality of supervision for hard full tasks, primarily due to cost considerations. Future work will include an investigation into the impact of step-wise error rates on subtask supervision. Additionally, we can perturb the supervision produced by the same teacher models to introduce more step-wise erroneous steps, further reinforcing the significance of step-wise error rates.

Supervision Combination Strategy. The experiments presented in §6 are conducted with a limited range of error rate combinations for hard full task and easy subtask supervision provided by GPT-4o-mini. We plan to test a broader array of combinations and extend our analysis to the supervision synthesized by other teacher models, such as GPT-3.5-turbo and Llama-3-70B-Instruct, if our training budget allows.

Exploration on More Reasoning Domains. Our experiments primarily target the math domain; however, we intend to explore other reasoning domains, such as scientific reasoning, to further bolster the arguments presented in our study.

Integrating More Alignment Methods into Our Supervision Strategy Analysis Framework. Following Bansal et al. (2024); Hase et al. (2024), we employ supervised fine-tuning (SFT) to investigate the effects of various types of supervision. Additionally, we plan to integrate more alignment training methods, such as DPO (Rafailov et al., 2024), to determine whether our observations remain consistent in the future work.

Acknowledgement

This research is based upon work supported by NSF CAREER #2339766, and an Amazon AGI Research Award. We thank anonymous reviewers and UCLA NLP group members for their valuable feedback.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94.

Anthropic. 2024. [Claude 3.5 Sonnet](#).

Daman Arora, Himanshu Singh, et al. 2023a. Have llms advanced enough? a challenging problem solving benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7527–7543.

Daman Arora, Himanshu Gaurav Singh, and Mausam. 2023b. [Have llms advanced enough? a challenging problem solving benchmark for large language models](#). *ArXiv*, abs/2305.15074.

Hritik Bansal, Arian Hosseini, Rishabh Agarwal, Vinh Q. Tran, and Mehran Kazemi. 2024. [Smaller, weaker, yet better: Training llm reasoners via compute-optimal sampling](#). *Preprint*, arXiv:2408.16737.

Antoine Bosselut, Hannah Rashkin, Maarten Sap, Chaitanya Malaviya, Asli Celikyilmaz, and Yejin Choi. 2019. [COMET: Commonsense transformers for automatic knowledge graph construction](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4762–4779, Florence, Italy. Association for Computational Linguistics.

Samuel R Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamilė Lukošiušė, Amanda Askell, Andy Jones, Anna Chen, et al. 2022. Measuring progress on scalable oversight for large language models. *arXiv preprint arXiv:2211.03540*.

Faeze Brahman, Chandra Bhagavatula, Valentina Pyatkin, Jena D Hwang, Xiang Lorraine Li, Hirona J Arai, Soumya Sanyal, Keisuke Sakaguchi, Xiang Ren, and Yejin Choi. 2023. Plasma: Making small language models better procedural knowledge models for (counterfactual) planning. *arXiv preprint arXiv:2305.19472*.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. 2023. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Cohere. 2024. [Command-R](#).

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Tobias Glasmachers. 2017. Limits of end-to-end learning. In *Asian conference on machine learning*, pages 17–32. PMLR.

- Çağlar Gülçehre and Yoshua Bengio. 2016. Knowledge matters: Importance of prior information for optimization. *The Journal of Machine Learning Research*, 17(1):226–257.
- Peter Hase, Mohit Bansal, Peter Clark, and Sarah Wiegrefe. 2024. [The unreasonable effectiveness of easy training data for hard tasks](#). *Preprint*, arXiv:2401.06751.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. [Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8003–8017, Toronto, Canada. Association for Computational Linguistics.
- Zhen Huang, Zengzhi Wang, Shijie Xia, Xuefeng Li, Haoyang Zou, Ruijie Xu, Run-Ze Fan, Lyumanshan Ye, Ethan Chern, Yixin Ye, et al. 2024. Olympiarena: Benchmarking multi-discipline cognitive reasoning for superintelligent ai. *arXiv preprint arXiv:2406.12753*.
- Kishor Jothimurugan, Steve Hsu, Osbert Bastani, and Rajeev Alur. 2023. Robust subtask learning for compositional generalization. In *International Conference on Machine Learning*, pages 15371–15387. PMLR.
- Chen Li, Weiqi Wang, Jingcheng Hu, Yixuan Wei, Nan-ning Zheng, Han Hu, Zheng Zhang, and Houwen Peng. 2024a. [Common 7b language models already possess strong math capabilities](#). *ArXiv*, abs/2403.04706.
- 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. 2024b. Numinamath. [<https://huggingface.co/AI-MO/NuminaMath-CoT>](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf).
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let’s verify step by step. *arXiv preprint arXiv:2305.20050*.
- Yu Meng, Jiaming Shen, Chao Zhang, and Jiawei Han. 2018. Weakly-supervised neural text classification. In *proceedings of the 27th ACM International Conference on information and knowledge management*, pages 983–992.
- Arindam Mitra, Hamed Khanpour, Corby Rosset, and Ahmed Awadallah. 2024. Orca-math: Unlocking the potential of slms in grade school math. *arXiv preprint arXiv:2402.14830*.
- OpenAI. 2022. [ChatGPT](#).
- OpenAI. 2024a. [GPT-4o mini: advancing cost-efficient intelligence](#).
- OpenAI. 2024b. [GPT-4o System Card](#).
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.
- Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy Lillicrap, Jean-baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.
- Zhiqing Sun, Longhui Yu, Yikang Shen, Weiyang Liu, Yiming Yang, Sean Welleck, and Chuang Gan. 2024. Easy-to-hard generalization: Scalable alignment beyond human supervision. *arXiv preprint arXiv:2403.09472*.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khoshabi, and Hannaneh Hajishirzi. 2023. [Self-instruct: Aligning language models with self-generated instructions](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.
- Peter West, Chandra Bhagavatula, Jack Hessel, Jena Hwang, Liwei Jiang, Ronan Le Bras, Ximing Lu, Sean Welleck, and Yejin Choi. 2022. [Symbolic knowledge distillation: from general language models to commonsense models](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4602–4625, Seattle, United States. Association for Computational Linguistics.
- Noam Wies, Yoav Levine, and Amnon Shashua. 2023. Sub-task decomposition enables learning in sequence to sequence tasks. In *The Eleventh International Conference on Learning Representations*.
- Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations*.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied Sanosi Saied, Weizhu Chen, and Nan Duan. 2023. [Agieval: A human-centric benchmark for evaluating foundation models](#). *ArXiv*, abs/2304.06364.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*.

A Hyperparameters for Training and Inference

As outlined in §4.1, we fine-tune Llama-3-8B base model using the elementary school and middle school-level math task annotations from the Orca-Math subset of NuminaMath-CoT, and MATH (level 1-3) to establish a base model equipped with foundational instruction-following and mathematical reasoning capabilities. In total, there are 164,162 training instances. Subsequently, we fine-tune this base model with our synthesized easy subtask and hard full task supervision. We set the learning rate to $2e-5$ and employed the cosine learning rate scheduler. Each model variant is trained on A100 and H100 GPUs for 2 epochs, taking around 1 hour. During inference on downstream tasks, we set temperature as 0.6, top-p as 0.9, top-k as 50 and maximal output token numbers as 1680.

B Supervision Source Data Details

Table 4 shows the original size, the number of tasks generated by the synthesis pipeline and the difficulty level of the five data sources for synthesizing hard task and easy subtask supervision. For the three subsets of NuminaMath-CoT, “cn_k12” consists of problems collected and translated from Chinese K-12 education math exercises. “synthetic_math” is synthesized from the MATH dataset following the methods in (Li et al., 2024a) and “synthetic_amc” subset is similarly synthesized based on the training split of the AMC-AIME dataset. Regarding the remaining two sources, PRM800K, initially a preference dataset, is utilized here by taking positive answers from the preference pairs as solutions to the problems. For MATH, we exclusively used its training split with difficulty level 4-5.

C Test Data Details

Our test data contains five subset and Table 5 shows the size and level of difficulty of downstream tasks in these five subsets. Our MATH test set is a small set split of size 500 from the original MATH test set, as mentioned in (Lightman et al., 2023).

JEE-Bench (Arora et al., 2023b) is derived from the IIT JEE-Advanced Exam, with high-school and college level problems. Some are multiple choice or fill-in-the-blank questions, we convert them to open-ended questions using GPT-4o-mini and filter out some unqualified converted problems like ill-defined ones or the ones with incorrect solutions.

Olympic-Arena (Huang et al., 2024) includes 11,163 text-only and interleaved text-image problems, covering multiple subjects like mathematics, physics, computer science, etc. We select the text-only math problems as our test data. Some problems are multiple choice or fill-in-the-blank questions, we also convert them to open-ended questions and manually filter out the unqualified ones, such as questions not well-defined or those with wrong solutions after conversion.

The other two subsets are sourced from the benchmark AGI-Eval (Zhong et al., 2023), SAT contains math problems from SAT exam in U.S. while Gaokao-Mathcloze is sourced from Chinese College Entrance Exam. For these two subsets, we also do format conversion and automatic filtering as described above.

D Details of Data Synthesis Pipeline and Examples for Demonstration

This section offers an in-depth demonstration of our pipeline through multiple examples, highlighting the effectiveness and comprehensiveness of our supervision synthesis approach discussed in §3.

D.1 Decomposing Hard Full Task into Subtasks

At Stage 1 of our pipeline (§3.2), we decompose the original hard full task into 2 or 3 subtasks. Here we show some examples of the original hard full tasks and the decomposed subtasks.

Decomposition Example 1: Table 8, 9 and 10 show the original hard full task, decomposed subtask 1 and decomposed subtask 2, respectively, as our first example.

Decomposition Example 2: Table 11, 12 and 13 show the original hard full task, decomposed subtask 1 and decomposed subtask 2, respectively, as our second example.

D.2 Filtering Unqualified Subtasks

As mentioned in §3.2, we use GPT-4o-mini to filter out the unqualified subtasks. Our filtering criteria are shown below:

- **Incomplete and Ill-defined Subtasks.** We filter out the decomposed subtasks that lack necessary conditions for solving problem, especially those where their solutions mention the information not present in the task. For example, if a solution mentions “ $f(x)$ is an odd function” or uses the

Sources	Original Sizes	# Hard Sup.	# Easy Sup.	Difficulty
NuminaMath-CoT (cn_k12)	105,785	22,397	35,210	high-school & college entrance exam
NuminaMath-CoT (synthetic_math)	40,580	11,172	18,107	high school
NuminaMath-CoT (synthetic_amc)	31,752	7,547	12,190	high school competition
PRM800k	2,041	340	523	high school
MATH (level 4-5)	1,040	150	261	high school competition

Table 4: Data sources used for constructing hard and easy task supervision, including their original sizes, final sizes after filtering (as described in §3.3) that the data sources contribute, and corresponding difficulty levels.

Test Tasks	Sizes	Difficulty
MATH	500	high school competition
JEE-Bench	163	high school & college
Olympic-Arena	213	Olympiad competition
SAT-Math	220	college entrance exam
Gaokao-Mathcloze	118	college entrance exam

Table 5: Detailed statistics of 5 downstream hard math reasoning tasks and their corresponding difficulty levels.

condition “Anna has three apples” but these conditions do not exist in the subtask, the subtask would not be utilized in the final supervision.

- **Overly Simple Subtasks.** We prompt GPT-4o-mini to count the computation and reasoning steps in each solution, filtering out easy subtasks that involve fewer than a threshold of 3 steps. These overly simple tasks, which require minimal reasoning, can already be handled by current LLMs and don’t need to be retaught. For instance, the discarded subtasks include “Calculate sine of 30° ”, “Solve for $2x + 6 = 18$ ” or “What’s the area of semicircle if its radius is 3?”, which are too simple and can be solved within 3 steps.
- **Subtasks That Are Too Similar to the Original Hard Full Tasks.** We prompt GPT-4o-mini to evaluate whether a decomposed subtask is merely a variation of the original full hard task, such as only changes in numerical values or conditions, and filter out such unqualified tasks to guarantee the lower difficulty of the subtasks. For example, if the original task is “Calculate the derivative of $f(x)$ and determine its monotonicity for $a = 1$,” and the subtask is “Determine whether $f(x)$ is increasing or decreasing for $a = 3$ by calculating its derivative,” the subtask is simply a variant of the original task and thus excluded.

Type 1 (Ill-defined) - Example 1. Table 14 shows the first example for Type 1 (ill-defined) of unqualified subtask.

Type 1 (Ill-defined) - Example 2. Table 15 shows the second example for Type 1 (ill-defined) of unqualified subtask.

Type 2 (Overly simple) - Example 1. Table 16 shows the first example for Type 2 (overly simple) of unqualified subtask.

Type 2 (Overly simple) - Example 2. Table 17 shows the second example for Type 2 (overly simple) of unqualified subtask.

Type 3 (Too similar to the original) - Example 1. Table 18 shows the first example for Type 3 (too similar to original hard task) of unqualified subtask.

Type 3 (Too similar to the original) - Example 2. Table 19 shows the second example for the type 3 (too similar to the original hard full task) of unqualified subtask.

D.3 Sampling Multiple Solutions for Both Hard and Subtasks

To adjust the outcome error rates in the supervision set during Stage 2 of the supervision synthesis process (§3.3), each task is paired with solutions that include both a ground-truth and an incorrect option. To obtain the incorrect solution, we sample three potential solutions for each hard full task and easy subtask in the supervision sets, then employ GPT-4o-mini to validate their correctness against the ground-truth final answer.

With regard to the correctness of the three sampled solutions for each hard full task and decomposed subtask, there are three situations: **Situation 1: All Correct** - All the three sampled solutions are correct; **Situation 2: Mixed** - The three sampled solutions contain both correct and incorrect solutions; **Situation 3: All Incorrect** - All the three sampled solutions are incorrect. In Situation 1, tasks without incorrect solutions are excluded from the final supervision set. In Situation 2, a correct sampled solution is used as the ground-truth, and an incorrect one is used for error rate adjustment. In Situation 3, an incorrect solution is selected and the ground truth is reformatted to match the style of incorrect solution. We ensure consistency between ground-truth and incorrect solutions to prevent LLMs from learning divergent styles during fine-tuning.

Once we construct the pairs of correct solution and incorrect solution, the incorrect solution will then be used to replace the ground-truth answer when we intentionally introduce more outcome errors to manage quality. Below, we provide examples that demonstrate the multi-sampling process and the creation of these solution pairs.

Sampling and Construction Example 1. Table 20 presents the first example of our process for sampling three solutions from LLMs and constructing solution pairs that include these three solutions and the ground-truth.

Sampling and Construction Example 2. Table 21 presents the second example of our process for sampling three solutions from LLMs and constructing solution pairs that include these three solutions and the ground-truth.

Sampling and Construction Example 3. Table 22 presents the third example of our process for sampling three solutions from LLMs and constructing solution pairs that include these three solutions and the ground-truth.

E Human Evaluation for SFT Supervision Quality

To further assess the quality of the synthesized data, we randomly sample a small set with 60 hard full tasks and their corresponding 87 easy subtasks, and conduct a human evaluation for these samples to examine data quality. Our evaluation suite includes the following dimensions: 1) checking correct solutions for both hard full tasks and subtasks; 2) checking incorrect solutions for both hard full tasks and subtasks; 3) checking the quality and appropriateness of the subtasks.

Correct Solution Checking. The correct solutions are required to be complete, correct and detailed, showing each essential computing and reasoning step. For hard full task, **59/60** are qualified and for easy subtask, **85/87** is qualified.

Incorrect Solution Checking. The incorrect solutions should reach incorrect final result and be detailed in solving process. **54/60** in hard full tasks is qualified while **83/87** in subtasks is qualified. Most of the unqualified solutions exhibit correct process and results, while very few are incomplete.

Decomposed Subtask Checking. As mentioned in §2.2, tasks in subtask supervision is decomposed from the tasks in hard full task supervision and we have filtered out three types of unqualified subtasks,

which corresponds to three essential features we expect in the subtask:

- *Criteria 1: Subtask should be well-defined, with no missing conditions.* In our review, **84/87** subtasks are qualified. Those unqualified are all missing conditions or information in the problem while the solution mentions and utilizes them.
- *Criteria 2: Subtask should not be overly simple, featuring less than 3 steps in solution.* In our review, **all the easy subtasks** are qualified.
- *Criteria 3: Subtask should not be too similar to the hard full task, with only some numerical conditions changed.* In this dimension, **86/87** easy subtasks are qualified.

With human evaluation, we further verify the overall quality of synthesized hard and easy data, which can be regarded as reliable foundations of our controlled supervision strategy analysis.

F Measure Possible Discrepancy between Hard Full tasks and Decomposed Subtasks

To investigate the potential discrepancies between hard full task questions and their decomposed subtask counterparts, which may be challenging for humans to detect, we measure the average cosine similarity of embeddings between these two task types on a small sample set. Additionally, we employ GPT-4o-mini to perform a style transfer on the question components of the sampled subtasks, aligning them with the full task questions in terms of descriptive manner, contextual setup, and overall style. The prompt used is shown in Table 6. Then we compare the similarity metric before and after this style transfer to further verify if there are some style differences between full task and subtask, which may potentially contribute to the performance gap between these two types of supervision data.

We sampled 200 full tasks, each paired with a subtask, to compute the embedding similarity using popular models such as bert-base-v2 from the BERT family and paraphrase-mpnet-base-v2 from the Sentence Transformer family. The results are presented in Table 7. The findings show that the cosine similarity between the embeddings of full tasks and their corresponding original subtasks remains consistently high, ranging from 0.7 to 0.85 across different embedding models. Additionally,

Below we provide some mathematical tasks, including several examples and one task T to be refined and modified in style.
Please modify the task T to make its style is as similar as possible to that of the examples, making the question type (if example tasks are not MCQ, then task T MUST NOT be MCQ), description manner, contextual setup, and the overall style are aligned closely. **BUT DO NOT CHANGE THE ORIGINAL MEANING OF THE TASK T** .
Input format:
Example Task 1: <example task 1>
Example Task 2: <example task 2>
Example Task 3: <example task 3>
Task to be refined: <task T to be modified>
Here we provide the examples and the task T to be modified, please ONLY output the modified version of task T, NO OTHER TEXTS!

Table 6: Prompt template for transferring style of the question part of subtasks to the style of full task in our measure of the style differences between two types of tasks.

after style transfer, the subtasks maintain the high similarity scores with the full tasks, with no significant differences compared to the original subtasks. These results further confirm that the gap between the two types of supervision data is marginal.

G Other Phenomenons Explained by Step-Wise Error Rates

In addition to the observations and conclusions regarding the correlation between step-wise error rates and downstream performance, several phenomena can be explained by this new metric:

To be Explained 1: When outcome error rates in hard full task supervision vary, performance remains relatively stable. As discussed in §4.2, GPT-4o-mini serves as the teacher model for hard full tasks and human evaluation in §5.3 shows GPT-4o-mini has a low step-wise error rates, with most errors involving just a single incorrect step, while the rest are accurate. This observation suggests that, despite the varying outcome error rates, the proportion of noisy supervision was minimal, which explains why models fine-tuned with hard full task supervision are robust to varying outcome error rates, as seen in §4.2, where performance remains stable across different outcome error rates on both MATH and SAT-Math.

To be Explained 2: Hard full task supervision consistently outperforms subtask supervision. We can imagine that, for example, where a complex task requires 10 steps, while a simpler task only needs 5 steps. If both tasks contain one erroneous step, the step-wise error rates for the subtask becomes proportionally higher. In line with this reasoning, our sample set shows that, for hard full tasks, there is an average of 1.36 erroneous steps

out of an average of 13.08 total steps, while subtasks have 1.14 erroneous steps out of 5.96 total steps. The number of erroneous steps is similar between hard full task and easy subtask supervision. This step-wise rate difference may become one of the possible factors about why easy subtask supervision tends to be less effective than hard full task supervision under identical outcome error rates, resulting in consistently lower performance.

H Case Study for Step-Wise Error Rate

As discussed in §4.2, models consistently perform better when trained with supervision from hard full tasks. They also demonstrate robustness against varying outcome error rates, even when these rates differ significantly, such as between 0% and 100%. This leads us to explore how other factors, such as the severity of erroneous solutions, impact supervision quality and downstream performance. To measure severity, we consider the metric, step-wise error rate. Defined in §5, the step-wise error rate represents the proportion of erroneous steps relative to the total number of steps. Importantly, we exclude steps that, while based on incorrect numerical values from prior errors, exhibit correct logic and reasoning. In this context, we focus solely on the logical correctness of each step.

We present case studies of sampled solutions from five different teacher models with varying reasoning capabilities: Command-R-2024-03, Command-R-2024-08, GPT-3.5-turbo, Llama-3-70B-Instruct, and GPT-4o-mini. Each case study includes annotations of the correctness of individual steps, along with explanations for any errors identified and step-wise error rate calculation.

Example 1 for Step-Wise Error Rate. Table 23, 24, 25, 26, 27 shows the first example of step-wise

Embedding Model	Sim(Full Task, Original Subtask)	Sim(Full Task, Style Transferred Subtask)	$ \delta $
albert-base	0.8311	0.8327	0.0016
bert-base-v2	0.8549	0.8535	0.0014
paraphrase-MiniLM-L12-v2	0.6970	0.6993	0.0023
paraphrase-mpnet-base-v2	0.7498	0.7493	0.0005

Table 7: Cosine similarity between the embeddings of full task and subtask text instructions, and the change in similarity after subtask style transfer. The marginal variation suggests that the subtask and full task instructions are already highly similar before applying any post-hoc methods to reduce the style disparity between them.

Hard task (full task) of Example 1:

In an arithmetic sequence $\{a_n\}$ where all terms are distinct, $a_1 = 1$, and a_1, a_2, a_5 form a geometric sequence. The sum of the first n terms of sequence $\{b_n\}$ is $S_n = 2^{n+1} - 2$. (1) Find the general formulas for sequences $\{a_n\}$ and $\{b_n\}$. (2) Let $c_n = 2^{a_n} + \log_2 b_n$, find the sum of the first n terms of sequence $\{c_n\}$ denoted as T_n .

Solution:

Part (1): Finding General Formulas

- Sequence $\{a_n\}$:

Given $a_1 = 1$, let the common difference be d . Thus, we have: $- a_2 = a_1 + d = 1 + d$ $- a_5 = a_1 + 4d = 1 + 4d$

Since a_1, a_2 , and a_5 form a geometric sequence, we apply the property that the square of the middle term equals the product of the first and third terms:

$$(a_1 + d)^2 = a_1(a_1 + 4d)$$

Substituting $a_1 = 1$ gives:

$$(1 + d)^2 = 1(1 + 4d)$$

Simplifying, we find:

$$1 + 2d + d^2 = 1 + 4d$$

$$d^2 - 2d = 0$$

$$d(d - 2) = 0$$

Thus, $d = 0$ or $d = 2$. Since $d = 0$ would imply no progression (which contradicts the distinct terms condition), we discard it and accept $d = 2$. Therefore, the general formula for $\{a_n\}$ is:

$$a_n = 1 + (n - 1)d = 1 + 2(n - 1) = 2n - 1$$

- Sequence $\{b_n\}$:

Given $S_n = 2^{n+1} - 2$, the n th term b_n can be found by subtracting the sum of the first $n - 1$ terms from the sum of the first n terms:

$$b_n = S_n - S_{n-1} = (2^{n+1} - 2) - (2^n - 2) = 2 \times 2^n - 2^n = 2^n$$

For $n = 1$, we directly verify that $b_1 = 2$ fits the formula. Thus, the general formula for $\{b_n\}$ is:

$$b_n = 2^n$$

Part (2): Sum of the First n Terms of Sequence $\{c_n\}$

Given $c_n = 2^{a_n} + \log_2 b_n$ and using the formulas derived in part (1), we have:

$$c_n = 2^{2n-1} + n$$

To find the sum T_n of the first n terms of $\{c_n\}$:

$$T_n = \sum_{k=1}^n (2^{2k-1} + k)$$

Breaking it down:

$$T_n = (2 + 2^3 + 2^5 + \dots + 2^{2n-1}) + (1 + 2 + 3 + \dots + n)$$

Using the formula for the sum of a geometric series and the sum of the first n natural numbers:

$$T_n = \frac{2(1 - 4^n)}{1 - 4} + \frac{n(n + 1)}{2}$$

Simplifying:

$$T_n = \frac{2^{2n+1} - 2}{3} + \frac{n^2 + n}{2}$$

Therefore, the sum of the first n terms of sequence $\{c_n\}$ is:

$$T_n = \frac{2^{2n+1} - 2}{3} + \frac{n^2 + n}{2}$$

Table 8: Decomposition - Example 1: Hard task.

Easy task 1 of Example 1:

Given an arithmetic sequence $\{a_n\}$ where $a_1 = 1$ and the common difference $d = 2$. Find the explicit formula for the n -th term a_n of the sequence.

Solution:

The general formula for an arithmetic sequence is given by:

$$a_n = a_1 + (n - 1)d$$

Substituting the known values, we get:

$$a_n = 1 + (n - 1) \cdot 2$$

Simplifying:

$$a_n = 1 + 2n - 2 = 2n - 1$$

Thus, the explicit formula for the n -th term of the sequence $\{a_n\}$ is:

$$\boxed{a_n = 2n - 1}$$

Table 9: Decomposition - Example 1: Easy task 1.

Easy task 2 of Example 1:

Given an arithmetic sequence $\{a_n\}$ where $a_1 = 1$ and the common difference $d = 2$. Find the explicit formula for the n -th term a_n of the sequence.

Solution:

To find the 5th term b_5 , we use the relationship between the sum of the first n terms and the n -th term:

$$b_n = S_n - S_{n-1}$$

Calculating S_5 and S_4 :

$$S_5 = 2^{5+1} - 2 = 2^6 - 2 = 64 - 2 = 62$$

$$S_4 = 2^{4+1} - 2 = 2^5 - 2 = 32 - 2 = 30$$

Now, substituting into the equation for b_5 :

$$b_5 = S_5 - S_4 = 62 - 30 = 32$$

Therefore, the 5th term of the sequence $\{b_n\}$ is:

$$\boxed{b_5 = 32}$$

Table 10: Decomposition - Example 1: Easy task 2.

Hard task (full task) of Example 2:

Find the range of real numbers m for which the quadratic equation $x^2 + (m - 1)x + 1 = 0$ has solutions in the interval $[0, 2]$.

Solution:

For the quadratic equation $x^2 + (m - 1)x + 1 = 0$ to have solutions in the interval $[0, 2]$, it must satisfy certain conditions related to its discriminant and the values of x at the endpoints of the interval.

First, let's consider the discriminant of the equation, $\Delta = (m - 1)^2 - 4 \cdot 1 \cdot 1 = m^2 - 2m - 3$.

For the equation to have real solutions, the discriminant must be non-negative, $\Delta \geq 0$.

This gives us $m^2 - 2m - 3 \geq 0$, which simplifies to $(m - 3)(m + 1) \geq 0$. This inequality holds for $m \leq -1$ or $m \geq 3$.

However, we also need to ensure that the solutions are within the interval $[0, 2]$. By evaluating the quadratic equation at the endpoints of the interval, we get:

1. For $x = 0$: $0^2 + (m - 1)0 + 1 = 1$, which is always positive.

2. For $x = 2$: $2^2 + (m - 1)2 + 1 = 4 + 2m - 2 + 1 = 2m + 3$, which needs to be non-negative for the solutions to lie within the interval. This gives us $2m + 3 \geq 0$, simplifying to $m \geq -\frac{3}{2}$.

Combining these conditions, we find that the range of m for which the quadratic equation has solutions in the interval $[0, 2]$ is $\boxed{(-\infty, -1]}$.

Table 11: Decomposition - Example 2: Hard task.

examination for the solutions to the same problem sampled from five supervision teacher models.

Example 2 for Step-Wise Error Rate. Table 28, 29, 30, 31, 32 shows the second example of step-wise examination for the solutions to the same prob-

lem sampled from five supervision teacher models.

Example 3 for Step-Wise Error Rate. Table 33, 34, 35, 36, 37 shows the third example of step-wise examination for the solutions to the same problem sampled from five supervision teacher models.

Easy task 1 of Example 2:

Determine the values of m for which the discriminant of the quadratic equation $x^2 + (m - 1)x + 1 = 0$ is non-negative.

Solution:

The discriminant Δ of the quadratic equation $x^2 + (m - 1)x + 1 = 0$ is given by:

$$\Delta = (m - 1)^2 - 4 \cdot 1 \cdot 1 = m^2 - 2m - 3.$$

To find the values of m for which the discriminant is non-negative, we need to solve the inequality:

$$m^2 - 2m - 3 \geq 0.$$

Factoring the quadratic expression yields:

$$(m - 3)(m + 1) \geq 0.$$

To solve this inequality, we can find the critical points where the expression equals zero:

1. $m - 3 = 0$ gives $m = 3$. 2. $m + 1 = 0$ gives $m = -1$.

Now we analyze the sign of the product $(m - 3)(m + 1)$:

- For $m < -1$, both factors are negative, hence the product is positive.

- For $-1 < m < 3$, one factor is negative and the other is positive, hence the product is negative.

- For $m > 3$, both factors are positive, hence the product is positive.

Thus, the solution to the inequality is:

$$m \leq -1 \quad \text{or} \quad m \geq 3.$$

The values of m for which the discriminant is non-negative are therefore:

$$(-\infty, -1) \cup (3, \infty).$$

Table 12: Decomposition - Example 2: Easy task 1.

Easy task 2 of Example 2:

Find the values of m such that the value of the quadratic equation $x^2 + (m - 1)x + 1 = 0$ evaluated at $x = 2$ is non-negative.

Solution:

Evaluating the quadratic equation at $x = 2$:

$$2^2 + (m - 1) \cdot 2 + 1 = 4 + 2(m - 1) + 1 = 2m + 3.$$

We want this expression to be non-negative:

$$2m + 3 \geq 0.$$

Solving for m :

$$2m \geq -3 \quad \Rightarrow \quad m \geq -\frac{3}{2}.$$

Thus, the values of m for which the quadratic evaluated at $x = 2$ is non-negative are:

$$\left[-\frac{3}{2}, \infty\right).$$

Table 13: Decomposition - Example 2: Easy task 2.

I Prompt Templates

We utilize multiple advanced LLMs in both supervision synthesis and quality controls. Here we show all prompt templates used in our experiments.

In Stage 1 of our supervision synthesis pipeline, we utilize GPT-4o-mini to decompose hard full tasks into 2 or 3 easy subtasks, as outlined in Table 38. Following this decomposition, we apply filtering process using GPT-4o-mini to eliminate unqualified subtasks, with the prompt in Table 39.

In Stage 2, we sample three solutions for each full hard full task and easy subtask, as shown in Table 40, and verify their correctness using the prompt as shown in Table 41. When construct-

ing pairs of correct and incorrect solutions, as discussed in §3.3, we employ style transfer to the ground-truth solution when all sampled solutions are incorrect, with the prompt for style transfer provided in Table 42, then we take the transferred version as the correct solution in the pair.

In §5, we study the severity of erroneous answers and utilize five different teacher models to sample solutions of hard full tasks. Here we use the same prompt templates as the data synthesis pipeline in §3 for both sampling (Table 40) and checking solutions (Table 41).

In §6, we explore the strategy of combining hard full tasks and easy subtasks, with two baselines of 1) training with doubled epochs and 2) training with

Unqualified easy task - Type 1 (Ill-defined) - Example 1

Task:

Let $b_n = L^{(2)}\left(\frac{25}{n}\right)$, where $L^{(2)}(x)$ denotes the second iterate of $L(x)$. Determine b_n as n approaches infinity

Solution:

First, we need to compute $L^{(2)}(x)$:

$$L^{(1)}(x) = \boxed{L(x) = x - \frac{x^2}{2}}.$$

Comment: The definition of $L(x)$ does not exist in the problem.

Now we apply L again to $L(x)$:

$$\boxed{L^{(2)}(x) = L(L(x))} = L\left(x - \frac{x^2}{2}\right).$$

Comment: It's NOT mentioned in the problem what's "iterate of $L(x)$ " and how to calculate it.

Substituting $x - \frac{x^2}{2}$ into L :

$$L\left(x - \frac{x^2}{2}\right) = \left(x - \frac{x^2}{2}\right) - \frac{1}{2}\left(x - \frac{x^2}{2}\right)^2.$$

Calculating $\left(x - \frac{x^2}{2}\right)^2$:

$$\left(x - \frac{x^2}{2}\right)^2 = x^2 - x^3 + \frac{x^4}{4}.$$

Now, substituting back:

$$L^{(2)}(x) = x - \frac{x^2}{2} - \frac{1}{2}\left(x^2 - x^3 + \frac{x^4}{4}\right),$$

which simplifies to:

$$L^{(2)}(x) = x - \frac{x^2}{2} - \frac{1}{2}x^2 + \frac{1}{2}x^3 - \frac{1}{8}x^4.$$

Combining terms gives:

$$L^{(2)}(x) = x - \frac{3}{4}x^2 + \frac{1}{2}x^3 - \frac{1}{8}x^4.$$

Now, we evaluate $b_n = L^{(2)}\left(\frac{25}{n}\right)$:

$$b_n = L^{(2)}\left(\frac{25}{n}\right) = \frac{25}{n} - \frac{3}{4}\left(\frac{25}{n}\right)^2 + \frac{1}{2}\left(\frac{25}{n}\right)^3 - \frac{1}{8}\left(\frac{25}{n}\right)^4.$$

Calculating each term:

$$1. \frac{25}{n} \quad 2. -\frac{3}{4} \cdot \frac{625}{n^2} = -\frac{1875}{4n^2} \quad 3. \frac{1}{2} \cdot \frac{15625}{n^3} = \frac{7812.5}{n^3} \quad 4. -\frac{1}{8} \cdot \frac{390625}{n^4} = -\frac{48828.125}{n^4}$$

As n approaches infinity, the higher-order terms vanish. Therefore, the leading term dominates:

$$b_n \approx \frac{25}{n}.$$

Thus, as n approaches infinity,

$$b_n \rightarrow 0.$$

The value that b_n approaches is $\boxed{0}$.

Table 14: Unqualified easy task - Type 1 (Ill-defined) - Example 1.

rephrased tasks merged with original data. The prompt template used for rephrasing in the second baseline is shown in Table 43.

Unqualified easy task - Type 1 (Ill-defined) - Example 2

Task:

Given triangle AMC with $AM = kx$, $AC = 3$, and $MC = \frac{x}{2}$, find the relationship between k , x , and $\cos(180^\circ - \theta)$.

Solution:

Using the Law of Cosines for triangle AMC :

$$\left(\frac{x}{2}\right)^2 + 3^2 - 2 \cdot \frac{x}{2} \cdot 3 \cdot \cos(180^\circ - \theta) = (kx)^2$$

Comment: Which angle of the triangle is $\cos(180^\circ - \theta)$ is not defined in the problem (we can infer from the formula above it's angle ACM). Substituting the cosine identity $\cos(180^\circ - \theta) = -\cos \theta$:

$$\frac{x^2}{4} + 9 + 3x \cos \theta = k^2 x^2$$

Rearranging gives:

$$3x \cos \theta = k^2 x^2 - 9 - \frac{x^2}{4}$$

Thus, isolating $\cos \theta$:

$$\cos \theta = \frac{k^2 x^2 - 9 - \frac{x^2}{4}}{3x}$$

This provides a relationship between k , x , and $\cos \theta$ in triangle AMC .

Conclusion:

$$\cos \theta = \frac{k^2 x^2 - 9 - \frac{x^2}{4}}{3x}$$

Table 15: Unqualified easy task - Type 1 (Ill-defined) - Example 2.

Unqualified easy task - Type 2 (Overly Simple) - Example 1

Task:

Given a quadratic function $f(x) = ax^2 + bx + c$ with $a > 0$ and $b > 0$, find the expression for the value of $f(1)$ in terms of a, b, c

Solution:

To find the expression for $f(1)$, we start with the quadratic function defined as $f(x) = ax^2 + bx + c$.

1. Substitute $x = 1$ into the function:

$$f(1) = a(1^2) + b(1) + c = a + b + c.$$

2. Therefore, the expression for $f(1)$ in terms of a, b , and c is:

$$f(1) = a + b + c.$$

Thus, the value of $f(1)$ is expressed as $a + b + c$.

Comment: the solution has only one step, which indicates this decomposed easy task is overly simple.

Table 16: Unqualified easy task - Type 2 (Overly Simple) - Example 1.

Unqualified easy task - Type 2 (Overly Simple) - Example 2

Task:

Find the expression for the first derivative of the function $f_0(x) = \frac{1}{x}$

Solution:

To find the first derivative $f_1(x)$ of the function $f_0(x) = \frac{1}{x}$, we can use the power rule for differentiation.

1. Rewrite the function: $f_0(x) = x^{-1}$. 2. Apply the power rule: The derivative of x^n is $n \cdot x^{n-1}$. Therefore, for $f_0(x)$:

$$f_1(x) = \frac{d}{dx}(x^{-1}) = -1 \cdot x^{-2} = -\frac{1}{x^2}.$$

Thus, the expression for the first derivative is:

$$f_1(x) = -\frac{1}{x^2}.$$

Comment: the solution has only two steps, which indicates this decomposed easy task is overly simple.

Table 17: Unqualified easy task - Type 2 (Overly Simple) - Example 2.

<p>Unqualified easy task - Type 3 (too similar to the original task) - Example 1</p> <hr/> <p>Hard Task (full task): Given the function $h(x) = 3 \ln(x) - x^3 + 1$ defined on the interval $[\frac{1}{e}, e]$, find the critical points of $h(x)$ and determine whether $h(x)$ has a maximum or minimum at these points.</p> <hr/> <p>Easy task 2 (unqualified, too similar): Evaluate the values of the function $h(x) = 3 \ln(x) - x^3 + 1$ at the endpoints of the interval $[\frac{1}{e}, e]$ and determine the minimum and maximum values of $h(x)$ over this interval.</p> <p><i>Comment: decomposed easy task is almost the same as hard task since they solve for the same things.</i></p> <hr/>

Table 18: Unqualified easy task - Type 3 (too similar to the original task) - Example 1.

<p>Unqualified easy task - Type 3 (too similar to the original task) - Example 2</p> <hr/> <p>Hard Task (full task): A conical hill stands with its base on level ground and has a total height of 10,000 feet. Given that the top one-tenth of the volume of the hill is above the cloud layer, what is the height of the cloud layer above the ground in feet?</p> <hr/> <p>Easy task 2 (unqualified, too similar): Given a conical hill with a total height of 10,000 feet, if the portion of the cone above the cloud layer represents one-tenth of the volume, what is the height of that portion above the cloud layer in feet?</p> <p><i>Comment: the decomposed easy task has the same difficulty of hard task, even though they solve for different things, their solutions are comparable.</i></p> <hr/>

Table 19: Unqualified easy task - Type 3 (too similar to the original task) - Example 2.

Sampling multiple solutions and constructing pairs - Example 1

Task (this example is from hard tasks):

Two circles intersect at points A(1, 3) and B(m, n), and both of their centers lie on the line $x - y - 2 = 0$. What is the value of $m + n$?

Ground Truth:

Given that the circles intersect at points A(1, 3) and B(m, n) and that the centers of both circles lie on the line $x - y - 2 = 0$, we can derive the slope of line segment AB as follows:

..... (process partly omitted here)

Substituting $n = -1$ back into equation (3) gives us:

$$m - (-1) = 6$$

$$m + 1 = 6$$

$$m = 5$$

Thus, we have found the coordinates B(m, n) to be B(5, -1). Now we calculate $m + n$:

$$m + n = 5 + (-1)$$

$$\boxed{m + n = 4}$$

Sampled solution 1: (Correct)

To solve the problem, we start by noting that the centers of both circles lie on the line given by the equation $x - y - 2 = 0$. We can rewrite this line in slope-intercept form as $y = x - 2$.

Let the centers of the two circles be denoted as $C_1(x_1, y_1)$ and $C_2(x_2, y_2)$. Since both centers lie on the line, we have:

..... (process partly omitted here)

Rearranging terms, we find:

$$n + m = 4$$

Thus, the value of $m + n$ is:

$$\boxed{4}$$

Sampled solution 2: (Incorrect)

Let the centers of the two circles be $C_1(h_1, k_1)$ and $C_2(h_2, k_2)$. Since both centers lie on the line $x - y - 2 = 0$, we can express the coordinates of the centers in terms of a single variable. The equation can be rewritten as $y = x - 2$. Therefore, we can say:

- For $C_1(h_1, k_1)$: $k_1 = h_1 - 2$ - For $C_2(h_2, k_2)$: $k_2 = h_2 - 2$

..... (process partly omitted here)

Thus,

$$m + n = 2 + 0 = 2.$$

Therefore, the value of $m + n$ is

$$\boxed{2}.$$

Sampled solution 3: (Incorrect)

Let the centers of the two circles be $C_1(h_1, k_1)$ and $C_2(h_2, k_2)$. Since both centers lie on the line $x - y - 2 = 0$, we can express their coordinates in terms of a single variable t :

..... (process partly omitted here)

$$n = 5 - 2 = 3.$$

Thus:

$$m + n = 5 + 3 = 8.$$

Therefore, the final result is:

$$\boxed{8}.$$

Comment: There are both correct and incorrect solutions in the samples, thus we choose sample 1 as correct solution in the pair and choose sample 3 (randomly chosed) as incorrect solution in the pair.

Table 20: Sampling multiple solutions and constructing pairs - Example 1.

Sampling multiple solutions and constructing pairs - Example 2

Task (this example is from easy tasks):

Solve the inequality $\frac{1-x}{x-5} \geq 1$ with respect to x

Ground Truth:

To solve the inequality $\frac{1-x}{x-5} \geq 1$, we follow these steps:

1. ****Given Inequality****: $\frac{1-x}{x-5} \geq 1$.
2. ****Transforming the inequality****: Rewrite as $\frac{1-x}{x-5} - 1 \geq 0$, which gives $\frac{1-x-(x-5)}{x-5} \geq 0$.

..... (process partly omitted here)

7. ****Conclusion****: The solution is in the interval $[3, 5)$, excluding $x = 5$.

Thus, the solution set is:

$$\{x \mid 3 \leq x < 5\}$$

Sampled solution 1: (Incorrect)

To solve the inequality $\frac{1-x}{x-5} \geq 1$, find a common denominator:

$$\frac{1-x}{x-5} \geq 1$$

Combine fractions:

..... (process partly omitted here)

3. $x > 5$: Choose $x = 6$ yields < 0 . Not a solution.

Therefore, solution is $\{3 < x < 5\}$

Sampled solution 2: (Incorrect)

To solve the inequality $\frac{1-x}{x-5} \geq 1$, find a common denominator:

Multiplying both sides by $(x-5)$:

$$1-x \geq x-5$$

Simplifying gives:

$$1+5 \geq 2x$$

Thus, solution is:

$$x \leq 3$$

Sampled solution 3: (Incorrect)

To solve the inequality $\frac{1-x}{x-5} \geq 1$, find common denominator:

Multiplying both sides by $x-5$:

$$1-x \geq x-5$$

Simplifying yields:

$$1 \geq 0, \text{ true for all } x.$$

Thus, solution is:

$$x \in \mathbb{R}$$

Ground truth with style transferred: (Reference)

To solve the inequality $\frac{1-x}{x-5} \geq 1$, we start by reformulating it: 1. Start with the inequality:

$$\frac{1-x}{x-5} \geq 1.$$

..... (process partly omitted here)

6. We will now test the sign of $(2x-6)(x-5)$ across the intervals created by these points: - For $x < 3$: both factors $2x-6 < 0$ and $x-5 < 0$ yield a positive product. - For $3 < x < 5$: $2x-6 < 0$ and $x-5 < 0$ yield a positive product. - For $x > 5$: both factors $2x-6 > 0$ and $x-5 > 0$ yield a positive product.

7. Evaluating the overall expression, we find it is non-positive exactly in the interval $[3, 5)$, since at $x = 3$ it equals zero, and we exclude $x = 5$ due to the denominator. Thus, the solution set is: $[3, 5)$.
-

Comment: All the samples are incorrect, thus we convert the style of ground truth to the style of sampled solution and take transferred ground truth as correct solution and choose sample 2 (randomly chosen) as incorrect solution in the pair.

Table 21: Sampling multiple solutions and constructing pairs - Example 2.

Sampling multiple solutions and constructing pairs - Example 3

Task (this example is from easy tasks):

Given an arithmetic sequence $\{a_n\}$ where $a_1 = 1$, $a_2 = b_2$, and $a_4 + 2 = b_3$, if the common difference of the sequence is d , find the value of d if it is known that $a_4 = 7$.

Ground Truth:

Since we know that $a_1 = 1$ and the common difference is d , we can express the terms of the arithmetic sequence as follows: - $a_2 = a_1 + d = 1 + d$ - $a_4 = a_1 + 3d = 1 + 3d$ Given $a_4 = 7$, we can set up the equation:

$$1 + 3d = 7$$

Subtracting 1 from both sides gives:

$$3d = 6$$

Dividing both sides by 3 yields:

$$d = 2$$

Thus, the value of the common difference d is 2

Sampled solution 1: (Correct)

The general formula for an arithmetic sequence is $a_n = a_1 + (n - 1)d$, where a_n is the n th term, a_1 is the first term, d is the common difference, and n is the term number.

Given $a_1 = 1$, $a_2 = b_2$, and $a_4 = 7$, we can write the following equations: 1. $a_2 = a_1 + d = 1 + d = b_2$ 2. $a_4 = a_1 + 3d = 1 + 3d = 7$.

From equation 2, we can solve for d : $1 + 3d = 7$, $3d = 6$, $d = 2$

Therefore, the common difference of the arithmetic sequence is 2

Sampled solution 2: (Correct)

Since the sequence is arithmetic, we know that $a_n = a_1 + (n - 1)d$. We are given that $a_4 = 7$, so we can substitute $n = 4$ into the formula: $a_4 = a_1 + (4 - 1)d = 1 + 3d = 7$.

Solving for d : $3d = 6$, $d = 2$

Therefore, the common difference of the sequence is 2

Sampled solution 3: (Correct)

The general formula for an arithmetic sequence is $a_n = a_1 + (n - 1)d$, where a_n is the n th term, a_1 is the first term, and d is the common difference.

Given that $a_1 = 1$ and $a_4 = 7$, we can write: $a_4 = a_1 + 3d = 7$

Substitute $a_1 = 1$ into the equation:

$$1 + 3d = 7 \text{ Solving for } d: 3d = 6, d = 2$$

Therefore, the common difference d is 2

Comment: All the sampled solutions are correct, in this case we skip and discard this example.

Table 22: Sampling multiple solutions and constructing pairs - Example 3.

Example 1 for step-wise error ratio - "Trigonometric identity transformations": Supervision sampled by **Command-R-2024-03**

Task:

Given $\sin\left(\alpha - \frac{\pi}{6}\right) = \frac{1}{3}$, then $\sin\left(2\alpha - \frac{\pi}{6}\right) + \cos 2\alpha = ()$

A: $-\frac{2}{3}$, B: $\frac{2}{3}$, C: $-\frac{7}{9}$, D: $\frac{7}{9}$

Solution:

We can write: $\sin\left(2\alpha - \frac{\pi}{6}\right) = \sin\left(\alpha - \frac{\pi}{3} + \alpha\right)$ (Step 1: wrong, it should be " $\sin\left(\alpha - \frac{\pi}{6} + \alpha\right)$ ")

$= \sin\left(\alpha - \frac{\pi}{3}\right) \cdot \cos \alpha + \cos\left(\alpha - \frac{\pi}{3}\right) \cdot \sin \alpha$ (Step 2: correct)

$= \sin \alpha \cdot \cos \alpha - \frac{1}{3} \cdot \sin \alpha$ (Step 3: wrong, it incorrectly take $\sin\left(\alpha - \frac{\pi}{3}\right)$ as $\sin \alpha$, and take $\cos\left(\alpha - \frac{\pi}{3}\right)$ as $-\frac{1}{3}$)

$= \frac{2}{3}$ (Step 4: wrong, the value of above equation can not be calculated since $\sin \alpha$ and $\cos \alpha$ are not known.)

Therefore, the final answer is B.

Comment:

There are 4 steps totally while 3 of them are wrong, step-wise error ratio is **75.0%**.

Table 23: Example 1 for step-wise error ratio - "Trigonometric identity transformations": Supervision sampled by **Command-R-2024-03**.

Example 1 for step-wise error ratio - “Trigonometric identity transformations”: Supervision sampled by **Command-R-2024-08**

Task:

Given $\sin(\alpha - \frac{\pi}{6}) = \frac{1}{3}$, then $\sin(2\alpha - \frac{\pi}{6}) + \cos 2\alpha = ()$

A: $-\frac{2}{3}$, B: $\frac{2}{3}$, C: $-\frac{7}{9}$, D: $\frac{7}{9}$

Solution:

Let’s think step by step to solve this trigonometry problem.

We are given that $\sin(\alpha - \frac{\pi}{6}) = \frac{1}{3}$. (Step 1: correct)

We can use the double-angle formula for sine to rewrite $\sin(2\alpha - \frac{\pi}{6})$ as:

$\sin(2\alpha - \frac{\pi}{6}) = \sin(2\alpha) \cos(\frac{\pi}{6}) - \cos(2\alpha) \sin(\frac{\pi}{6})$ (Step 2: correct)

Now, we can substitute $\sin(\alpha - \frac{\pi}{6}) = \frac{1}{3}$:

$\sin(\alpha) \cos(\frac{\pi}{6}) = \frac{1}{3} \cos(\frac{\pi}{6}) = \frac{1}{3} \cdot \frac{1}{2} = \frac{1}{6}$ (Step 3: wrong, $\cos(\frac{\pi}{6})$ is $\frac{\sqrt{3}}{2}$, not $\frac{1}{2}$)

$\cos^2(\alpha) = 1 - \sin^2(\alpha) = 1 - \frac{1}{9} = \frac{8}{9}$ (Step 4: correct)

$\sin^2(\alpha) = \frac{1}{9}$ (Step 5: correct)

Substituting these values back into our equation:

$\sin(2\alpha - \frac{\pi}{6}) = \frac{1}{6} - \frac{1}{2} \cdot \frac{8}{9} + \frac{1}{2} \cdot \frac{1}{9}$ (Step 6: correct, this step substitutes values into equation of Step 2, $\sin(2\alpha) \cos(\frac{\pi}{6}) = 2 \sin(\alpha) \cos(\alpha) \cos(\frac{\pi}{6})$, since Step 3 mistakenly take $\cos(\frac{\pi}{6})$ as $\cos(\frac{1}{2})$, so here $\sin(2\alpha) \cos(\frac{\pi}{6}) = 2 \sin(\alpha) \cos(\alpha) \frac{1}{2} = \sin(\alpha) \cos(\alpha) = \frac{1}{6}$. The error is due to previous wrong numerical values, not this step.)

$= \frac{1}{6} - \frac{4}{9} + \frac{1}{18}$ (Step 7: correct)

$= \frac{1}{6} - \frac{11}{18}$ (Step 8: wrong, $-\frac{4}{9} + \frac{1}{18} = -\frac{7}{18}$, not $-\frac{11}{18}$)

$= \frac{3-11}{18}$ (Step 9: correct, Step 9 carries wrong values from Step 8 but its calculation is correct, thus this step is not included in “erroneous steps”)

$= -\frac{8}{18}$ (Step 10: correct)

$= -\frac{2}{3}$ (Step 11: wrong, $-\frac{8}{18} = -\frac{4}{9}$, not $-\frac{2}{3}$)

So, the value of $\sin(2\alpha - \frac{\pi}{6}) + \cos(2\alpha)$ is $-\frac{2}{3}$.

Therefore, the correct answer is **A: $-\frac{2}{3}$ **.

Comment:

There are 11 steps totally while 3 of them are wrong, step-wise error ratio is **27.3%**.

Table 24: Example 1 for step-wise error ratio - “Trigonometric identity transformations”: Supervision sampled by **Command-R-2024-08**.

Example 1 for step-wise error ratio - "Trigonometric identity transformations": Supervision sampled by **GPT-3.5-turbo**

Task:

Given $\sin(\alpha - \frac{\pi}{6}) = \frac{1}{3}$, then $\sin(2\alpha - \frac{\pi}{6}) + \cos 2\alpha = ()$

A: $-\frac{2}{3}$, B: $\frac{2}{3}$, C: $-\frac{7}{9}$, D: $\frac{7}{9}$

Solution:

Using the angle subtraction formula for sine, we have:

$$\sin(\alpha) \cos(\frac{\pi}{6}) - \cos(\alpha) \sin(\frac{\pi}{6}) = \frac{1}{3} \text{ (Step 1: correct)}$$

$$\sin(\alpha) \frac{\sqrt{3}}{2} - \cos(\alpha) \frac{1}{2} = \frac{1}{3} \text{ (Step 2: correct)}$$

$$\sqrt{3} \sin(\alpha) - \cos(\alpha) = \frac{2}{3} \dots (1) \text{ (Step 3: correct)}$$

Now, we need to find $\sin(2\alpha)$ and $\cos(2\alpha)$:

$$\sin(2\alpha) = 2 \sin(\alpha) \cos(\alpha) \text{ (Step 4: correct)}$$

$$\cos(2\alpha) = 1 - 2 \sin^2(\alpha) \text{ (Step 5: correct)}$$

Substitute $\sin(\alpha)$ from equation (1):

$$\sin(2\alpha) = 2 \left(\frac{\sqrt{3} \sin(\alpha) - \frac{2}{3}}{\sqrt{3}} \right) \cos(\alpha) \text{ (Step 6: wrong, if substituting } \sin(\alpha) \text{ into equation (1), it should be } 2 \left(\frac{\frac{2}{3} + \cos(\alpha)}{\sqrt{3}} \right) \cos(\alpha))$$

$$= \sqrt{3} \sin(\alpha) \cos(\alpha) - \frac{2}{3} \cos(\alpha) \text{ (Step 7: correct)}$$

$$= \sqrt{3} \cdot \frac{2}{3} - \frac{2}{3} \cos(\alpha) \text{ (Step 8: wrong, } \sin \alpha \cos \alpha \text{ is not } \frac{2}{3}).$$

$$= 2\sqrt{3} - 2 \cos(\alpha) \text{ (2) (Step 9: wrong, it should be } \frac{2\sqrt{3}}{3} - \frac{2}{3} \cos(\alpha))$$

Now, the given expression $\sin(2\alpha - \frac{\pi}{6}) + \cos(2\alpha)$:

$$\sin(2\alpha) \cos(\frac{\pi}{6}) - \cos(2\alpha) \sin(\frac{\pi}{6}) + 1 - 2 \sin^2(\alpha) \text{ (Step 10: correct)}$$

Substitute $\sin(2\alpha)$ from equation (2):

$$(2\sqrt{3} - 2 \cos(\alpha)) \frac{\sqrt{3}}{2} - 1 - 2 \sin^2(\alpha) \text{ (Step 11: wrong, it should be } (2\sqrt{3} - 2 \cos(\alpha)) \frac{\sqrt{3}}{2} + \frac{1}{2}(1 - 2 \sin^2(\alpha)))$$

$$= 3 - \sqrt{3} \cos(\alpha) - 1 - 2 \left(\frac{\sqrt{3} \sin(\alpha) - \frac{2}{3}}{\sqrt{3}} \right)^2 \text{ (Step 12: wrong, it should be } 3 - \sqrt{3} \cos(\alpha) - 1 - 2 \sin^2(\alpha))$$

$$= 2 - \sqrt{3} \cos(\alpha) - \sqrt{3} \sin(\alpha) + \frac{4}{3} \text{ (Step 13: wrong, it should be } 2 - \sqrt{3} \cos(\alpha) - \frac{2\sqrt{3}}{3} \sin(\alpha) + \frac{4}{3})$$

$$= 2 - \frac{2\sqrt{3}}{3} + \frac{4}{3} \text{ (Step 14: wrong, } \sin(\alpha) \text{ and } \cos(\alpha) \text{ are not } \frac{1}{3})$$

$$= 2 - \frac{2}{3} \text{ (Step 15: wrong, } -\frac{2\sqrt{3}}{3} + \frac{4}{3} \neq -\frac{2}{3})$$

$$= \frac{4}{3} \text{ (Step 16: correct)}$$

Therefore, the final result is $\frac{4}{3}$, which is not one of the provided answer choices.

Comment:

There are 16 steps totally while 8 of them are wrong, step-wise error ratio is **50.0%**.

Table 25: Example 1 for step-wise error ratio - "Trigonometric identity transformations": Supervision sampled by **GPT-3.5-turbo**.

Example 1 for step-wise error ratio - "Trigonometric identity transformations": Supervision sampled by **Llama-3-70B-Instruct**

Task:

Given $\sin(\alpha - \frac{\pi}{6}) = \frac{1}{3}$, then $\sin(2\alpha - \frac{\pi}{6}) + \cos 2\alpha = ()$

A: $-\frac{2}{3}$, B: $\frac{2}{3}$, C: $-\frac{7}{9}$, D: $\frac{7}{9}$

Solution:

To solve this problem, we'll use the identities for the sine of a sum and the cosine of a double angle.

Given: $\sin(\alpha - \frac{\pi}{6}) = \frac{1}{3}$

We need to find: $\sin(2\alpha - \frac{\pi}{6}) + \cos 2\alpha$

First, let's find $\sin(2\alpha - \frac{\pi}{6})$ using the sine of a sum identity:

$$\sin(2\alpha - \frac{\pi}{6})$$

$$= \sin(\alpha - \frac{\pi}{6} + \alpha) \text{ (Step 1: correct)}$$

$$= \sin(\alpha - \frac{\pi}{6}) \cos \alpha + \cos(\alpha - \frac{\pi}{6}) \sin \alpha \text{ (Step 2: correct)}$$

We are given $\sin(\alpha - \frac{\pi}{6}) = \frac{1}{3}$. Let's find $\cos(\alpha - \frac{\pi}{6})$: (Step 3: correct)

$$\cos(\alpha - \frac{\pi}{6})$$

$$= \pm \sqrt{1 - \sin^2(\alpha - \frac{\pi}{6})} \text{ (Step 4: correct)}$$

$$= \pm \sqrt{1 - \frac{1}{9}} \text{ (Step 5: correct)}$$

$$= \pm \frac{2\sqrt{2}}{3} \text{ (Step 6: correct)}$$

We'll choose the positive value for $\cos(\alpha - \frac{\pi}{6})$ since we don't know the quadrant of $\alpha - \frac{\pi}{6}$. (Step 7: wrong, since we don't know the quadrant of $\alpha - \frac{\pi}{6}$, we cannot decide whether $\cos(\alpha - \frac{\pi}{6})$ is positive or not.)

Now, let's find $\sin \alpha$ and $\cos \alpha$:

$$\sin \alpha$$

$$= \sin(\frac{\pi}{6} + (\alpha - \frac{\pi}{6})) \text{ (Step 8: correct)}$$

$$= \sin \frac{\pi}{6} \cos(\alpha - \frac{\pi}{6}) + \cos \frac{\pi}{6} \sin(\alpha - \frac{\pi}{6}) \text{ (Step 9: correct)}$$

$$= \frac{\sqrt{3}}{2} \times \frac{2\sqrt{2}}{3} + \frac{1}{2} \times \frac{1}{3} \text{ (Step 10: wrong, } \sin \frac{\pi}{6} \neq \frac{\sqrt{3}}{2} \text{ and } \cos \frac{\pi}{6} \neq \frac{1}{2} \text{)}$$

$$= \frac{\sqrt{6}+1}{6} \text{ (Step 11: wrong, it should be } \frac{2\sqrt{6}+1}{6} \text{)}$$

$$\cos \alpha = \cos(\frac{\pi}{6} + (\alpha - \frac{\pi}{6})) \text{ (Step 12: correct)}$$

$$= \cos \frac{\pi}{6} \cos(\alpha - \frac{\pi}{6}) - \sin \frac{\pi}{6} \sin(\alpha - \frac{\pi}{6}) \text{ (Step 13: correct)}$$

$$= \frac{\sqrt{3}}{2} \times \frac{2\sqrt{2}}{3} - \frac{1}{2} \times \frac{1}{3} \text{ (Step 14: correct)}$$

$$= \frac{\sqrt{6}-1}{6} \text{ (Step 15: wrong, it should be } \frac{2\sqrt{6}-1}{6} \text{)}$$

Now we can find $\sin(2\alpha - \frac{\pi}{6})$:

$$\sin(2\alpha - \frac{\pi}{6})$$

$$= \frac{1}{3} \times \frac{\sqrt{6}-1}{6} + \frac{2\sqrt{2}}{3} \times \frac{\sqrt{6}+1}{6} \text{ (Step 16: correct)}$$

$$= \frac{\sqrt{6}-1+4\sqrt{2}+4}{18} \text{ (Step 17: wrong, it should be } \frac{\sqrt{6}-1+4\sqrt{3}+2\sqrt{2}}{18} \text{)}$$

Next, let's find $\cos 2\alpha$ using the cosine of a double angle identity:

$$\cos 2\alpha$$

$$= \cos^2 \alpha - \sin^2 \alpha \text{ (Step 18: correct)}$$

$$= \left(\frac{\sqrt{6}-1}{6}\right)^2 - \left(\frac{\sqrt{6}+1}{6}\right)^2 \text{ (Step 19: correct)}$$

$$= \frac{6-2\sqrt{6}+1}{36} - \frac{6+2\sqrt{6}+1}{36} \text{ (Step 20: correct)}$$

$$= -\frac{2\sqrt{6}}{18} \text{ (Step 21: correct)}$$

Now, we can find the final answer:

$$\sin(2\alpha - \frac{\pi}{6}) + \cos 2\alpha$$

$$= \frac{\sqrt{6}-1+4\sqrt{2}+4}{18} - \frac{2\sqrt{6}}{18} \text{ (Step 22: correct)}$$

$$= \frac{\sqrt{6}-1+4\sqrt{2}+4-2\sqrt{6}}{18} \text{ (Step 23: correct)}$$

$$= \frac{4+4\sqrt{2}-3\sqrt{6}-1}{18} \text{ (Step 24: wrong, it should be } \frac{4+4\sqrt{2}-\sqrt{6}-1}{18} \text{)}$$

$$= -\frac{7}{9} \text{ (Step 25: wrong, it should be } \frac{3+4\sqrt{2}-\sqrt{6}}{18} \text{)}$$

Final Result: $\boxed{-\frac{7}{9}}$

Comment:

There are 25 steps totally while 7 of them are wrong, step-wise error ratio is **28.0%**.

Table 26: Example 1 for step-wise error ratio - "Trigonometric identity transformations": Supervision sampled by **Llama-3-70B-Instruct**.

Example 1 for step-wise error ratio - “Trigonometric identity transformations”: Supervision sampled by **GPT-4o-mini**

Task:

Given $\sin\left(\alpha - \frac{\pi}{6}\right) = \frac{1}{3}$, then $\sin\left(2\alpha - \frac{\pi}{6}\right) + \cos 2\alpha = ()$

A: $-\frac{2}{3}$, B: $\frac{2}{3}$, C: $-\frac{7}{9}$, D: $\frac{7}{9}$

Solution:

Given the equation:

$$\sin\left(\alpha - \frac{\pi}{6}\right) = \frac{1}{3}$$

We can use the sine addition formula to express $\sin(2\alpha - \frac{\pi}{6})$:

$$\sin(2\alpha - \frac{\pi}{6}) = \sin(2\alpha) \cos\left(\frac{\pi}{6}\right) - \cos(2\alpha) \sin\left(\frac{\pi}{6}\right) \text{ (Step 1: correct)}$$

Since $\cos\left(\frac{\pi}{6}\right) = \frac{\sqrt{3}}{2}$ and $\sin\left(\frac{\pi}{6}\right) = \frac{1}{2}$, the equation becomes: (Step 2: correct)

$$\sin(2\alpha - \frac{\pi}{6}) = \sin(2\alpha) \cdot \frac{\sqrt{3}}{2} - \cos(2\alpha) \cdot \frac{1}{2} \text{ (Step 3: correct)}$$

Next, we need to find $\sin(2\alpha)$ and $\cos(2\alpha)$. We can use the double angle formulas:

$$\sin(2\alpha) = 2 \sin(\alpha) \cos(\alpha) \text{ (Step 4: correct)}$$

$$\cos(2\alpha) = \cos^2(\alpha) - \sin^2(\alpha) \text{ (Step 5: correct)}$$

To find $\sin(\alpha)$ and $\cos(\alpha)$, we can use the identity:

$$\sin^2(\alpha) + \cos^2(\alpha) = 1 \text{ (Step 6: correct)}$$

From $\sin(\alpha - \frac{\pi}{6}) = \frac{1}{3}$, we can rewrite:

$$\sin(\alpha) = \sin\left(\alpha - \frac{\pi}{6}\right) \cos\left(\frac{\pi}{6}\right) + \cos\left(\alpha - \frac{\pi}{6}\right) \sin\left(\frac{\pi}{6}\right) \text{ (Step 7: correct)}$$

Let $x = \sin(\alpha)$ and $y = \cos(\alpha)$, we have:

$$x - \frac{1}{2}y = \frac{1}{3} \cdot \frac{\sqrt{3}}{2} \text{ (Step 8: wrong, it should be } x = \frac{1}{3} \cdot \frac{\sqrt{3}}{2} + \frac{1}{2} \cos\left(\alpha - \frac{\pi}{6}\right), \text{ i.e., } \cos\left(\alpha - \frac{\pi}{6}\right) \neq y)$$

Now we can express $\sin^2(\alpha) + \cos^2(\alpha) = 1$:

$$x^2 + y^2 = 1 \text{ (Step 9: correct)}$$

We substitute y from the first equation into the second. Solving these will yield x and y .

Instead, let's calculate $\sin(2\alpha - \frac{\pi}{6}) + \cos(2\alpha)$:

Using $\sin(2\alpha) = 2xy$ and $\cos(2\alpha) = y^2 - x^2$: (Step 10: correct)

$$\sin(2\alpha - \frac{\pi}{6}) + \cos(2\alpha) = \left(2xy \cdot \frac{\sqrt{3}}{2} - (y^2 - x^2) \cdot \frac{1}{2}\right) + (y^2 - x^2) \text{ (Step 11: correct)}$$

Simplifying:

$$= \sqrt{3}xy - \frac{1}{2}(y^2 - x^2) + (y^2 - x^2) \text{ (Step 12: correct)}$$

$$= \sqrt{3}xy + \frac{1}{2}(y^2 - x^2) \text{ (Step 13: correct)}$$

Next, substituting $x = \frac{1}{3}$ to find y : (Step 14: wrong, $x = \sin \alpha \neq \frac{1}{3}$)

$$y = \sqrt{1 - \left(\frac{1}{3}\right)^2} = \sqrt{\frac{8}{9}} = \frac{2\sqrt{2}}{3} \text{ (Step 15: wrong, } \sqrt{\frac{8}{9}} = \pm \frac{2\sqrt{2}}{3})$$

Now we can compute the final value:

$$= \sqrt{3} \left(\frac{1}{3} \cdot \frac{2\sqrt{2}}{3}\right) + \frac{1}{2} \left(\left(\frac{2\sqrt{2}}{3}\right)^2 - \left(\frac{1}{3}\right)^2\right) \text{ (Step 16: correct)}$$

Calculating the expression yields:

$$= \sqrt{3} \left(\frac{2\sqrt{2}}{9}\right) + \frac{1}{2} \left(\frac{8}{9} - \frac{1}{9}\right) \text{ (Step 17: correct)}$$

$$= \frac{2\sqrt{6}}{9} + \frac{7}{18} \text{ (Step 18: correct)}$$

Comparing with the options, we find:

$$\sin(2\alpha - \frac{\pi}{6}) + \cos(2\alpha) = -\frac{2}{3} \text{ (Step 19: wrong, this estimation based on options is not reasonable.)}$$

Thus, the answer is:

$$\boxed{-\frac{2}{3}}$$

Comment:

There are 19 steps totally while 4 of them are wrong, step-wise error ratio is **21.1%**.

Table 27: Example 1 for step-wise error ratio - “Trigonometric identity transformations”: Supervision sampled by **GPT-4o-mini**.

Example 2 for step-wise error ratio - “Absolute value of Numbers”: Supervision sampled by **Command-R-2024-03**

Task:

Given that $a, b, c,$ and d are nonzero real numbers, find all possible values of the expression $\frac{a}{|a|} + \frac{b}{|b|} + \frac{c}{|c|} + \frac{d}{|d|} + \frac{abcd}{|abcd|}$.

Solution:

We can rewrite the expression as

$$\begin{aligned} & \frac{a}{|a|} + \frac{b}{|b|} + \frac{c}{|c|} + \frac{d}{|d|} + \frac{abcd}{|abcd|} \\ &= \frac{|a| + |b| + |c| + |d| + |abcd|}{|abcd|} \quad (\text{Step 1: wrong, the process of making common denominator of fractions is written wrong}) \\ &= \frac{|a| + |b| + |c| + |d|}{|abcd|} \quad (\text{Step 2: wrong, term } |abcd| \text{ is missing.}) \\ &= \frac{|a+b+c+d|}{|abcd|}. \quad (\text{Step 3: wrong, } |a| + |b| + |c| + |d| \neq |abcd|) \end{aligned}$$

We have four cases to consider, depending on the sign of the expression $a + b + c + d$.

1. If $a + b + c + d > 0$, then the numerator is positive, and so is the entire expression. (Step 4: wrong)

Therefore, we have

$$\frac{a}{|a|} + \frac{b}{|b|} + \frac{c}{|c|} + \frac{d}{|d|} + \frac{abcd}{|abcd|} = \boxed{1}. \quad (\text{Step 5: wrong, the calculation result is not 1.})$$

2. If $a + b + c + d < 0$, then the numerator is negative, and the expression is also negative. (Step 6: wrong, since both the numerator and denominator in Step 3 are absolute values, it's always non-negative.)

So we can write

$$\frac{a}{|a|} + \frac{b}{|b|} + \frac{c}{|c|} + \frac{d}{|d|} + \frac{abcd}{|abcd|} = -\frac{|-(a+b+c+d)|}{|abcd|} = \boxed{-1}. \quad (\text{Step 7: wrong, the calculation result is not } -1)$$

3. If $a + b = -c - d$, the expression is undefined as it involves dividing by zero. (Step 8: wrong, the denominator is $|abcd|$, it's not 0 when $a + b = -c - d$.)

4. If $a + b \neq -c - d$, we can write

$$\begin{aligned} & \frac{a}{|a|} + \frac{b}{|b|} + \frac{c}{|c|} + \frac{d}{|d|} + \frac{abcd}{|abcd|} \\ &= \frac{|a+b| + |c+d|}{|abcd|} \quad (\text{Step 9: wrong, the process of making common denominator of fractions is written wrong}) \\ &\geq \frac{|a+b-(c+d)|}{|abcd|} \quad (\text{Step 10: wrong, } |a+b| + |c+d| \neq |a+b-(c+d)|) \\ &= \frac{|(a+b)-(-c-d)|}{|abcd|} \quad (\text{Step 11: wrong, } |a+b-(c+d)| \neq |a+b-(-c-d)|) \\ &= \frac{2 \max\{|a+b|, |c+d|\}}{|abcd|} \quad (\text{Step 12: wrong, } |a+b-(-c-d)| \neq 2 \max\{|a+b|, |c+d|\}) \\ &\geq \frac{2}{\min\{|abcd|\}}. \quad (\text{Step 13: wrong, this result is not reasonable at all.}) \end{aligned}$$

Equality occurs when $a + b = \pm(c + d)$, and both $a + b$ and $c + d$ have the same sign. (Step 14: wrong)

$$\text{The answer is } \frac{a}{|a|} + \frac{b}{|b|} + \frac{c}{|c|} + \frac{d}{|d|} + \frac{abcd}{|abcd|} \geq \boxed{\frac{2}{\min\{|abcd|\}}}$$

Comment:

There are 14 steps totally while 14 of them are wrong, step-wise error ratio is **100.0%**.

Table 28: Example 2 for step-wise error ratio - “Absolute value of Numbers”: Supervision sampled by **Command-R-2024-03**.

Example 2 for step-wise error ratio - “Absolute value of Numbers”: Supervision sampled by **Command-R-2024-08**

Task:

Given that $a, b, c,$ and d are nonzero real numbers, find all possible values of the expression $\frac{a}{|a|} + \frac{b}{|b|} + \frac{c}{|c|} + \frac{d}{|d|} + \frac{abcd}{|abcd|}$.

Solution:

Let's consider the possible cases for the signs of $a, b, c,$ and d .

1. All of $a, b, c,$ and d are positive:

In this case, $\frac{a}{|a|} = 1, \frac{b}{|b|} = 1, \frac{c}{|c|} = 1, \frac{d}{|d|} = 1,$ and $\frac{abcd}{|abcd|} = 1.$ (Step 1: correct)

So, the expression evaluates to $1 + 1 + 1 + 1 + 1 = 5.$ (Step 2: correct)

2. One of $a, b, c,$ or d is negative, and the rest are positive:

Without loss of generality, let's assume a is negative and $b, c,$ and d are positive. (Step 3: correct)

Then, $\frac{a}{|a|} = -1, \frac{b}{|b|} = 1, \frac{c}{|c|} = 1, \frac{d}{|d|} = 1,$ and $\frac{abcd}{|abcd|} = -1.$ (Step 4: correct)

The expression becomes $-1 + 1 + 1 + 1 - 1 = 0.$ (Step 5: wrong, $-1 + 1 + 1 + 1 - 1 = 1$)

3. Two of $a, b, c,$ or d are negative, and the other two are positive:

There are two sub-cases to consider:

Case 3.1: a and b are negative, and c and d are positive.

Case 3.2: a and c are negative, and b and d are positive.

For Case 3.1:

$\frac{a}{|a|} = -1, \frac{b}{|b|} = -1, \frac{c}{|c|} = 1, \frac{d}{|d|} = 1,$ and $\frac{abcd}{|abcd|} = 1.$ (Step 6: correct)

The expression evaluates to $-1 - 1 + 1 + 1 = 0.$ (Step 7: correct)

For Case 3.2:

$\frac{a}{|a|} = -1, \frac{b}{|b|} = 1, \frac{c}{|c|} = -1, \frac{d}{|d|} = 1,$ and $\frac{abcd}{|abcd|} = -1.$ (Step 8: wrong, when a and c are negative, b and d are positive, $\frac{abcd}{|abcd|} = 1$)

The expression becomes $-1 + 1 - 1 + 1 = 0.$ (Step 9: wrong, $-1 + 1 - 1 + 1 = -1$)

4. Three of $a, b, c,$ or d are negative, and one is positive:

Similarly, there are three sub-cases:

Case 4.1: $a, b,$ and c are negative, and d is positive.

Case 4.2: $a, b,$ and d are negative, and c is positive.

Case 4.3: $a, c,$ and d are negative, and b is positive.

For Case 4.1:

$\frac{a}{|a|} = -1, \frac{b}{|b|} = -1, \frac{c}{|c|} = -1, \frac{d}{|d|} = 1,$ and $\frac{abcd}{|abcd|} = -1.$ (Step 10: correct)

The expression gives $-1 - 1 - 1 + 1 = -2.$ (Step 11: correct)

For Case 4.2:

$\frac{a}{|a|} = -1, \frac{b}{|b|} = -1, \frac{c}{|c|} = 1, \frac{d}{|d|} = -1,$ and $\frac{abcd}{|abcd|} = 1.$ (Step 12: wrong, when $a, b,$ and d are negative, and c is positive, $\frac{abcd}{|abcd|} = -1$)

The result is $-1 - 1 + 1 - 1 = -2.$ (Step 13: wrong, $-1 - 1 + 1 - 1 = -1$)

For Case 4.3:

$\frac{a}{|a|} = -1, \frac{b}{|b|} = 1, \frac{c}{|c|} = -1, \frac{d}{|d|} = -1,$ and $\frac{abcd}{|abcd|} = 1.$ (Step 14: wrong, when $a, c,$ and d are negative, and b is positive, $\frac{abcd}{|abcd|} = -1$)

The expression evaluates to $-1 + 1 - 1 - 1 = -1.$ (Step 15: correct)

5. All of $a, b, c,$ and d are negative:

In this situation, $\frac{a}{|a|} = -1, \frac{b}{|b|} = -1, \frac{c}{|c|} = -1, \frac{d}{|d|} = -1,$ and $\frac{abcd}{|abcd|} = 1.$ (Step 16: correct)

Thus, the expression evaluates to $-1 - 1 - 1 - 1 + 1 = -3.$ (Step 17: correct)

Final result: $[-3, -2, -1, 0, 1, 5]$

Comment:

There are 17 steps totally while 6 of them are wrong, step-wise error ratio is **35.3%**.

Table 29: Example 2 for step-wise error ratio - “Absolute value of Numbers”: Supervision sampled by **Command-R-2024-08**.

Example 2 for step-wise error ratio - "Absolute value of Numbers": Supervision sampled by **GPT-3.5-turbo**

Task:

Given that a , b , c , and d are nonzero real numbers, find all possible values of the expression $\frac{a}{|a|} + \frac{b}{|b|} + \frac{c}{|c|} + \frac{d}{|d|} + \frac{abcd}{|abcd|}$.

Solution:

For any non-zero real number x , we have $|x| = x$ if $x > 0$ and $|x| = -x$ if $x < 0$. (Step 1: correct)

Therefore, we can simplify the given expression as follows:

1. $\frac{a}{|a|} = 1$ if $a > 0$, and $\frac{a}{|a|} = -1$ if $a < 0$. (Step 2: correct)
2. $\frac{b}{|b|} = 1$ if $b > 0$, and $\frac{b}{|b|} = -1$ if $b < 0$. (Step 3: correct)
3. $\frac{c}{|c|} = 1$ if $c > 0$, and $\frac{c}{|c|} = -1$ if $c < 0$. (Step 4: correct)
4. $\frac{d}{|d|} = 1$ if $d > 0$, and $\frac{d}{|d|} = -1$ if $d < 0$. (Step 5: correct)
5. $\frac{abcd}{|abcd|} = 1$ if $abcd > 0$, and $\frac{abcd}{|abcd|} = -1$ if $abcd < 0$. (Step 6: correct)

Therefore, the expression simplifies to the sum of 5 terms, each 1 or -1 based on the sign of a , b , c , d , and $abcd$.

After analyzing the sign possibilities, we can see that there are $2^5 = 32$ different combinations of signs that these terms can take. However, not all of these combinations are possible since a , b , c , and d are non-zero real numbers. (Step 7: correct)

Let's consider the possible cases:

1. If all a , b , c , and d are positive, then the expression evaluates to $1 + 1 + 1 + 1 + 1 = 5$. (Step 8: correct)
2. If one of a , b , c , and d is negative, then the expression evaluates to $-1 + 1 + 1 + 1 - 1 = 1$. (Step 9: correct)
3. If two of a , b , c , and d are negative, then the expression evaluates to $-1 - 1 + 1 + 1 + 1 = 1$. (Step 10: correct)
4. If three of a , b , c , and d are negative, then the expression evaluates to $-1 - 1 - 1 + 1 + 1 = -1$. (Step 11: wrong, if three of a , b , c , d are negative, the expression is $-1 - 1 - 1 + 1 + 1 = -3$.)
5. If all a , b , c , and d are negative, then the expression evaluates to $-1 - 1 - 1 - 1 - 1 = -5$. (Step 12: wrong, if all of a , b , c , d are negative, the expression is $-1 - 1 - 1 - 1 + 1 = -3$.)

Therefore, the possible values for the expression are $\boxed{5, 1, -1, -5}$.

Comment:

There are 12 steps totally while 2 of them are wrong, step-wise error ratio is **16.7%**.

Table 30: Example 2 for step-wise error ratio - "Absolute value of Numbers": Supervision sampled by **GPT-3.5-turbo**.

Example 2 for step-wise error ratio - “Absolute value of Numbers”: Supervision sampled by **Llama-3-70B-Instruct**

Task:

Given that a, b, c , and d are nonzero real numbers, find all possible values of the expression $\frac{a}{|a|} + \frac{b}{|b|} + \frac{c}{|c|} + \frac{d}{|d|} + \frac{abcd}{|abcd|}$.

Solution:

To solve this problem, we need to consider the possible signs of a, b, c , and d .

If all of a, b, c , and d are positive, then $|a| = a, |b| = b, |c| = c$, and $|d| = d$, (Step 1: correct)

so the expression becomes $\frac{a}{a} + \frac{b}{b} + \frac{c}{c} + \frac{d}{d} + \frac{abcd}{abcd} = 1 + 1 + 1 + 1 + 1 = 5$. (Step 2: correct)

If all of a, b, c , and d are negative, then $|a| = -a, |b| = -b, |c| = -c$, and $|d| = -d$, (Step 3: correct)

so the expression becomes $\frac{a}{-a} + \frac{b}{-b} + \frac{c}{-c} + \frac{d}{-d} + \frac{abcd}{abcd} = -1 - 1 - 1 - 1 + 1 = -3$. (Step 4: correct)

If three of a, b, c , and d are positive and one is negative, then we have either

$\frac{a}{a} + \frac{b}{b} + \frac{c}{c} - \frac{d}{d} + \frac{abcd}{-abcd} = 1 + 1 + 1 - 1 - 1 = 0$, (Step 5: wrong, $1 + 1 + 1 - 1 = 1$)

or

$\frac{a}{a} + \frac{b}{b} - \frac{c}{c} + \frac{d}{d} + \frac{abcd}{-abcd} = 1 + 1 - 1 + 1 - 1 = 0$, (Step 6: wrong, $1 + 1 - 1 + 1 = 1$)

or

$\frac{a}{a} - \frac{b}{b} + \frac{c}{c} + \frac{d}{d} + \frac{abcd}{-abcd} = 1 - 1 + 1 + 1 - 1 = 0$, (Step 7: wrong, $1 - 1 + 1 + 1 = 1$)

or

$-\frac{a}{a} + \frac{b}{b} + \frac{c}{c} + \frac{d}{d} + \frac{abcd}{-abcd} = -1 + 1 + 1 + 1 - 1 = 0$. (Step 8: wrong, $1 - 1 + 1 + 1 = 1$)

If three of a, b, c , and d are negative and one is positive, then we have either

$-\frac{a}{a} - \frac{b}{b} - \frac{c}{c} + \frac{d}{d} + \frac{abcd}{abcd} = -1 - 1 - 1 + 1 + 1 = 0$, (Step 9: wrong, $1 - 1 + 1 + 1 = -1$)

or

$-\frac{a}{a} - \frac{b}{b} + \frac{c}{c} - \frac{d}{d} + \frac{abcd}{abcd} = -1 - 1 + 1 - 1 + 1 = 0$, (Step 10: wrong, $-1 - 1 + 1 - 1 = -1$)

or

$-\frac{a}{a} + \frac{b}{b} - \frac{c}{c} - \frac{d}{d} + \frac{abcd}{abcd} = -1 + 1 - 1 - 1 + 1 = 0$, (Step 11: wrong, $-1 + 1 - 1 - 1 = -1$)

or

$\frac{a}{a} - \frac{b}{b} - \frac{c}{c} - \frac{d}{d} + \frac{abcd}{abcd} = 1 - 1 - 1 - 1 + 1 = 0$. (Step 12: wrong, $-1 + 1 - 1 - 1 = -1$)

Therefore, the possible values of the expression are $\boxed{5, -3, 0}$.

Comment:

There are 12 steps totally while 8 of them are wrong, step-wise error ratio is **66.7%**.

Table 31: Example 2 for step-wise error ratio - “Absolute value of Numbers”: Supervision sampled by **Llama-3-70B-Instruct**.

Example 2 for step-wise error ratio - “Absolute value of Numbers”: Supervision sampled by **GPT-4o-mini**

Task:

Given that a , b , c , and d are nonzero real numbers, find all possible values of the expression $\frac{a}{|a|} + \frac{b}{|b|} + \frac{c}{|c|} + \frac{d}{|d|} + \frac{abcd}{|abcd|}$.

Solution:

To solve the problem, we start by analyzing the individual terms in the expression:

$$\frac{a}{|a|}, \frac{b}{|b|}, \frac{c}{|c|}, \frac{d}{|d|}, \frac{abcd}{|abcd|}.$$

Each of these fractions represents the sign of the respective variable:

- $\frac{a}{|a|} = 1$ if $a > 0$ and -1 if $a < 0$. (Step 1: correct)
- $\frac{b}{|b|} = 1$ if $b > 0$ and -1 if $b < 0$. (Step 2: correct)
- $\frac{c}{|c|} = 1$ if $c > 0$ and -1 if $c < 0$. (Step 3: correct)
- $\frac{d}{|d|} = 1$ if $d > 0$ and -1 if $d < 0$. (Step 4: correct)
- $\frac{abcd}{|abcd|} = 1$ if $abcd > 0$ and -1 if $abcd < 0$ (Step 5: correct)

Now, let's denote the number of positive values among a , b , c , and d as p and the number of negative values as n . Since a , b , c , and d are nonzero, we have:
 $p + n = 4$. (Step 6: correct)

The term $\frac{abcd}{|abcd|}$ evaluates depending on the evenness or oddness of the product $abcd$. Specifically:

- If p (positive numbers) is even, $abcd > 0$, hence $\frac{abcd}{|abcd|} = 1$. (Step 7: correct)
- If p is odd, $abcd < 0$, so $\frac{abcd}{|abcd|} = -1$. (Step 8: correct)

Next, we can rewrite our entire expression:

$$\frac{a}{|a|} + \frac{b}{|b|} + \frac{c}{|c|} + \frac{d}{|d|} + \frac{abcd}{|abcd|} = p - n + \frac{abcd}{|abcd|}. \text{ (Step 9: correct)}$$

Since $n = 4 - p$, we can substitute:

$$p - (4 - p) + \frac{abcd}{|abcd|} = 2p - 4 + \frac{abcd}{|abcd|}. \text{ (Step 10: correct)}$$

From the previous discussion:

- If p is even, $\frac{abcd}{|abcd|} = 1$, giving:
 $2p - 4 + 1 = 2p - 3$. (Step 11: correct)
- If p is odd, $\frac{abcd}{|abcd|} = -1$ (Step 12: correct), yielding:
 $2p - 4 - 1 = 2p - 5$.

Now let's evaluate the possible values of p :

- For $p = 0$:
- Both cases result in $2(0) - 3 = -3$. (Step 13: correct)
- For $p = 1$:
- Both cases yield $2(1) - 3 = -1$. (Step 14: wrong, here p is odd, the expression should be $2p - 5 = 2(1) - 5 = -3$.)
- For $p = 2$:
- Results are $2(2) - 3 = 1$. (Step 15: correct)
- For $p = 3$:
- Yields $2(3) - 3 = 3$. (Step 16: wrong, here p is odd, the expression should be $2p - 5 = 2(3) - 5 = 1$.)
- For $p = 4$:
- Gives $2(4) - 3 = 5$. (Step 17: correct)

Now, collecting the unique values derived from all possible p :

- Values from $p = 0$: -3
- Values from $p = 1$: -1
- Values from $p = 2$: 1
- Values from $p = 3$: 3
- Values from $p = 4$: 5

Combining all these unique values, we have:

$$\{-3, -1, 1, 3, 5\}. \text{ (Step 18: correct)}$$

Thus, the possible values of the expression are:

$$\boxed{-3, -1, 1, 3, 5}.$$

Comment:

There are 18 steps totally while 2 of them are wrong, step-wise error ratio is **11.1%**.

Table 32: Example 2 for step-wise error ratio - “Absolute value of Numbers”: Supervision sampled by **GPT-4o-mini**.

Example 3 for step-wise error ratio - “Absolute value of Numbers”: Supervision sampled by **Command-R-2024-03**

Task:

Given point G is the centroid of $\triangle ABC$, with the sides opposite to angles A, B, C having lengths a, b, c respectively, and it is known that $\frac{a}{5}\vec{GA} + \frac{b}{7}\vec{GB} + \frac{c}{8}\vec{GC} = \vec{0}$, then the measure of angle B is ?.

Solution:

From the given information, we have:

$$\begin{aligned} \frac{a}{5}\vec{GA} + \frac{b}{7}\vec{GB} + \frac{c}{8}\vec{GC} &= \vec{0} \\ \implies \frac{a}{5} \cdot \frac{2\vec{HA}}{\sin A} + \frac{b}{7} \cdot \frac{2\vec{HB}}{\sin B} + \frac{c}{8} \cdot \frac{2\vec{HC}}{\sin C} &= \vec{0} \text{ (Step 1: wrong, } \vec{HA}, \vec{HB}, \vec{HC} \text{ are not defined.)} \\ \implies \frac{2a}{5\sin A} \cdot \vec{HA} + \frac{2b}{7\sin B} \cdot \vec{HB} + \frac{2c}{8\sin C} \cdot \vec{HC} &= \vec{0} \text{ (Step 2: correct)} \end{aligned}$$

Since the three vectors \vec{HA}, \vec{HB} and \vec{HC} are coplanar and sum to $\vec{0}$, the only way for their scaled versions above to also sum to $\vec{0}$, is for each of them to be zero individually. (Step 3: correct)

This means that: $\frac{2a}{5\sin A} = 0 \implies \boxed{\sin A = \frac{5a}{2}}$ (Step 4: wrong, $\sin A = \frac{5a}{2}$ can not be inferred from $\frac{2a}{5\sin A} = 0$.)

This also implies $\sin B = \frac{5b}{2}, \sin C = \frac{5c}{2}$. (Step 5: wrong, the reason is same as above step.)

Using the fact that $\sin A + \sin B + \sin C = 1$, we get: (Step 6: wrong, $\sin A + \sin B + \sin C = 1$ does not always hold for general cases.)

$$\frac{5a}{2} + \frac{5b}{2} + \frac{5c}{2} = 1 \implies \frac{5}{2}(a + b + c) = 1 \text{ (Step 7: correct)}$$

Since $a + b + c$ is just twice the area of the triangle, we get that the area of $\triangle ABC$ is $\frac{1}{10}$. (Step 8: wrong, it's not mentioned in the problem or we cannot infer that $a + b + c$ is twice the area of the triangle)

In particular, this means that the angle B , which measures $\frac{2\pi}{10}$, or $\boxed{12^\circ}$, is the smallest angle of the three. (Step 9: wrong, we cannot infer from the above steps that angle B is $\frac{2\pi}{10}$. Also, $\frac{2\pi}{10} \neq 12^\circ$)

Comment:

There are 9 steps totally while 6 of them are wrong, step-wise error ratio is **66.7%**.

Table 33: Example 3 for step-wise error ratio - “Absolute value of Numbers”: Supervision sampled by **Command-R-2024-03**.

Example 3 for step-wise error ratio - “Absolute value of Numbers”: Supervision sampled by **Command-R-2024-08**

Task:

Given point G is the centroid of $\triangle ABC$, with the sides opposite to angles A, B, C having lengths a, b, c respectively, and it is known that $\frac{a}{5}\vec{GA} + \frac{b}{7}\vec{GB} + \frac{c}{8}\vec{GC} = \vec{0}$, then the measure of angle B is ?.

Solution:

Let's think step by step to solve this problem involving the centroid of a triangle.

Given that point G is the centroid of $\triangle ABC$, we know that G is the center of mass of the triangle, and it divides the medians (lines connecting a vertex to the midpoint of the opposite side) in a 2 : 1 ratio.

From the equation:

$$\frac{a}{5}\vec{GA} + \frac{b}{7}\vec{GB} + \frac{c}{8}\vec{GC} = \vec{0}$$

We can see that the vector sum of the scaled medians from the centroid to the sides of the triangle is equal to the zero vector. This implies that the centroid G is equidistant from the sides of the triangle, which is a property of the centroid.

Now, let's focus on angle B . Since G is the centroid, it divides the median from B to the midpoint of side AC in a 2 : 1 ratio. This means that $GB : GC = 2 : 1$. (Step 1: wrong, $GB : GC \neq 2 : 1$ if G is the centroid and B, C are vertices of triangle.)

From the equation, we have:

$$\frac{b}{7}\vec{GB} = -\left(\frac{a}{5}\vec{GA} + \frac{c}{8}\vec{GC}\right) \text{ (Step 2: correct)}$$

Taking the magnitude of both sides, we get:

$$\frac{b}{7} \cdot |\vec{GB}| = \frac{a}{5} \cdot |\vec{GA}| + \frac{c}{8} \cdot |\vec{GC}| \text{ (Step 3: wrong, } |-\left(\frac{a}{5}\vec{GA} + \frac{c}{8}\vec{GC}\right)| \neq \frac{a}{5} \cdot |\vec{GA}| + \frac{c}{8} \cdot |\vec{GC}| \text{)}$$

Since $GB : GC = 2 : 1$, we can write:

$$\frac{b}{7} \cdot 2 \cdot |\vec{GC}| = \frac{a}{5} \cdot |\vec{GA}| + \frac{c}{8} \cdot |\vec{GC}| \text{ (Step 4: correct)}$$

Simplifying the equation:

$$\frac{2b}{7} \cdot |\vec{GC}| = \frac{a}{5} \cdot |\vec{GA}| + \frac{c}{8} \cdot |\vec{GC}| \text{ (Step 5: correct)}$$

Now, we can solve for $|\vec{GC}|$:

$$\frac{2b}{7} \cdot |\vec{GC}| - \frac{c}{8} \cdot |\vec{GC}| = \frac{a}{5} \cdot |\vec{GA}| \text{ (Step 6: correct)}$$

$$\left(\frac{2b}{7} - \frac{c}{8}\right) \cdot |\vec{GC}| = \frac{a}{5} \cdot |\vec{GA}| \text{ (Step 7: correct)}$$

Dividing both sides by $\left(\frac{2b}{7} - \frac{c}{8}\right)$:

$$|\vec{GC}| = \frac{\frac{a}{5} \cdot |\vec{GA}|}{\frac{2b}{7} - \frac{c}{8}} \text{ (Step 8: correct)}$$

Now, we can use the fact that $GB : GC = 2 : 1$ to find $|\vec{GB}|$:

$$|\vec{GB}| = 2 \cdot |\vec{GC}| = 2 \cdot \frac{\frac{a}{5} \cdot |\vec{GA}|}{\frac{2b}{7} - \frac{c}{8}} \text{ (Step 9: correct)}$$

Now, we have the magnitudes of the vectors \vec{GB} and \vec{GC} . To find the measure of angle B , we can use the cosine rule, which states:

$$c^2 = a^2 + b^2 - 2ab \cos C \text{ (Step 10: correct)}$$

In our case, we have:

$$|\vec{GC}|^2 = |\vec{GB}|^2 + |\vec{GA}|^2 - 2|\vec{GB}||\vec{GA}| \cos B \text{ (Step 11: correct)}$$

Plugging in the values:

$$\left(\frac{\frac{a}{5} \cdot |\vec{GA}|}{\frac{2b}{7} - \frac{c}{8}}\right)^2 = |\vec{GB}|^2 \text{ (Step 12: wrong, with the equation from Step 9, it should be } \left(\frac{\frac{a}{5} \cdot |\vec{GA}|}{\frac{2b}{7} - \frac{c}{8}}\right)^2 = \frac{1}{4}|\vec{GB}|^2 \text{)}$$

Comment:

There are 12 steps totally while 3 of them are wrong, step-wise error ratio is **25.0%**, but this solution is incomplete.

Table 34: Example 3 for step-wise error ratio - “Absolute value of Numbers”: Supervision sampled by **Command-R-2024-08**.

Example 3 for step-wise error ratio - "Absolute value of Numbers": Supervision sampled by **GPT-3.5-turbo**

Task:

Given point G is the centroid of $\triangle ABC$, with the sides opposite to angles A, B, C having lengths a, b, c respectively, and it is known that $\frac{a}{5}\vec{GA} + \frac{b}{7}\vec{GB} + \frac{c}{8}\vec{GC} = \vec{0}$, then the measure of angle B is ?.

Solution:

Since point G is the centroid of $\triangle ABC$, we know that $\vec{GA} + \vec{GB} + \vec{GC} = \vec{0}$. (Step 1: correct)

Given that $\frac{a}{5}\vec{GA} + \frac{b}{7}\vec{GB} + \frac{c}{8}\vec{GC} = \vec{0}$, we can rewrite it as:

$$\frac{a}{5}\vec{GA} + \frac{b}{7}\vec{GB} + \frac{c}{8}\vec{GC} = \frac{a}{5}(\vec{GB} + \vec{GC}) + \frac{b}{7}(\vec{GA} + \vec{GC}) + \frac{c}{8}(\vec{GA} + \vec{GB}) \quad (\text{Step 2: correct})$$

This simplifies to:

$$\frac{a}{5}\vec{BC} + \frac{b}{7}\vec{AC} + \frac{c}{8}\vec{AB} = \vec{0} \quad (\text{Step 3: wrong, } \frac{a}{5}(\vec{GB} + \vec{GC}) + \frac{b}{7}(\vec{GA} + \vec{GC}) + \frac{c}{8}(\vec{GA} + \vec{GB}) = 0 \text{ does not imply } \frac{a}{5}\vec{BC} + \frac{b}{7}\vec{AC} + \frac{c}{8}\vec{AB} = \vec{0}, \text{ in other words, } \vec{GB} + \vec{GC} \neq \vec{BC}, \text{ so do the other two.})$$

Now, applying the vector sum formula in terms of sides of a triangle, we get:

$$a\vec{BC} + b\vec{AC} + c\vec{AB} = \vec{0} \quad (\text{Step 4: wrong, we cannot obtain this equation from } \frac{a}{5}\vec{BC} + \frac{b}{7}\vec{AC} + \frac{c}{8}\vec{AB} = \vec{0} \text{ in Step 3.})$$

This indicates that $\triangle ABC$ is a degenerate triangle, which means the points A, B, C are collinear. (Step 5: wrong, this conclusion is not reasonable.)

Therefore, the angle B must be $\boxed{180^\circ}$ (straight angle) for the sides to form a degenerate triangle. (Step 6: correct)

Comment:

There are 6 steps totally while 3 of them are wrong, step-wise error ratio is **50.0%**.

Table 35: Example 3 for step-wise error ratio - "Absolute value of Numbers": Supervision sampled by **GPT-3.5-turbo**.

Example 3 for step-wise error ratio - "Absolute value of Numbers": Supervision sampled by **Llama-3-70B-Instruct**

Task:

Given point G is the centroid of $\triangle ABC$, with the sides opposite to angles A, B, C having lengths a, b, c respectively, and it is known that $\frac{a}{5}\vec{GA} + \frac{b}{7}\vec{GB} + \frac{c}{8}\vec{GC} = \vec{0}$, then the measure of angle B is ?.

Solution:

Since G is the centroid of $\triangle ABC$, we know that $\vec{GA} + \vec{GB} + \vec{GC} = \vec{0}$. (Step 1: correct)

Now, we are given that $\frac{a}{5}\vec{GA} + \frac{b}{7}\vec{GB} + \frac{c}{8}\vec{GC} = \vec{0}$.

We can multiply the first equation by a constant k such that:

$$k\vec{GA} + k\vec{GB} + k\vec{GC} = \vec{0} \quad (\text{Step 2: correct})$$

Comparing this with the given equation, we can see that:

$$k = \frac{a}{5} = \frac{b}{7} = \frac{c}{8} \quad (\text{Step 3: correct})$$

Solving for a, b, c , we get:

$$a = 5k, b = 7k, c = 8k \quad (\text{Step 4: correct})$$

Now, using the law of cosines on $\triangle ABC$, we get:

$$c^2 = a^2 + b^2 - 2ab \cos(B) \quad (\text{Step 5: wrong, the implementation of cosine law is wrong, it should be } b^2 = a^2 + c^2 - 2ac \cos(B))$$

Substituting the values of a, b, c , we get:

$$(8k)^2 = (5k)^2 + (7k)^2 - 2(5k)(7k) \cos(B) \quad (\text{Step 6: correct})$$

Simplifying, we get:

$$64k^2 = 25k^2 + 49k^2 - 70k^2 \cos(B) \quad (\text{Step 7: correct})$$

$$64 = 74 - 70 \cos(B) \quad (\text{Step 8: correct})$$

$$70 \cos(B) = 10 \quad (\text{Step 9: correct})$$

$$\cos(B) = \frac{1}{7} \quad (\text{Step 10: correct})$$

$$B = \cos^{-1}\left(\frac{1}{7}\right) \quad (\text{Step 11: correct})$$

$$\boxed{B = \cos^{-1}\left(\frac{1}{7}\right)}$$

Comment:

There are 11 steps totally while 1 of them are wrong, step-wise error ratio is **9.1%**.

Table 36: Example 3 for step-wise error ratio - "Absolute value of Numbers": Supervision sampled by **Llama-3-70B-Instruct**.

Example 3 for step-wise error ratio - "Absolute value of Numbers": Supervision sampled by **GPT-4o-mini**

Task:

Given point G is the centroid of $\triangle ABC$, with the sides opposite to angles A, B, C having lengths a, b, c respectively, and it is known that $\frac{a}{5}\vec{GA} + \frac{b}{7}\vec{GB} + \frac{c}{8}\vec{GC} = \vec{0}$, then the measure of angle B is ?.

Solution:

Let's denote the position vectors of points A, B , and C as \vec{A}, \vec{B} , and \vec{C} respectively. The position vector of the centroid G of triangle ABC is given by:
 $\vec{G} = \frac{1}{3}(\vec{A} + \vec{B} + \vec{C})$ (Step 1: correct)

We can express \vec{GA}, \vec{GB} , and \vec{GC} as follows:

$$\vec{GA} = \vec{A} - \vec{G} = \vec{A} - \frac{1}{3}(\vec{A} + \vec{B} + \vec{C}) = \frac{2}{3}\vec{A} - \frac{1}{3}\vec{B} - \frac{1}{3}\vec{C} \text{ (Step 2: correct)}$$

$$\vec{GB} = \vec{B} - \vec{G} = \frac{2}{3}\vec{B} - \frac{1}{3}\vec{A} - \frac{1}{3}\vec{C} \text{ (Step 3: correct)}$$

$$\vec{GC} = \vec{C} - \vec{G} = \frac{2}{3}\vec{C} - \frac{1}{3}\vec{A} - \frac{1}{3}\vec{B} \text{ (Step 4: correct)}$$

Substituting these into the given equation:

$$\frac{a}{5}\vec{GA} + \frac{b}{7}\vec{GB} + \frac{c}{8}\vec{GC} = \vec{0} \text{ (Step 5: correct)}$$

This becomes:

$$\frac{a}{5} \left(\frac{2}{3}\vec{A} - \frac{1}{3}\vec{B} - \frac{1}{3}\vec{C} \right) + \frac{b}{7} \left(\frac{2}{3}\vec{B} - \frac{1}{3}\vec{A} - \frac{1}{3}\vec{C} \right) + \frac{c}{8} \left(\frac{2}{3}\vec{C} - \frac{1}{3}\vec{A} - \frac{1}{3}\vec{B} \right) = \vec{0} \text{ (Step 6: correct)}$$

Combining the terms gives:

$$\frac{2a}{15}\vec{A} - \frac{a}{15}\vec{B} - \frac{a}{15}\vec{C} + \frac{2b}{21}\vec{B} - \frac{b}{21}\vec{A} - \frac{b}{21}\vec{C} + \frac{2c}{24}\vec{C} - \frac{c}{24}\vec{A} - \frac{c}{24}\vec{B} = \vec{0} \text{ (Step 7: correct)}$$

Collecting the coefficients of \vec{A}, \vec{B} , and \vec{C} :

$$\left(\frac{2a}{15} - \frac{b}{21} - \frac{c}{24} \right) \vec{A} + \left(-\frac{a}{15} + \frac{2b}{21} - \frac{c}{24} \right) \vec{B} + \left(-\frac{a}{15} - \frac{b}{21} + \frac{2c}{24} \right) \vec{C} = \vec{0} \text{ (Step 8: correct)}$$

For this equation to hold, each coefficient must equal zero:

$$\frac{2a}{15} - \frac{b}{21} - \frac{c}{24} = 0, -\frac{a}{15} + \frac{2b}{21} - \frac{c}{24} = 0, -\frac{a}{15} - \frac{b}{21} + \frac{2c}{24} = 0$$

(Step 9: wrong, since \vec{A}, \vec{B} and \vec{C} are all 2-D vectors, the expression $\left(\frac{2a}{15} - \frac{b}{21} - \frac{c}{24} \right) \vec{A} + \left(-\frac{a}{15} + \frac{2b}{21} - \frac{c}{24} \right) \vec{B} + \left(-\frac{a}{15} - \frac{b}{21} + \frac{2c}{24} \right) \vec{C} = \vec{0}$ does not imply all the coefficients are zeros, there exist non-zero solutions for those coefficients.)

After solving, we can use the Law of Cosines to find the angle B .

Finding Angle B :

To find B using the cosine rule:

$$\cos B = \frac{a^2 + c^2 - b^2}{2ac} \text{ (Step 10: correct)}$$

From the relationships derived from the above equations, substituting a, b , and c will yield the angle B .

After computation, we find:

$$B = 90^\circ \text{ (Step 11: correct)}$$

Thus, the measure of angle B is:

$$\boxed{90^\circ}$$

Comment:

There are 18 steps totally while 2 of them are wrong, step-wise error ratio is **9.1%**.

Table 37: Example 3 for step-wise error ratio - "Absolute value of Numbers": Supervision sampled by **GPT-4o-mini**.

1. Definition

We show a math or scientific problem P and its solution S. Your task is:

- (1) break down the complete solution S properly into multiple steps,
- (2) select part of these steps properly (NEITHER ONE STEP NOR NEARLY ALL THE STEPS),
- (3) imagine $\{\text{num_subtasks}\}$ new problems based on your selected partial steps.

These problems can be seen as “simplified problems” or “partial problems” for the original one and are more simple to solve.

The new problems are denoted as P-1, P-2, ..., the corresponding solutions are denoted as S-1, S-2, ...

2. Guidelines

- (1) Completeness: Each new problem P-1, ..., MUST be an independent, self-contained and well-defined problem, with clean definitions and all necessary conditions to solve it.

For example problem P-1 CANNOT quote conditions or results directly from original problem P or original solution S, because P-1 is an independent problem from P and S. P-1 is also NOT allowed to quote conditions or results directly from other new problems and new solutions.

Specifically, the following words are NOT allowed to exist in your output: ‘original problem’, ‘original solution’, ‘previous problem’, ‘previous solution’, ‘all the new problems’, because their existence definitely imply this new problem or solution quote results from other problems/solutions and are not independent.

- (2) Difficulty and Complexity level: Even the new problems are seen as “sub-problems” of the original problem, but they SHOULD NOT be too simple like computing “ $\sqrt{64} + 2 = 10$ ”, “ $\sin(\pi/6) = 0.5$ ”. On the other hand, each new problem SHOULD NOT be too similar or even same as the whole original problem; the solution of new problem SHOULD only be a part of the original full one.

- (3) The solutions S-1, ..., MUST be self-contained, detailed and complete, providing thorough reasoning and calculations. Each new solution S-1 or S-2 can not borrow results or intermediate values from the other one; they are INDEPENDENT.

- (4) The solutions S-1, S-2, MUST be derived entirely from the original solution S. DO NOT introduce any new methods to S-1 and S-2 that were not mentioned in S. Do not quote results from original solution S or other new solutions.

- (5) It’s acceptable for the solutions S-1, S-2, to overlap. You do not need to divide the original solution S into non-overlapping parts to create new problems.

3. I/O format:

- (1) input format:

The original problem P: ...

The original solution S: ...

- (2) output format:

Item 1:

New problem 1: ...

New solution 1: ...

...

End

- (3) other output details:

The solution can be either in continuous plain text without explicit sequence numbers leading the solving steps or be listed step by step explicitly, both is okay. Please always make the final results boxed.

4. User Input

Please show:

the $\{\text{num_subtasks}\}$ new problems and solutions P-1 & S-1, ...

Remind all the guidelines above again.

Here’s the input:

The original problem P: $\{\text{original_problem}\}$

The original solution S: $\{\text{original_solution}\}$

Table 38: Prompt template for decomposing hard tasks into several easy tasks.

<p>1. Filter out sub-tasks that are ill-defined or missing conditions.</p> <p>We are providing a math or scientific problem with its solution. Check if the solution use some conditions (like “n=4”, “f(x)=2x-1”, “Jenny has 6 apples in total”, “the water is poured in this way”, etc) that do not exist in the problem. If so, then the problem is considered not well-defined.</p> <p>Output format:</p> <p>(if problem is well-defind) Conclusion: 1.</p> <p>(if problem is not well-defind) Conclusion: 0. Reason: (brief reason for it’s not well-defined).</p> <p>Follow the output format strictly. Here’s the input:</p> <p>### Problem: \${problem}</p> <p>### Solution: \${solution}</p>
<p>2. Filter out sub-tasks that contains less than three steps, which are considered as overly simple task.</p> <p>Here we provide a detailed solution to a math or scientific problem, please count how many arithmetic computation steps or reasoning steps are there.</p> <p>Output format (**only output the following things**):</p> <p>Steps count: (integer).</p> <p>Here we provide the solution: \${solution}</p>
<p>3. Filter out sub-tasks that are variants of original task with only some numerical values or conditions changed.</p> <p>Here we provide two math or scientific problems A and B. If A is a variant of or almost the same as B, which means the only difference in A and B are values of variables and other things are almost all identical.</p> <p>The example of two almost same problems:</p> <p>Problem A: Given a set (S) of eleven distinct integers, where six of the integers are (5, 7, 8, 13, 16,), find the sum of smallest 3 elements.</p> <p>Problem B: Given a set (T) of eleven distinct integers, where six of the integers are (5, 7, 16,), find the sum of smallest 2 elements.</p> <p>We can see Problem A is just a variant of Problem B, they are very similar and solve for the same things. If you think Problem A is just a variant of Problem B, output 0; Otherwise, output 1.</p> <p>Output format (**only output the following things**):</p> <p>Conclusion: <int, 0 or 1>.</p> <p>Here is the input, we provide the problems A and B:</p> <p>### Problem A: \${problem1}</p> <p>### Problem B: \${problem2}</p>

Table 39: Prompt template for filtering out unqualified decomposed sub-tasks.

<p>Suppose you’re a student learning to solve math/scientific problems, here we provide a math/scientific problem, please solve it and show **DETAILED** process of computation and reasoning, but **DO NOT** be too naggy and lengthy. Remember to **BOX FINAL RESULT**.</p> <p>Input format: ### Problem: ...</p> <p>Output format: ### Solution: ...</p> <p>Here’s the problem:</p> <p>### Problem: \${problem}</p>

Table 40: Prompt template for sampling multiple solutions for hard and easy tasks.

Suppose you're a teacher grading students' homework of math or scientific problems. Here we provide a math problem, its correct solution S_{ref} and a solution to be judged for correctness, denoted as S . Please determine if S is correct based on the ****FINAL RESULT of S and S_{ref} ****.

Note in the following several exemptions, S should be considered correct:

- (1) If the final result of S looks different from S_{ref} ****only because they round to different digits**** during computation or for the final result, like 1.89, 1.891, 1.8912.
- (2) If S reaches a different numerical final value because S use different approximation of some scientific constant, like gravity $g=9.8$ or $g=10$.
- (3) If S uses a different method to solve the problem but it reaches the ****CORRECT FINAL RESULT****.

In other case, like S has some computation error, or reasoning error or method error, which leads to a wrong result, S should be considered as an incorrect solution. In one word, ****the "FINAL RESULT" is the main thing we use to judge the correctness****.

Input format:
 ### Problem:
 ### Reference Solution:
 ### Solution to be judged:
 Output format:
 ### Judgement: <int, 1 for correct, 0 for wrong, N/A if you can't judge it>

Here's the input:
 ### Problem: $\{problem\}$
 ### Reference Solution: $\{reference_solution\}$
 ### Solution to be judged: $\{student_solution\}$

Table 41: Prompt template for checking the correctness of sampled solutions based on the reference solution.

You're a model that helps generate a solution with the help of reference answer. You can refer to the provided information, pretend you haven't seen it, and generate a concise solution from scratch, with a little bit rephrasing.

Problem:
 $\{problem\}$
 ### Reference Info:
 $\{reference_solution\}$
 Let's think step by step from scratch! (Use the symbols you commonly utilize in the solution, such as Latex symbols. Don't follow the symbols in the reference info. Use less ****Conclusion:**** paragraph title in the end. Use less 'Let's' or '##' to start with in the first solution sentence. NOTE that the number of tokens of your solution ****SHOULD be in range of**** $\{int(sample_avg_token_num*0.8)\}$ to $\{int(sample_avg_token_num*1.2)\}$)

Table 42: Prompt template for transferring style of ground-truth to the style of sampled solution, when all the sampled solutions are incorrect in the construction of {correct/incorrect solution} pairs.

Suppose you're an expert in generating variants of math or scientific problems and their solutions. Here we provide a problem or a solution, please rephrase it and make sure ****DO NOT**** change the original meaning of the problem or the solution.

Output format:
 ### Rephrased version: $\{your\ output\}$.

Here's the input, which is either a math problem or a student's solution: $\{problem/solution\}$

Table 43: Prompt template for rephrasing the problem or the solution.