MR-Ben: A Meta-Reasoning Benchmark for Evaluating System-2 Thinking in LLMs

Zhongshen Zeng[1] **Yinhong Liu**[2] **Yingjia Wan**[2]
Jingyao Li[1] **Pengguang Chen**[1] **Jianbo Dai**[3] **Yuxuan Yao**[4]
Rongwu Xu[5] **Zehan Qi**[5] **Wanru Zhao**[2] **Linling Shen**[6]
Jianqiao Lu[7] **Haochen Tan**[4] **Yukang Chen**[1] **Hao Zhang**[8]
Zhan Shi[6] **Bailin Wang**[9] **Zhijiang Guo**[2†] **Jiaya Jia**[1†]

[1]Chinese University of Hong Kong [2]University of Cambridge [3]University of Edinburgh
[4]City University of Hong Kong [5]Tsinghua University [6]University of Texas at Austin
[7]University of Hong Kong [8]Nanyang Technological University
[9]Massachusetts Institute of Technology

Abstract

Large language models (LLMs) have shown increasing capability in problem-solving and decision-making, largely based on the step-by-step chain-of-thought reasoning processes. However, evaluating these reasoning abilities has become increasingly challenging. Existing *outcome-based* benchmarks are beginning to saturate, becoming less effective in tracking meaningful progress. To address this, we present a *process-based* benchmark MR-Ben that demands a meta-reasoning skill, where LMs are asked to locate and analyse potential errors in automatically generated reasoning steps. Our meta-reasoning paradigm is especially suited for system-2 slow thinking, mirroring the human cognitive process of carefully examining assumptions, conditions, calculations, and logic to identify mistakes. MR-Ben comprises 5,975 questions curated by human experts across a wide range of subjects, including physics, chemistry, logic, coding, and more. Through our designed metrics for assessing meta-reasoning on this benchmark, we identify interesting limitations and weaknesses of current LLMs (open-source and closed-source models). For example, with models like the o1 series from OpenAI demonstrating strong performance by effectively scrutinizing the solution space, many other state-of-the-art models fall significantly behind on MR-Ben, exposing potential shortcomings in their training strategies and inference methodologies[1].

1 Introduction

Reasoning, the cognitive process of using evidence, arguments, and logic to reach conclusions, is crucial for problem-solving, decision-making, and critical thinking [65, 19]. With the rapid advancement of Large Language Models (LLMs), there is an increasing interest in exploring their reasoning capabilities [30, 57]. Consequently, evaluating reasoning in LLMs reliably becomes paramount. Current evaluation methodologies primarily focus on the final result [16, 28, 22, 60], disregarding the intricacies of the reasoning process. While effective to some extent, such evaluation practices may conceal underlying issues like logical errors or unnecessary steps that compromise the accuracy and efficiency of reasoning [68, 41].

†Correspondence to: Zhijiang Guo (zg283@cam.ac.uk) and Jiaya Jia (leojia@cse.cuhk.edu.hk).
[1]Our dataset and codes are available on `https://randolph-zeng.github.io/Mr-Ben.github.io`.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Therefore, it is important to complement outcome-based evaluation with an intrinsic evaluation of the quality of the reasoning process. However, current benchmarks for evaluating LLMs' reasoning capabilities have certain limitations in terms of their scope and size. For instance, PRM800K [38] categorizes each reasoning step as positive, negative, or neutral. Similarly, BIG-Bench Mistake [64] focuses on identifying errors in step-level answers. We follow the same meta-reasoning paradigm as MR-GSM8K [77] and MR-Math [68], which go a step further by providing the error reason for the first negative step in the reasoning chain. However, these benchmarks are limited to a narrower task scope—MR-GSM8K and MR-Math focus solely on mathematical reasoning, while BIG-Bench Mistake mainly assesses logical reasoning. To ensure a comprehensive evaluation of reasoning abilities, it is crucial to identify reasoning errors and assess the LLMs' capacity to elucidate them across wider domains.

To bridge this gap, we construct a comprehensive benchmark MR-Ben comprising 6k questions covering a wide range of subjects, including natural sciences like math, biology, and physics, as well as coding and logic. One unique aspect of MR-Ben is its meta-reasoning paradigm, which involves challenging LLMs to reason about different forms of reasoning. In this paradigm, LLMs take on the role of a teacher, evaluating the reasoning process by assessing correctness, analyzing potential errors, and providing corrections, as depicted in Figure 1.

Our analysis of various LLMs [50, 51, 5, 33, 47] uncovers distinct limitations and previously unidentified weaknesses in their reasoning abilities. While many LLMs are capable of generating correct answers, they often struggle to identify errors within their reasoning processes and explain the underlying rationale. To excel under our meta-reasoning paradigm, models must meticulously scrutinize assumptions, conditions, calculations, and logical steps, even inferring step outcomes counterfactually. These requirements align with the characteristics of "System-2" slow thinking [35, 9], which we believe remains underdeveloped in most of the state-of-the-art models we evaluated.

We suspect that a key reason for this gap lies in current fine-tuning paradigms, which prioritize correct solutions and limit effective exploration of the broader solution space. Echoing this hypothesis, we observed that models like o1-preview [52], which reportedly incorporate effective search and disambiguation techniques across trajectories in the solution space, outperform other models by a large margin. Moreover, we found that leveraging high-quality and diverse synthetic data [1] significantly mitigates this issue, offering a promising path to enhance performance regardless of model size. Additionally, our results indicate that different LLMs excel in distinct reasoning paradigms, challenging the notion that domain-specific enhancements necessarily yield broad cognitive improvements. We hope that MR-Ben will guide researchers in comprehensively evaluating their models' capabilities and foster the development of more robust AI reasoning frameworks.

Our key contributions are summarized as follows:

- We introduced MR-Ben, which includes around 6k questions across a wide range of subjects, from natural sciences to coding and logic, and employs a unique meta-reasoning paradigm.
- We conduct an extensive analysis of various LLMs on MR-Ben, revealing various limitations and previously unidentified weaknesses in their reasoning abilities.
- We offer potential pathways for enhancing the reasoning abilities of LLMs and challenge the assumption that domain-specific enhancements necessarily lead to broad improvements.

2 Related Works

Reasoning Benchmarks Evaluating the reasoning capabilities of LLMs is crucial for understanding their potential and limitations. While existing benchmarks often assess reasoning by measuring performance on tasks that require reasoning, such as accuracy, they often focus on specific reasoning types like arithmetic, knowledge, logic, or algorithmic reasoning. Arithmetic reasoning, involving mathematical concepts and operations, has been explored in benchmarks ranging from elementary word problems [37, 4, 55, 16] to more complex and large-scale tasks [28, 48]. Knowledge reasoning, on the other hand, requires either internal (commonsense) or external knowledge, or a combination of both [14, 62, 22]. Logical reasoning benchmarks, encompassing deductive and inductive reasoning, use synthetic rule bases for the former [15, 61, 18] and specific observations for the latter to formulate general principles [78, 71]. Algorithmic reasoning often involves understanding the coding problem description and performing multi-step reasoning to solve it [17, 25]. Benchmarks like BBH [59] and

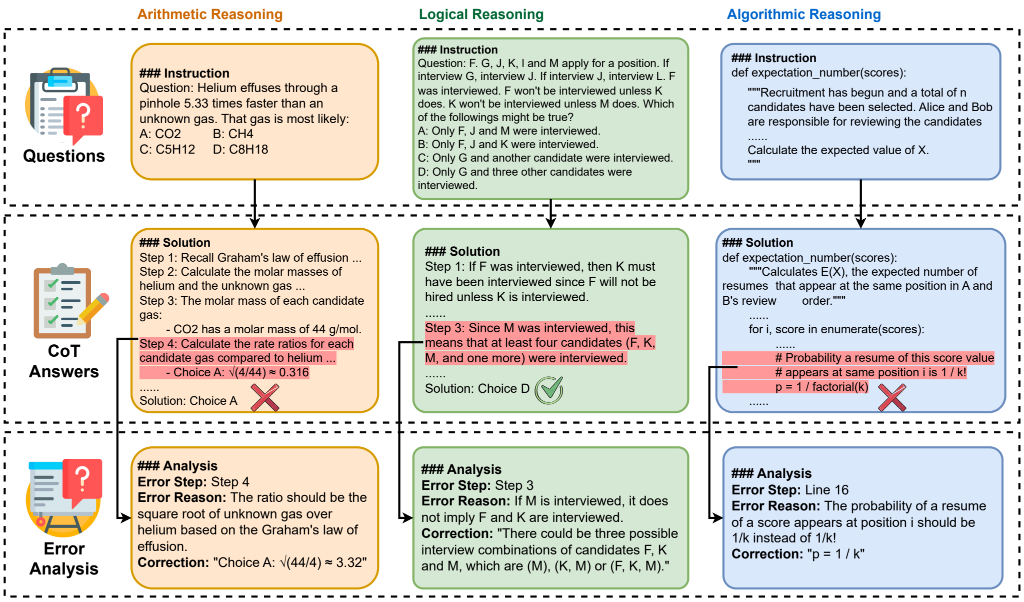

Figure 1: Overview of the evaluation paradigm and representative examples in MR-Ben. Each data point encompasses three key elements: a question, a Chain-of-Thought (CoT) answer, and an error analysis. The CoT answer is generated by various LLMs. Human experts annotate the error analyses, which include error steps, reasons behind the error, and subsequent corrections. The three examples shown are selected to represent arithmetic, logical, and algorithmic reasoning types.

MMLU [27] indirectly assess reasoning by evaluating performance on tasks that require it. However, these benchmarks primarily focus on final results, neglecting the analysis of potential errors in the reasoning process. Unlike prior efforts, MR-Ben goes beyond accuracy by assessing the ability to locate potential errors in the reasoning process and provide explanations and corrections. Moreover, MR-Ben covers different types of reasoning, offering a more comprehensive assessment.

Evaluation Beyond Accuracy Many recent studies have shifted their focus from using only the final result to evaluating the reasoning quality beyond accuracy. This shift has led to the development of two approaches: reference-free and reference-based evaluation. Reference-free methods aim to assess reasoning quality without relying on human-provided solutions. For example, ROSCOE [23] evaluates reasoning chains by quantifying reasoning errors such as redundancy and hallucination. Other approaches convert reasoning steps into structured forms, like subject-verb-object frames [56] or symbolic proofs [58], allowing for automated analysis. Reference-based methods depend on human-generated step-by-step solutions. For instance, PRM800K [38] offers solutions to MATH problems [28], categorizing each reasoning step as positive, negative, or neutral. Building on this, MR-GSM8K [77] and MR-Math [68] further provide the error reason behind the first negative step. MR-GSM8K focuses on elementary math problems, sampling questions from GSM8K [16]. MR-Math samples a smaller set of 459 questions from MATH [28]. Using the same annotation scheme, BIG-Bench Mistake [64] focuses on symbolic reasoning. It encompasses 2,186 instances from 5 tasks in BBH [59]. Despite the progress made by these datasets, limitations in scope and size remain. To address this, we introduce MR-Ben, a benchmark consisting of 5,975 manually annotated instances covering a wide range of subjects, including natural sciences, coding, and logic. MR-Ben also features more challenging questions, spanning high school, graduate, and professional levels.

3 MR-Ben: Dataset Construction

3.1 Dataset Structure

To comprehensively evaluate the reasoning capabilities of LLMs, MR-Ben employs a meta-reasoning paradigm. This paradigm casts LLMs in the role of a teacher, where they assess the reasoning process by evaluating its correctness, analyzing errors, and providing corrections. As shown in Figure 1, each data point within MR-Ben consists of three key elements: a question, a CoT answer, and an error analysis. The construction pipeline is shown in Figure 6 in Appendix-D.

Question The questions in MR-Ben are designed to cover a diverse range of reasoning types and difficulty levels, spanning from high school to professional levels. To ensure this breadth, we curated questions from various subjects, including natural sciences (mathematics, biology, physics), coding, and logic. Specifically, we sampled questions from mathematics, physics, biology, chemistry, and medicine from MMLU [27], which comprehensively assesses LLMs across academic and professional domains. For logic questions, we draw from LogiQA [40], which encompasses a broad spectrum of logical reasoning types, including categorical, conditional, disjunctive, and conjunctive reasoning. Finally, we select coding problems from MHPP [17], which focuses on function-level code generation requiring advanced algorithmic reasoning. Questions in MMLU and LogiQA require a single-choice answer, while MHPP requires a snippet of code as the answer.

CoT Answer We queried GPT-3.5-Turbo-0125 [50], Claude2 [5], and Mistral-Medium [32] (as of February 2024) using a prompt template (provided in Figure-7 in Appendix-D) designed to elicit step-by-step solutions [66]. For clarity, all LLMs were instructed to format their solutions with numbered steps, except for coding problems. To encourage diverse solutions, we set the temperature parameter to 1 during sampling. This empirical setting yielded satisfactory instruction following and desirable fine-grained reasoning errors, which annotators and evaluated models are expected to identify.

3.2 Annotation Process

After acquiring the questions and their corresponding Chain-of-Thought (CoT) answers, we engage annotators to provide error analyses. The annotation process is divided into three stages.

Answer Correctness CoT answers that result in a final answer different from the ground truth are automatically flagged as incorrect. However, for cases where the final answer matches the ground truth, manual annotation is required. This is because there are instances where the reasoning process leading to the correct answer is flawed, as illustrated in the middle example of Figure 1. Therefore, annotators are tasked with meticulously examining the entire reasoning path to determine if the correct final answer is a direct result of the reasoning process.

Error Step This stage is applicable for solutions with either an unmatched final output or a matched final output underpinned by flawed reasoning. Following the prior effort [38], each step in the reasoning process is categorized as positive, neutral, or negative. Positive and neutral steps represent stages where the correct final output remains attainable. Conversely, negative steps indicate a divergence from the path leading to the correct solution. Annotators are required to identify the first step in the reasoning process where the conditions, assumptions, or calculations are incorrect, making the correct final result unreachable for the subsequent reasoning steps.

Error Reason and Correction Annotators are tasked with conducting an in-depth analysis of the reasoning that led to the identified error. As shown in Figure 1, annotators are required to provide the error reason and the corresponding correction to this reasoning step. This comprehensive approach ensures a thorough understanding and rectification of errors in the reasoning process.

3.3 Data Statistics

Table 1 presents the statistics of MR-Ben. The benchmark exhibits a balanced distribution of correct and incorrect solutions, with an overall correct solution rate of 40.3%. Solutions, on average, involve 9.5 steps, and errors typically manifest around the fourth step (4.5). The questions and solutions are substantial, with average lengths of 85.6 and 308.8 words, respectively. The subject-wise analysis reveals that Math is the most challenging, with a correct solution rate of a mere 16.2%. This could be attributable to the intricacy of the arithmetic operations involved. Conversely, Biology emerges as the least daunting, with a high correct solution rate of 59.6%. Coding problems have the longest solutions, averaging 950.3 number of words. This underscores the complexity and the detailed procedural reasoning inherent in coding tasks. Similarly, Logic problems have the longest questions, averaging

Table 1: Statistics of MR-Ben. The length of questions and solutions are measured inthe number of words. Notice that the steps for coding denote the number of lines of code. They are not directly comparable with other subjects.

	Math	Medicine	Biology	Physics	Chemistry	Logic	Coding	Total
Question-Solution Pairs	918	828	1035	667	848	1441	238	5975
Correct Solution Ratio	16.2%	31.0%	59.6%	47.8%	45.0%	51.1%	31.1%	40.3%
Avg Solution Steps	6.8	5.3	5.1	5.7	5.6	5.3	32.5*	9.5
Avg First Error Step	3.1	3.0	2.7	3.1	3.0	2.8	14.0*	4.5
Avg Length of Questions	44.3	88.7	56.3	66.6	48.1	154.8	140.1	85.6
Avg Length of Solutions	205.9	206.1	187.6	199.4	194.5	217.7	950.3	308.8

154.8 words. This is in line with the need for elaborate descriptions in logical reasoning. The typical step at which the first error occurs is fairly consistent across most subjects, usually around the 3rd step out of a total of 5. However, Coding deviates from this trend. The first error tends to appear earlier, specifically around the 14th line out of a total of 32.5 lines. This suggests that the problem-solving process in Coding may have distinct dynamics compared to other subjects.

3.4 Quality Control

Annotators Given the complexity of the questions, which span a range of subjects from high school to professional levels, we enlisted the services of an annotation company. This company meticulously recruited annotators, each holding a minimum of a bachelor's degree. Before their trial labeling, annotators are thoroughly trained and are required to review the annotation guidelines. We've included the guidelines for all subjects in Appendix H for reference. The selection of annotators is based on their performance on a balanced, small hold-out set of problems for each subject. In addition to the annotators, a team of 14 quality controllers diligently monitors the quality of the annotation weekly. As a final layer of assurance, we have 4 meta controllers who scrutinize the quality of the work.

Quality Assurance Every problem in MR-Ben undergoes a rigorous three-round quality assurance process to ensure its accuracy and clarity. Initially, each question is labeled by two different annotators. Any inconsistencies in the solution correctness or the first error step are identified and reviewed by a quality controller for arbitration. Following this, every annotated problem is subjected to a secondary review by annotators who were not involved in the initial labeling. This is to ensure that the annotations for different solutions to the same problem are consistent and coherent. In the final phase of the review, 10% of the problems are randomly sampled and reviewed by the meta controllers. Throughout the entire evaluation process, all annotated fields are meticulously examined in multiple rounds for their accuracy and clarity. Any incorrect annotations or those with disagreements are progressively filtered out and rectified, ensuring a high-quality dataset. This rigorous process allows us to maintain a high level of annotation quality.

Dataset Artifacts & Biases Table 1 reveals a relatively balanced distribution of correct and incorrect solutions. However, an exception was observed in mathematical subjects, where the distribution tends to skew towards incorrect solutions. This skew could suggest an inherent complexity or ambiguity in mathematical problem statements. Our analysis of the first error step across all subjects indicated that errors predominantly occur in the initial stages ($n \leq 7$) of problem-solving and are distributed relatively uniformly. This pattern was consistent across most subjects, with no significant skew towards later steps. More detailed discussions of biases are provided in the Appendix C.

4 Evaluation

For each question-solution pair annotated, the evaluated model are supposed to decide the correctness of the solution and report the first-error-step and error-reason if any. The solution-correctness and first-error-step is scored automatically based on the manual annotation result. Only when the evaluated model correctly identified the incorrect solution and first-error-step will its error-reason be further examined manually or automatically by models. Therefore in order to provide a unified and normalized score to reflect the overall competence of the evaluated model, we follow the work of [77] and apply a metric named **MR-Score**, which consist of three sub-metrics.

The first one is the Matthews Correlation Coefficient (a.k.a MCC, 46) for the binary classification of solution-correctness.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP) \times (TP+FN) \times (TN+FP) \times (TN+FN)}} \tag{1}$$

where TP, TN, FP, FN stand for true positive, true negative, false positive and false negative. The MCC score ranges from -1 to +1 with -1 means total disagreement between prediction and observation, 0 indicates near random performance and +1 represents perfect prediction. In the context of this paper, we interpret negative values as no better than random guess and set 0 as cut-off threshold for normalization purpose.

The second metric is the ratio between numbers of solutions with correct first-error-step predicted and the total number of incorrect solutions.

$$ACC_{\text{step}} = \frac{N_{\text{correct\_first\_error\_step}}}{N_{\text{incorrect\_sols}}} \tag{2}$$

The third metrics is likewise the ratio between number of solutions with correct first-error-step plus correct error-reason predicted and the total number of incorrect solutions.

$$ACC_{\text{reason}} = \frac{N_{\text{correct\_error\_reason}}}{N_{\text{incorrect\_sols}}} \tag{3}$$

MR-Score is then a weighted combination of three metrics, given by

$$MR\text{-}Score = w_1 * \max(0, MCC) + w_2 * ACC_{\text{step}} + w_3 * ACC_{\text{reason}} \tag{4}$$

For the weights w_1, w_2 and w_3, they are chosen based on our evaluation results to maximize the differentiation between different models. It is important to note that the Matthews Correlation Coefficient (MCC) and the accuracy of locating the first error step can be directly calculated by comparing the responses of the evaluated model with the ground truth annotations. However, assessing the accuracy of the error reason explained by the evaluated model presents more complexity. While consulting domain experts for annotations is a feasible approach, we instead utilized GPT-4-Turbo as a proxy to examine the error reasons, as detailed in Figure-11 in Appendix-D.

We operate under the assumption that while our benchmark presents a significant challenge for GPT-4 in evaluating complete solution correctness—identifying the first error step and explaining the error reason—it is comparatively easier for GPT-4 to assess whether the provided error reasons align with the ground truth. Specifically, in a hold-out set of sampled error reasons, there was a 92% agreement rate between the manual annotations by the authors and those generated by GPT-4. For more detailed evaluations on the robustness of MR-Score and its design thinking, please refer to our discussion in Appendix-B.

5 Experiments

5.1 Experiment Setup

To evaluate the performance of different models on our new benchmark, we selected a diverse array of models based on size and source accessibility [2]. This included smaller models like Gemma-2B[63], Phi-3[1], Qwen1.5-1.8B [7], as well as larger counterparts such as Llama3-70B [47], Deepseek-67B[10], and Qwen1.5-72B[7]. We also compared open-source models (e.g. models from the Llama3 and Qwen1.5/Qwen2 series) against closed-source models from the GPT [51], Claude [6], Mistral [32], GLM [3], Yi [39], Moonshot [2], Doubao [12] families. Additionally, models from the Deepseek-Coder [10] series were included to assess the impact of coding-focused pretraining on reasoning performance.

Given the complexity of our benchmark, even larger open-source models like Llama3-70B-Instruct struggle to produce accurate evaluation results without the use of prompting methods, often achieving

[2]Note: All models used in our experiments are instruction-finetuned versions, although this is not indicated in their abbreviated names

Table 2: Evaluation results on MR-Ben: This table presents a detailed breakdown of each model's performance evaluated under metric MR-Score across different subjects, where K stands for the number of demo examples here.

Model	**Bio.**		**Phy.**		**Math**		**Chem.**		**Med.**		**Logic**		**Coding**		**Avg.**	
	k=0	k=1	k=0	k=1	k=0	k=1	k=0	k=1	k=0	k=1	k=0	k=1	k=0	k=1	k=0	k=1
Closed-Source LLMs																
Claude3-Haiku	5.7	5.8	3.3	3.5	3.1	3.1	6.5	6.4	2.0	2.0	1.2	1.2	9.0	0.0	4.4	3.1
GPT-3.5-Turbo	3.6	6.6	5.7	6.7	5.7	5.4	4.9	6.7	3.6	4.4	1.7	4.5	3.0	4.1	4.0	5.5
Doubao-pro-4k	8.4	13.5	10.0	11.7	12.3	15.5	10.6	17.5	5.9	10.0	4.5	5.5	9.8	7.4	8.8	11.6
Mistral-Large	22.2	28.0	26.7	25.4	24.3	28.2	24.0	27.0	15.9	19.3	14.7	17.1	21.1	21.4	21.3	23.8
Yi-Large	35.3.	40.7	37.2	36.8	36.5	20.6	40.0	39.1	29.3	32.1	25.1	31.3	21.9	25.7	32.2	32.3
Moonshot-v1-8k	35.0	36.8	33.8	33.8	34.9	33.0	36.7	35.0	29.4	32.3	25.0	29.2	32.7	31.2	32.5	33.0
GPT-4o-mini	37.7	38.9	38.5	37.4	44.4	40.4	39.2	37.0	33.9	25.1	23.6	17.7	41.6	34.9	37.0	33.1
Zhipu-GLM-4	40.7	46.2	37.7	42.5	38.4	36.6	43.1	44.0	34.5	41.0	37.5	32.5	38.8	32.8	38.7	39.4
GPT-4-Turbo	44.7	47.3	42.8	45.2	44.3	45.4	44.0	46.0	38.8	38.4	34.1	33.6	53.6	57.3	43.2	44.7
GPT-4o	48.3	49.1	45.5	48.2	42.6	41.3	48.2	49.1	47.9	47.7	31.9	28.4	56.5	54.6	45.8	45.5
o1-mini	45.8	46.9	56.0	53.8	68.5	67.0	55.2	56.1	45.9	47.2	30.7	28.7	55.1	55.6	51.0	50.8
o1-preview	54.1	56.0	62.2	61.7	69.8	70.3	60.6	60.3	54.3	55.1	46.1	45.3	65.1	70.0	**58.9**	**59.8**
Open-Source Small																
Qwen1.5-1.8B	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.1	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0
Gemma-2B	0.1	0.0	0.0	0.0	0.0	1.0	0.1	0.0	0.0	0.4	0.0	0.2	0.7	0.0	0.1	0.2
Qwen2-1.5B	2.2	2.8	2.2	1.3	3.3	6.3	2.5	3.3	2.9	11.2	1.5	9.4	0.0	3.6	2.1	5.4
Phi3-3.8B	13.4	12.5	12.7	10.8	13.3	13.1	16.4	17.1	10.2	8.1	8.4	5.3	9.1	10.2	**11.9**	**11.0**
Open-Source LLMs Medium																
GLM-4-9B	4.4	2.4	9.6	1.2	8.1	4.7	8.7	2.9	2.3	1.9	2.5	1.6	11.4	0.0	6.7	2.1
DeepSeek-7B	5.7	6.2	4.7	2.6	4.9	5.2	4.2	4.9	3.1	1.6	3.0	3.8	0.0	1.2	3.7	3.6
Deepseek-Coder-33B	7.4	5.5	7.8	5.6	7.2	8.6	7.8	7.4	6.0	5.5	4.6	6.7	8.4	4.9	7.0	6.3
DeepSeek-Coder-7B	10.5	9.9	11.8	9.6	11.8	12.1	12.3	11.9	10.4	11.0	9.8	10.7	5.0	5.8	10.2	10.2
LLaMA3-8B	12.0	11.9	10.9	7.5	15.0	9.0	12.6	12.7	9.3	8.0	9.4	9.6	15.8	10.0	**12.2**	9.8
Yi-1.5-9B	10.4	14.8	11.9	12.9	12.5	15.6	13.1	14.4	9.5	14.8	9.1	9.5	4.8	6.3	10.2	**12.6**
Open-Source LLMs Large																
Qwen1.5-72B	15.3	19.2	12.9	13.6	12.0	10.0	13.9	16.3	11.7	14.7	10.4	12.9	3.9	5.9	11.5	13.3
DeepSeek-67B	17.1	19.7	14.9	17.3	15.4	16.2	16.3	20.6	14.7	12.2	13.6	14.3	14.5	15.2	15.2	16.5
LLaMA3-70B	20.4	27.1	17.4	20.5	14.9	15.8	19.5	25.1	16.3	19.3	16.3	16.8	29.8	16.7	19.2	20.2
DeepSeek-V2-236B	30.0	37.1	32.2	36.5	32.2	30.0	32.5	35.4	26.5	32.4	23.6	27.4	34.2	27.1	30.2	32.3
Qwen2-72B	36.0	40.8	36.7	40.9	38.0	38.7	37.2	38.8	28.3	29.3	25.6	20.5	31.3	30.4	**33.3**	**34.2**

MR-Scores near zero. Consequently, we employed a step-wise chain-of-thought prompting technique similar to those described in [77, 64]. This approach guides models in systematically reasoning through solution traces before making final decisions, as detailed in Appendix-D.

Considering the complexity of the task, which includes question comprehension, reasoning through the provided solutions, and adhering to format constraints, few-shot demonstration setups are also explored to investigate if models can benefit from In-Context Learning (ICL) examples. Due to the context token limits, we report zero and one-shot results in the main result table (Table 2)[3]. The performance of additional few-shot configurations on a selection of models with various capabilities is further discussed in Section 6.1.

5.2 Experiment Results

The MR-Ben benchmark presents a significant shift in the challenge for state-of-the-art large language models, transitioning from question-answering to the nuanced role of question-solution scoring. This section details our findings, emphasizing variations in model performances and their implications.

Overall Performance Among the evaluated models, o1-preview consistently achieves the highest MR-Scores across all subjects, significantly outperforming most competitors from both open and

[3]For the breakdown performances of models in the sub-tasks, please refer to Table 7

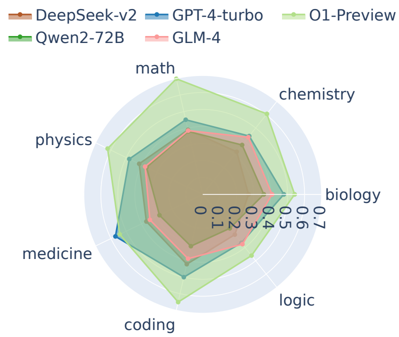

Figure 2: Model performance across subjects

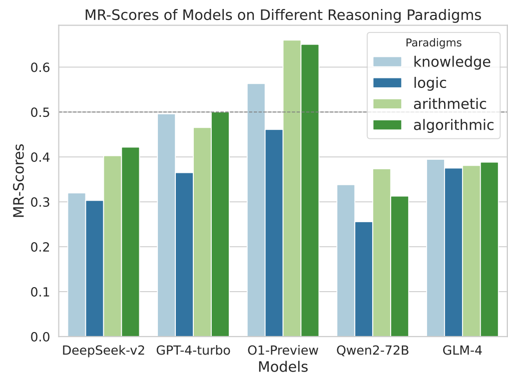

Figure 3: Model performance on different reasoning paradigms

closed-source communities. Notably, the open-sourced Qwen2-72B and Deepseek-V2-236B models are performing exceptionally well, surpassing every other open-sourced model including Llama3 by a large margin. Their scores are even comparable to or greater than some of the most capable models from commercial companies, such as Mistral, Yi, and Moonshot AI. In the small language model category, the performance of Phi3-3.8B exceeds many of the mid-size models, including Deepseek-Coder-33B, whose size is around tenfold larger.

Performance across Model Size and Reasoning Paradigm Table 2 reveals a general trend where larger models tend to perform better, highlighting the correlation between model size and the efficacy in complex reasoning tasks. However, this relationship is not strictly linear, as demonstrated by models like Phi3-3.8B, which excel despite their smaller size. Since MR-Ben challenges the language models to reason about the reasoning in the solution space among a diverse range of domains, models like Phi-3 that are trained with effective data synthesis techniques and broader coverage of the solution space, intuitively achieve higher MR-Score. This suggests that while larger model sizes generally yield superior performances, techniques like knowledge distillation can also significantly boost reasoning performance. Similarly, although the size of the o1 model series remains undisclosed, these models reportedly employ mechanisms that scale computation efficiently through effective exploration, frequent retrospection, and meticulous reflection within the solution space. These characteristics align closely with the principles of "system-2" thinking, which emphasizes deliberate, reflective problem-solving. As a result, the o1 models demonstrate a more effective reasoning process, achieving significantly higher MR-Scores than other models by a large margin.

Performance across Reasoning Types Our categorization into four reasoning types—knowledge, arithmetic, algorithmic, and logic—illustrates the unique challenges each model faces within these paradigms (Figure 3). Logic reasoning emerges as the most formidable due to the intricate logical operations required by questions from the LogiQA dataset. In stark contrast, o1-Preview and GPT-4-turbo demonstrate exceptional prowess in algorithmic reasoning, where their capabilities markedly surpass other models. Notably, models excel in different reasoning paradigms, reflecting their varied strengths and training backgrounds. For instance, despite Deepseek-Coder's specialized pre-training focused on coding tasks, it does not necessarily confer superior abilities in algorithmic reasoning, underscoring that targeted pretraining does not guarantee enhanced performance across all reasoning types. Comparing the performance of the Deepseek-Coder with that of the Phi-3 model, which excels despite its much smaller size, highlights the potential significance of high-quality synthetic data in achieving broad-based reasoning capabilities.

Sensitivity to Task Difficulty and Solution Length An examination across educational levels shows most models perform better at high school-level questions than college-level ones, indicating an intuitive level of sensitivity to the difficulty levels of the questions. Additionally, our analysis finds a minor negative correlation between the length of solution steps and MR-Scores, as detailed in Figure 4 and Figure 5.

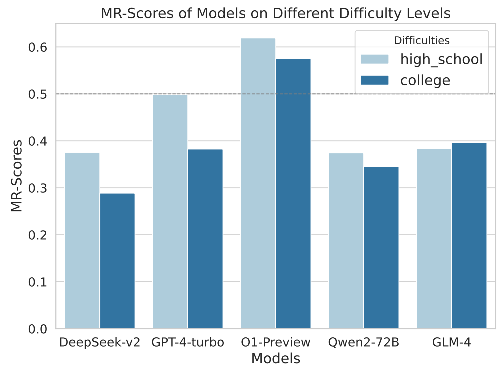

Figure 4: MR-Scores of different models on different levels of difficulty

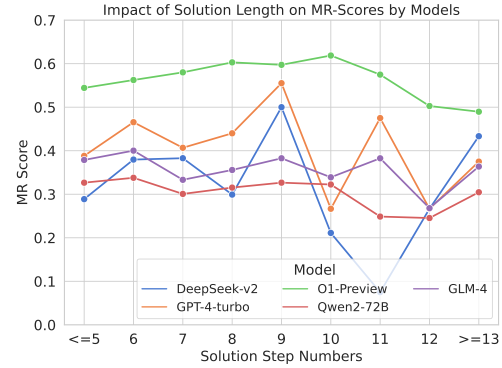

Figure 5: The MR-Scores of models on solutions with different step numbers.

Summary: MR-Ben effectively differentiates model capabilities, often obscured in simpler settings. It not only identifies top performers but also underscores the influence of model size on outcomes, while demonstrating that techniques like knowledge distillation and test-time compute scaling, as seen with the Phi-3 and o1 models, can notably enhance smaller models' performance, challenging the dominance of larger models. The analysis further reveals that specialized training, such as in coding, does not guarantee superior algorithmic reasoning. This suggests the potential need for more balanced data approaches or improved data synthesis methods.

6 Further Analysis & Discussion

6.1 Few Shot Prompting

As previously discussed and exemplified by our prompt template (Figure 10 in Appendix-D), our evaluation method is characterized by its high level of difficulty and complexity. In this experiment, we aimed to determine whether providing a few step-wise chain-of-thought (CoT) examples could improve model performance in terms of format adherence and reasoning quality. The results, as presented in Table 9 in Appendix, do not show a consistent pattern as the number of shots increases. While smaller language models like Gemma-2B exhibit performance improvements with additional shots, the performance of larger language models tends to fluctuate with an increasing number of shots. We hypothesize that for our complex tasks, the lengthy few-shot demonstrations may act more as a hindrance, providing distracting information rather than aiding in format adherence and reasoning. Our empirical findings suggest that a one-shot demonstration strikes the optimal balance between providing guidance and minimizing distraction. This supports our decision to focus on zero-shot versus one-shot comparisons in our primary experiments, as detailed in Table 2.

6.2 Self Refine Prompting

As suggested by [31], large language models typically cannot perform self-correction without external ground truth feedback. To explore whether this phenomenon occurs in our benchmark, we adopted a similar setting by prompting the language model to verify its own answer across a three-round interaction sequence: query, examine, and refine. Our prompting template, detailed in Figure 8 in Appendix D, is minimalistic and designed solely to encourage the model to self-examine.

The results of this self-refinement process are recorded in Table 4. Notably, models smaller than Llama3-70B exhibit performance degradation with self-refinement, while larger models, such as GPT-4, show marginal benefits from the process. Conversely, from Llama3-8B to Llama3-70B, despite a significant portion of correct predictions shifting to incorrect ones, as previously reported by [31], our benchmark shows an increasing trend of incorrect predictions shifting to correct ones as model size increases. This shift results in the significant performance improvements observed in models like Llama3-70B.

To understand the disproportionate improvement observed in the 70B model, we analyzed performance breakdown at the task level. These results are visualized and discussed in Figure 9 of

Table 3: Comparison of average accuracy in identifying the first error step and the corresponding error reason, with and without prior knowledge of the solutions' correctness.

Model	Detection Acc.		Reason Acc.	
	w/o	with	w/o	with
Gemma-2B	0.3	0.1	0.1	0.0
Llama3-8B	15.5	26.4	6.6	11.9
Llama3-70B	14.5	34.6	9.1	25.7
GPT-4-Turbo	40.9	41.6	37.9	38.0

Table 4: Comparison of prompting methods: MR-Scores achieved by zero-shot step-wise CoT and Self-Refine technique.

Model	0-shot CoT	Self-Refine
Gemma-2B	0.1	0.2
Llama3-8B	11.7	11.3
Llama3-70B	17.7	27.5
GPT-4-Turbo	43.2	45.5

Appendix E. In short, we believe the lack of consistency does not necessarily indicate a more robust or advanced reasoning ability, despite the increase of the evaluation results.

6.3 Solution Correctness Prior

To verify the influence of external ground truth signals, we sampled 100 incorrect solutions from each subject respectively as our test set. By observing the same set of language models under a zero-shot CoT setting, we aim to determine whether the knowledge of the solution's incorrectness enhances their ability to identify the first error step and the reason for the error.

The results in Table 3 illustrate that the benefits of knowing the solution correctness prior generally increase with the model's competence but begin to plateau at the level of sophisticated models like GPT-4. Specifically, the Gemma-2b model struggles significantly in our benchmark, showing nearly zero performance due to its limited ability to follow formats and comprehend complex tasks. Consequently, having the solution correctness prior does not improve its performance metrics. In contrast, models with moderate capabilities benefit substantially from this prior knowledge, which aids in accurately locating the first error step and elucidating the error reason. However, as model capabilities improve, the incremental benefits of this prior knowledge quickly diminish. For instance, GPT-4 shows only a marginal improvement in identifying the first error step and an almost negligible impact on error reason analysis when provided with the prior.

7 Conclusion

This paper highlights the importance of evaluating the reasoning capabilities of LLMs with process-oriented design and presents a comprehensive benchmark called MR-Ben that addresses the limitations of existing evaluation methodologies. MR-Ben consists of questions from a diverse range of subjects and incorporates a meta-reasoning paradigm, where LLMs act as teachers to evaluate the reasoning process. Our evaluation of a diverse suite of LLMs on MR-Ben reveals several key limitations and weaknesses. Many models struggle with identifying and correcting errors within reasoning chains, demonstrating difficulty in performing system-2 style thinking—such as scrutinizing assumptions, calculations, and intermediate steps. Furthermore, even state-of-the-art models often fail to maintain consistency across reasoning paradigms, exposing gaps in their generalization abilities. Additionally, our findings emphasize the importance of searching and reflecting on the solution space during inference. Models like the o1 series showcase the potential of scaling test-time computation, where frequent retrospection and iterative search through multiple solution paths significantly enhance reasoning performance. Nevertheless, improving LLMs' reasoning abilities on complex and nuanced tasks remains an open research question, and we encourage future work to develop upon MR-Ben.

8 Acknowledgement

This work was supported in part by the Research Grants Council under the Areas of Excellence scheme grant AoE/E-601/22-R.

References

[1] Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

[2] Moonshot AI. Moonshot ai, 2024. URL `https://www.moonshot.cn/`.

[3] Zhipu AI. Welcome to glm-4, 2024. URL `https://en.chatglm.cn/`.

[4] Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2357–2367. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1245. URL `https://doi.org/10.18653/v1/n19-1245`.

[5] Anthropic. Claude 2, 2024. URL `https://www.anthropic.com/news/claude-2`.

[6] Anthropic. Introducing the next generation of claude, 2024. URL `https://www.anthropic.com/news/claude-3-family`.

[7] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[8] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosiute, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemí Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional AI: harmlessness from AI feedback. *CoRR*, abs/2212.08073, 2022. doi: 10.48550/ARXIV.2212.08073. URL `https://doi.org/10.48550/arXiv.2212.08073`.

[9] Yoshua Bengio. Deep learning for system 2 processing. Presentation at the AAAI-20 Turing Award Winners 2018 Special Event, February 9 2020.

[10] Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wenjie Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, Alex X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Junxiao Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Minghui Tang, Bingxuan Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yiliang Xiong, Hanwei Xu, R. X. Xu, Yanhong Xu, Dejian Yang, Yuxiang You, Shuiping Yu, Xingkai Yu, B. Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghua Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. Deepseek LLM: scaling open-source language models with longtermism. *CoRR*, abs/2401.02954, 2024. doi: 10.48550/ARXIV.2401.02954. URL `https://doi.org/10.48550/arXiv.2401.02954`.

[11] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M.

Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *CoRR*, abs/2005.14165, 2020. URL `https://arxiv.org/abs/2005.14165`.

[12] Bytedance. Doubao team - crafting the industry's most advanced llms., 2024. URL `https://www.doubao.com/chat/`.

[13] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Y. Zhao, Yanping Huang, Andrew M. Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *CoRR*, abs/2210.11416, 2022. doi: 10.48550/ARXIV.2210.11416. URL `https://doi.org/10.48550/arXiv.2210.11416`.

[14] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the AI2 reasoning challenge. *CoRR*, abs/1803.05457, 2018. URL `http://arxiv.org/abs/1803.05457`.

[15] Peter Clark, Oyvind Tafjord, and Kyle Richardson. Transformers as soft reasoners over language. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3882–3890. ijcai.org, 2020. doi: 10.24963/IJCAI.2020/537. URL `https://doi.org/10.24963/ijcai.2020/537`.

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

[17] Jianbo Dai, Jianqiao Lu, Yunlong Feng, Rongju Ruan, Ming Cheng, Haochen Tan, and Zhijiang Guo. Mhpp: Exploring the capabilities and limitations of language models beyond basic code generation. *arXiv preprint arXiv:2405.11430*, 2024.

[18] Bhavana Dalvi, Peter Jansen, Oyvind Tafjord, Zhengnan Xie, Hannah Smith, Leighanna Pipatanangkura, and Peter Clark. Explaining answers with entailment trees. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 7358–7370. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.EMNLP-MAIN.585. URL `https://doi.org/10.18653/v1/2021.emnlp-main.585`.

[19] Ronald Fagin and Joseph Y. Halpern. Reasoning about knowledge and probability. *J. ACM*, 41(2):340–367, 1994. doi: 10.1145/174652.174658. URL `https://doi.org/10.1145/174652.174658`.

[20] Patrick Fernandes, Aman Madaan, Emmy Liu, António Farinhas, Pedro Henrique Martins, Amanda Bertsch, José G. C. de Souza, Shuyan Zhou, Tongshuang Wu, Graham Neubig, and André F. T. Martins. Bridging the gap: A survey on integrating (human) feedback for natural language generation. *CoRR*, abs/2305.00955, 2023. doi: 10.48550/ARXIV.2305.00955. URL `https://doi.org/10.48550/arXiv.2305.00955`.

[21] Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y. Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. RARR: researching and revising what language models say, using language models. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 16477–16508. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.910. URL `https://doi.org/10.18653/v1/2023.acl-long.910`.

[22] Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies. *Trans. Assoc. Comput. Linguistics*, 9:346–361, 2021. doi: 10.1162/TACL_A_00370. URL https://doi.org/10.1162/tacl_a_00370.

[23] Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. ROSCOE: A suite of metrics for scoring step-by-step reasoning. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=xYlJRpzZtsY.

[24] Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: large language models can self-correct with tool-interactive critiquing. *CoRR*, abs/2305.11738, 2023. doi: 10.48550/ARXIV.2305.11738. URL https://doi.org/10.48550/arXiv.2305.11738.

[25] Alex Gu, Baptiste Rozière, Hugh James Leather, Armando Solar-Lezama, Gabriel Synnaeve, and Sida Wang. Cruxeval: A benchmark for code reasoning, understanding and execution. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=Ffpg52swvg.

[26] Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need. *CoRR*, abs/2306.11644, 2023. doi: 10.48550/ARXIV.2306.11644. URL https://doi.org/10.48550/arXiv.2306.11644.

[27] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

[28] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In Joaquin Vanschoren and Sai-Kit Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

[29] Dong Huang, Jianbo Dai, Han Weng, Puzhen Wu, Yuhao Qing, Jie M. Zhang, Heming Cui, and Zhijiang Guo. SOAP: enhancing efficiency of generated code via self-optimization. *CoRR*, abs/2405.15189, 2024. doi: 10.48550/ARXIV.2405.15189. URL https://doi.org/10.48550/arXiv.2405.15189.

[30] Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 1049–1065. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-ACL.67. URL https://doi.org/10.18653/v1/2023.findings-acl.67.

[31] Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*, 2023.

[32] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *CoRR*, abs/2310.06825, 2023. doi: 10.48550/ARXIV.2310.06825. URL https://doi.org/10.48550/arXiv.2310.06825.

[33] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *CoRR*, abs/2401.04088, 2024. doi: 10.48550/ARXIV. 2401.04088. URL `https://doi.org/10.48550/arXiv.2401.04088`.

[34] Jaehun Jung, Lianhui Qin, Sean Welleck, Faeze Brahman, Chandra Bhagavatula, Ronan Le Bras, and Yejin Choi. Maieutic prompting: Logically consistent reasoning with recursive explanations. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 1266–1279. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.82. URL `https://doi.org/10.18653/v1/2022.emnlp-main.82`.

[35] Daniel Kahneman. Thinking, fast and slow. *Farrar, Straus and Giroux*, 2011.

[36] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL `http://papers.nips.cc/paper_files/paper/2022/hash/8bb0d291acd4acf06ef112099c16f326-Abstract-Conference.html`.

[37] Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. MAWPS: A math word problem repository. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1152–1157. The Association for Computational Linguistics, 2016. doi: 10.18653/V1/N16-1136. URL `https://doi.org/10.18653/v1/n16-1136`.

[38] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *CoRR*, abs/2305.20050, 2023. doi: 10.48550/ARXIV.2305.20050. URL `https://doi.org/10.48550/arXiv.2305.20050`.

[39] LingYiWanWu. Yi ai, 2024. URL `https://platform.lingyiwanwu.com/`.

[40] Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 3622–3628. ijcai.org, 2020. doi: 10.24963/IJCAI.2020/501. URL `https://doi.org/10.24963/ijcai.2020/501`.

[41] Yinhong Liu, Zhijiang Guo, Tianya Liang, Ehsan Shareghi, Ivan Vulić, and Nigel Collier. Measuring, evaluating and improving logical consistency in large language models. *arXiv preprint arXiv:2410.02205*, 2024.

[42] Yinhong Liu, Han Zhou, Zhijiang Guo, Ehsan Shareghi, Ivan Vulić, Anna Korhonen, and Nigel Collier. Aligning with human judgement: The role of pairwise preference in large language model evaluators, 2024.

[43] Yiqi Liu, Nafise Sadat Moosavi, and Chenghua Lin. Llms as narcissistic evaluators: When ego inflates evaluation scores, 2024.

[44] Jianqiao Lu, Zhiyang Dou, Hongru Wang, Zeyu Cao, Jianbo Dai, Yingjia Wan, Yinya Huang, and Zhijiang Guo. Autocv: Empowering reasoning with automated process labeling via confidence variation. *CoRR*, abs/2405.16802, 2024. doi: 10.48550/ARXIV.2405.16802. URL `https://doi.org/10.48550/arXiv.2405.16802`.

[45] Jianqiao Lu, Zhengying Liu, Yingjia Wan, Yinya Huang, Haiming Wang, Zhicheng Yang, Jing Tang, and Zhijiang Guo. Process-driven autoformalization in lean 4. *CoRR*, abs/2406.01940, 2024. doi: 10.48550/ARXIV.2406.01940. URL `https://doi.org/10.48550/arXiv.2406.01940`.

[46] Brian W. Matthews. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et biophysica acta*, 405 2:442–51, 1975. URL `https://api.semanticscholar.org/CorpusID:44596673`.

[47] Meta. Introducing meta llama 3: The most capable openly available llm to date, 2024. URL `https://ai.meta.com/blog/meta-llama-3/`.

[48] Swaroop Mishra, Matthew Finlayson, Pan Lu, Leonard Tang, Sean Welleck, Chitta Baral, Tanmay Rajpurohit, Oyvind Tafjord, Ashish Sabharwal, Peter Clark, and Ashwin Kalyan. LILA: A unified benchmark for mathematical reasoning. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5807–5832. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.EMNLP-MAIN.392. URL `https://doi.org/10.18653/v1/2022.emnlp-main.392`.

[49] Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of GPT-4. *CoRR*, abs/2306.02707, 2023. doi: 10.48550/ARXIV.2306.02707. URL `https://doi.org/10.48550/arXiv.2306.02707`.

[50] OpenAI. GPT-3.5 Turbo, 2023. URL `https://platform.openai.com/docs/models/gpt-3-5`.

[51] OpenAI. GPT-4 Technical Report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/arXiv.2303.08774. URL `https://doi.org/10.48550/arXiv.2303.08774`.

[52] OpenAI. Introducing openai o1-preview, 2024. URL `https://openai.com/index/introducing-openai-o1-preview/`.

[53] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL `http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html`.

[54] Arjun Panickssery, Samuel R. Bowman, and Shi Feng. Llm evaluators recognize and favor their own generations, 2024.

[55] Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 2080–2094. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.NAACL-MAIN.168. URL `https://doi.org/10.18653/v1/2021.naacl-main.168`.

[56] Archiki Prasad, Swarnadeep Saha, Xiang Zhou, and Mohit Bansal. Receval: Evaluating reasoning chains via correctness and informativeness. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 10066–10086. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.622. URL `https://doi.org/10.18653/v1/2023.emnlp-main.622`.

[57] Shuofei Qiao, Yixin Ou, Ningyu Zhang, Xiang Chen, Yunzhi Yao, Shumin Deng, Chuanqi Tan, Fei Huang, and Huajun Chen. Reasoning with language model prompting: A survey. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 5368–5393. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.ACL-LONG.294. URL `https://doi.org/10.18653/v1/2023.acl-long.294`.

[58] Abulhair Saparov and He He. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL `https://openreview.net/pdf?id=qFVVBzXxR2V`.

[59] Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazarv, Alice Xiang, Alicia Parrish, Allen Nie, Aman Hussain, Amanda Askell, Amanda Dsouza, Ameet Rahane, Anantharaman S. Iyer, Anders Andreassen, Andrea Santilli, Andreas Stuhlmüller, Andrew M. Dai, Andrew La, Andrew K. Lampinen, Andy Zou, Angela Jiang, Angelica Chen, Anh Vuong, Animesh Gupta, Anna Gottardi, Antonio Norelli, Anu Venkatesh, Arash Gholamidavoodi, Arfa Tabassum, Arul Menezes, Arun Kirubarajan, Asher Mullokandov, Ashish Sabharwal, Austin Herrick, Avia Efrat, Aykut Erdem, Ayla Karakas, and et al. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *CoRR*, abs/2206.04615, 2022. doi: 10.48550/ARXIV.2206.04615. URL `https://doi.org/10.48550/arXiv.2206.04615`.

[60] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 13003–13051. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-ACL.824. URL `https://doi.org/10.18653/v1/2023.findings-acl.824`.

[61] Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. Proofwriter: Generating implications, proofs, and abductive statements over natural language. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 3621–3634. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.FINDINGS-ACL.317. URL `https://doi.org/10.18653/v1/2021.findings-acl.317`.

[62] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4149–4158. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1421. URL `https://doi.org/10.18653/v1/n19-1421`.

[63] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[64] Gladys Tyen, Hassan Mansoor, Peter Chen, Tony Mak, and Victor Carbune. Llms cannot find reasoning errors, but can correct them! *CoRR*, abs/2311.08516, 2023. doi: 10.48550/ARXIV.2311.08516. URL `https://doi.org/10.48550/arXiv.2311.08516`.

[65] Peter Cathcart Wason and Philip Nicholas Johnson-Laird. *Psychology of reasoning: Structure and content*, volume 86. Harvard University Press, 1972.

[66] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

[67] Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. Generating sequences by learning to self-correct. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=hH36JeQZDaO.

[68] Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. Evaluating mathematical reasoning beyond accuracy. *CoRR*, abs/2404.05692, 2024. doi: 10.48550/ARXIV.2404.05692. URL https://doi.org/10.48550/arXiv.2404.05692.

[69] Jing Xiong, Zixuan Li, Chuanyang Zheng, Zhijiang Guo, Yichun Yin, Enze Xie, Zhicheng Yang, Qingxing Cao, Haiming Wang, Xiongwei Han, Jing Tang, Chengming Li, and Xiaodan Liang. Dq-lore: Dual queries with low rank approximation re-ranking for in-context learning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=qAoxvePSlq.

[70] Kevin Yang, Yuandong Tian, Nanyun Peng, and Dan Klein. Re3: Generating longer stories with recursive reprompting and revision. *CoRR*, abs/2210.06774, 2022. doi: 10.48550/ARXIV.2210.06774. URL https://doi.org/10.48550/arXiv.2210.06774.

[71] Zonglin Yang, Li Dong, Xinya Du, Hao Cheng, Erik Cambria, Xiaodong Liu, Jianfeng Gao, and Furu Wei. Language models as inductive reasoners. In Yvette Graham and Matthew Purver, editors, *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2024 - Volume 1: Long Papers, St. Julian's, Malta, March 17-22, 2024*, pages 209–225. Association for Computational Linguistics, 2024. URL https://aclanthology.org/2024.eacl-long.13.

[72] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/271db9922b8d1f4dd7aaef84ed5ac703-Abstract-Conference.html.

[73] Yuxuan Yao, Han Wu, Zhijiang Guo, Biyan Zhou, Jiahui Gao, Sichun Luo, Hanxu Hou, Xiaojin Fu, and Linqi Song. Learning from correctness without prompting makes LLM efficient reasoner. *CoRR*, abs/2403.19094, 2024. doi: 10.48550/ARXIV.2403.19094. URL https://doi.org/10.48550/arXiv.2403.19094.

[74] Michihiro Yasunaga, Xinyun Chen, Yujia Li, Panupong Pasupat, Jure Leskovec, Percy Liang, Ed H. Chi, and Denny Zhou. Large language models as analogical reasoners. *CoRR*, abs/2310.01714, 2023. doi: 10.48550/ARXIV.2310.01714. URL https://doi.org/10.48550/arXiv.2310.01714.

[75] Xi Ye and Greg Durrett. The unreliability of explanations in few-shot prompting for textual reasoning. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/c402501846f9fe03e2cac015b3f0e6b1-Abstract-Conference.html.

[76] Wenhao Yu, Zhihan Zhang, Zhenwen Liang, Meng Jiang, and Ashish Sabharwal. Improving language models via plug-and-play retrieval feedback. *CoRR*, abs/2305.14002, 2023. doi: 10.48550/ARXIV.2305.14002. URL https://doi.org/10.48550/arXiv.2305.14002.

[77] Zhongshen Zeng, Pengguang Chen, Shu Liu, Haiyun Jiang, and Jiaya Jia. Mr-gsm8k: A meta-reasoning benchmark for large language model evaluation. *CoRR*, abs/2312.17080, 2023. doi: 10.48550/ARXIV.2312.17080. URL `https://doi.org/10.48550/arXiv.2312.17080`.

[78] Chi Zhang, Baoxiong Jia, Mark Edmonds, Song-Chun Zhu, and Yixin Zhu. ACRE: abstract causal reasoning beyond covariation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 10643–10653. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.01050. URL `https://openaccess.thecvf.com/content/CVPR2021/html/Zhang_ACRE_Abstract_Causal_REasoning_Beyond_Covariation_CVPR_2021_paper.html`.

[79] Han Zhou, Xingchen Wan, Lev Proleev, Diana Mincu, Jilin Chen, Katherine A Heller, and Subhrajit Roy. Batch calibration: Rethinking calibration for in-context learning and prompt engineering. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=L3FHMoKZcS`.

[80] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL `https://openreview.net/pdf?id=92gvk82DE-`.

A Appendix

A.1 Limitations

The meta-reasoning evaluation framework in MR-Ben, while innovative, is not without its limitations. Firstly, its applicability may be restricted when it comes to subjects that are inherently holistic or creative in nature, such as humanities or sociology. These subjects often require a comprehensive understanding and modification (e.g. essay writing), which can be challenging to break down into specific, sequential reasoning steps and corrections. Secondly, MR-Ben is currently confined to questions in English. This could potentially limit the scope of reasoning challenges that can be explored, as different languages may present unique cognitive and linguistic hurdles. Lastly, the analysis and correction of errors in the reasoning steps are currently based on solutions generated by three LLMs, namely GPT-3.5, Mistral-Medium, and Claude 2. It's important to note that different LLMs and different individuals, may exhibit distinct reasoning and error patterns. Therefore, it would be beneficial to broaden the spectrum of solutions analyzed, incorporating a more diverse range of LLMs and even human responses. This would not only enhance the robustness of the evaluation framework but also provide a more nuanced understanding of the reasoning processes at play.

A.2 Broader Impact

Positive Societal Impacts The proposed dataset MR-Ben has the potential to bring about significant positive societal impacts. It can contribute to the development and enhancement of LLMs by providing a comprehensive benchmark suite, which researchers and developers can use to identify and address the limitations and weaknesses of their models. This can lead to more accurate, efficient, and reliable LLMs. The meta-reasoning paradigm might open new avenues in AI research, leading to a deeper understanding of reasoning capabilities and the development of innovative methodologies for their evaluation and improvement. Moreover, with a wide range of subjects, MR-Ben can be a valuable resource for educational AI tools, providing personalized learning experiences and helping students understand and improve their reasoning skills. AI systems with improved reasoning capabilities can also be instrumental in various sectors, including healthcare, finance, and environmental management, aiding in complex decision-making and problem-solving tasks.

Negative Societal Impacts MR-Ben may also present potential negative societal impacts. As with any technology, there is a risk of LLMs being misused or used maliciously. For instance, LLMs with advanced reasoning capabilities could be used to manipulate information or deceive people. The use of LLMs in decision-making and problem-solving tasks could lead to an over-reliance on these systems, potentially undermining human judgment and critical thinking skills. Advanced LLMs, especially those used in sensitive sectors like healthcare and finance, need to handle vast amounts of data, which can raise privacy and security concerns if not managed properly.

A.3 Additional Related Work

Improving Reasoning Abilities of LLMs To enhance the reasoning capabilities of LLMs, prior research primarily focuses on specific prompting techniques [11]. Existing efforts include few-shot prompting with intermediate steps augmented demonstrations [66, 72, 69] or zero-shot prompting with specific instructions [36, 74]. Although these methods have shown promising results, their effectiveness is often constrained by their task-specific nature and the labour-intensive process of designing prompts, leading to inconsistent outcomes across different tasks [75, 80]. Another strategy to facilitate reasoning involves instruction tuning or knowledge distillation, which elicits reasoning paths from LLMs without explicit prompting [13, 49, 26, 44]. These approaches typically involve resource-intensive fine-tuning over LLMs and require a large set of examples annotated with CoT.

Learning From Feedback Improving LLMs through learning from feedback has become a prevalent strategy, notably through reinforcement learning from human feedback, which seeks to align LLMs with human values by refining their outputs based on feedback [53, 8]. However, this method faces challenges such as high costs due to manual labor and a lack of real-time feedback capabilities [20]. An alternative strategy involves using self-correcting LLMs, which rely on automated feedback to iteratively adapt and understand the consequences of their actions without relying on hu-

Model	Coding	Phy.	Bio.	Math	Med.	Chem.	Logic
gpt-4-turbo	83/55	137/15	164/11	305/46	194/25	166/27	192/16
deepseek_coder	100/38	145/7	167/8	321/30	200/19	172/21	193/15
Qwen2-72B	99/39	142/10	167/8	312/39	195/24	172/21	200/8

Table 5: Scoring of error reasons from different models across subjects.

	Coding	Phy.	Bio.	Med.	Chem.	Logic	Math
Agreement Ratio	7/8	12/13	21/21	12/12	15/17	15/16	10/13

Table 6: Agreement ratio between the author and the proxy scoring model across different subjects.

mans. This feedback can be derived from outside sources such as other models [70, 45], tools [24, 29], knowledge bases [21, 76], evaluation metrics [34, 67] or generation logits [73].

B Robustness of MR-Score

Question: Does the `ACC_reason` metric's dependency on the judgments of different LLMs or human evaluators lead to variability in scoring ?

Answer: We would like to argue that due to the careful design of our evaluation mechanism, the automatic scoring of error reasons is both robust and economically feasible:

- **Multiple annotators**: During the annotation stage, we collected multiple annotations for the first error reasons and potential error rectification from different annotators who agreed on the solution correctness and the first error step.
- **Proxy Model Evaluation**: Based on the ground truth annotations collected from various perspectives, the proxy language model (e.g., GPT-4-Turbo) then examines the **error reasons** provided by evaluating models. Given the question/solution pair and information regarding the first error step, error reasons, and rectification, the potential flaws of the error reasons provided by the evaluating models are easy to diagnose under contrast.
- **ACC_reason robustness**: Table-5 shows the scores of error reasons sampled from our evaluation results. For the same set of error reasons collected in each subject, three different models made their predictions on correctness/incorrectness. We can clearly see the consistency of their predictions among the three models over questions in all subjects. Since the MR-Score is a weighted metric, the final score variability is less than 1 percent in total.

Human-Model Agreement Rate: As mentioned in 3, the agreement rate between manual annotations and the GPT-4 predictions over 100 samples randomly collected from all subjects is 92%. Below is the exact detail of our setup:

We randomly collected 100 data instances where the evaluating model correctly identified the solution correctness and the first error step across all subjects. We then manually examined whether the proxy scoring model (e.g., GPT-4-Turbo-2024-04-09) correctly scored the error reasons of the evaluating models. Table-6 is the detailed composition of the ratio in which the author agrees with the proxy scoring model. The annotation time varies significantly across subjects, as some problems—such as coding and chemistry—can take more than 10 minutes to evaluate, while subjects like biology are easier to assess. This high agreement rate further supports the reliability of our evaluation, thus avoiding the need for manual annotation of potentially 138,000 problems (6,000 benchmark size times 23 models evaluated).

Question: Is the MR-Score sensitive to different weightings? Is MR-Score a robust unified metric?

Table-7 shows breakdown performance for models in all four metrics (MR-Score, MCC, ACC_step, and ACC_reason):

1. **Metric Robustness**: Due to the progressive nature of the definitions of our subtasks (e.g., the success of subsequent tasks depends on the previous ones), we can see the diminishing trend in the scores of MCC, ACC_step, and ACC_reason. However, thanks to the design

Table 7: Evaluation results breakdown on MR-Ben: This table presents a detailed breakdown of each model's performance evaluated under metric MCC/ACC-step/ACC-reason across different subjects. Here k stands for number of shot and every model we used in this experiment are instruction-tuned.

Model	Bio.		Phy.		Math		Chem.		Med.		Logic		Coding		Avg.	
	k=0	k=1	k=0	k=1	k=0	k=1	k=0	k=1	k=0	k=1	k=0	k=1	k=0	k=1	k=0	k=1
MR-Scores																
Claude3-Haiku	5.7	5.8	3.3	3.5	3.1	3.1	6.5	6.4	2.0	2.0	1.2	1.2	9.0	0.0	4.4	3.1
GPT-3.5-Turbo	3.6	6.6	5.7	6.7	5.7	5.4	4.9	6.7	3.6	4.4	1.7	4.5	3.0	4.1	4.0	5.5
Phi3-3.8B	13.4	12.5	12.7	10.8	13.3	13.1	16.4	17.1	10.2	8.1	8.4	5.3	9.1	10.2	**11.9**	**11.0**
Deepseek-Coder-33B	7.4	5.5	7.8	5.6	7.2	8.6	7.8	7.4	6.0	5.5	4.6	6.7	8.4	4.9	7.0	6.3
DeepSeek-Coder-7B	10.5	9.9	11.8	9.6	11.8	12.1	12.3	11.9	10.4	11.0	9.8	10.7	5.0	5.8	10.2	10.2
LLaMA3-8B	12.0	11.9	10.9	7.5	15.0	9.0	12.6	12.7	9.3	8.0	9.4	9.6	15.8	10.0	**12.2**	9.8
Qwen1.5-72B	15.3	19.2	12.9	13.6	12.0	10.0	13.9	16.3	11.7	14.7	10.4	12.9	3.9	5.9	11.5	13.3
DeepSeek-67B	17.1	19.7	14.9	17.3	15.4	16.2	16.3	20.6	14.7	12.2	13.6	14.3	14.5	15.2	15.2	16.5
LLaMA3-70B	20.4	27.1	17.4	20.5	14.9	15.8	19.5	25.1	16.3	19.3	16.3	16.8	29.8	16.7	19.2	20.2
Mistral-Large	22.2	28.0	26.7	25.4	24.3	28.2	24.0	27.0	15.9	19.3	14.7	17.1	21.1	21.4	21.3	23.8
DeepSeek-V2-236B	30.0	37.1	32.2	36.5	32.2	30.0	32.5	35.4	26.5	32.4	23.6	27.4	34.2	27.1	30.2	32.3
GPT-4-Turbo	44.7	47.3	42.8	45.2	44.3	45.4	44.0	46.0	38.8	38.4	34.1	33.6	53.6	57.3	**43.2**	**44.7**
MCC-Matthews Correlation Coefficient																
Claude3-Haiku	13.96	17.72	16.47	13.62	15.09	10.74	16.54	19.96	8.52	8.35	6.21	4.94	4.36	0	11.59	10.76
GPT-3.5-Turbo	10.72	19.44	16.66	21.33	17.48	17.45	18.24	12.6	11.19	13.28	4.07	0	12.35	12.35	12.96	13.78
Deepseek-Coder-33B	7.51	8.57	11.73	6.81	9.69	21.06	9.98	7.94	1.62	6.28	0	0	26.18	15.44	9.53	9.44
Deepseek-Coder-7B	4.96	9.79	8.77	6.72	9.05	10.82	10.49	9.39	5.02	3.17	3.22	2.58	10.91	6.27	7.49	6.96
LlaMA3-8B	19.37	21.15	16.24	18.64	26.55	21.87	25.99	28.6	14.92	18.95	11.8	16.24	14.54	15.72	18.49	20.17
Phi3-3.8B	27.66	28.48	21.61	21.44	22.29	25.17	30.92	33.37	17.36	14.9	13.03	9.56	14.48	18.76	21.05	21.67
Qwen1.5-72B	33.64	42.44	31.4	31.56	29.2	23.28	35.47	36.47	21.76	29.64	24.42	27.74	13.8	15.69	27.1	29.55
Deepseek-67B	43.61	41.73	24.16	28.77	24.95	23.87	36.58	37.29	27.8	28.93	26.74	25.09	28.23	29.06	30.3	30.68
LlaMA3-70B	45.67	56.14	40.34	41.3	32.76	30.94	41.72	52.12	33.18	37.75	32.0	33.87	47.86	29.67	39.08	40.26
Mistral-Large	41.67	49.0	34.24	33.47	29.0	37.05	41.99	47.07	23.76	32.05	25.66	33.25	37.05	33.52	33.34	37.92
Deepseek-v2-236B	52.96	53.38	41.81	46.48	43.75	40.53	54.32	50.15	37.61	44.53	36.36	35.41	45.89	35.7	44.67	43.74
GPT-4-Turbo	63.33	62.59	52.9	52.7	50.67	52.84	53.05	54.59	56.79	54.66	40.95	42.94	52.5	57.53	52.88	53.98
Accuracy of First Error Step																
Claude3-Haiku	2.15	3.1	1.4	1.12	2.38	1.59	1.77	4.42	1.69	0.68	1.01	0.29	0.0	0.0	1.49	1.6
GPT-3.5-Turbo	2.86	4.53	4.2	4.76	4.37	3.84	2.87	8.17	2.37	3.05	1.73	7.63	0.61	2.44	2.72	4.92
Deepseek-Coder-33B	14.83	10.29	14.94	12.36	14.69	10.92	15.67	16.31	14.54	12.43	12.22	18.18	5.49	3.05	13.2	11.93
Deepseek-Coder-7B	21.77	18.18	23.28	19.83	23.41	20.03	24.46	23.18	23.29	26.09	20.03	24.72	4.27	6.1	20.07	19.73
LlaMA3-8B	14.35	14.35	17.53	8.62	20.29	7.8	14.16	11.59	13.13	8.41	13.64	11.36	17.68	9.76	15.83	10.27
Phi3-3.8B	12.68	11.48	16.38	12.07	17.69	16.12	18.03	17.17	12.78	8.76	10.23	6.96	8.54	9.15	13.76	11.67
Qwen1.5-72B	11.48	15.31	10.63	11.49	10.79	9.88	12.45	14.38	11.03	13.49	8.1	10.94	1.83	4.27	9.47	11.39
Deepseek-67B	13.16	19.14	19.25	21.84	20.81	22.11	17.17	23.39	14.71	12.08	12.78	13.49	12.2	14.02	15.72	18.01
LlaMA3-70B	15.79	22.25	14.66	18.39	13.65	15.08	17.81	22.32	14.36	16.64	14.91	14.91	26.83	13.41	16.86	17.57
Mistral-Large	18.38	25.54	26.33	28.29	27.28	33.25	26.05	27.59	16.92	19.29	14.53	14.96	19.51	19.51	21.29	24.06
Deepseek-v2-236B	27.51	35.41	37.64	40.23	36.28	34.33	33.91	37.55	27.5	32.57	22.87	27.56	33.54	26.83	31.32	33.5
GPT-4-Turbo	41.77	46.06	42.58	46.5	46.49	47.81	42.6	46.8	37.06	36.72	29.93	33.24	50.61	59.15	41.58	45.18
Accuracy of First Error Reason																
Claude3-Haiku	1.67	2.63	0.56	0.84	1.19	0.93	0.22	2.21	1.18	0.34	0.72	0.29	0.0	0.0	0.79	1.03
GPT-3.5-Turbo	1.19	2.63	2.24	1.96	1.85	1.46	0.88	3.53	1.35	1.69	0.72	4.17	0.61	1.83	1.26	2.47
Deepseek-Coder-33B	2.87	1.44	2.01	1.15	1.69	2.21	2.15	1.93	2.63	1.05	1.85	2.56	3.05	1.83	2.32	1.74
Deepseek-Coder-7B	5.98	5.02	6.03	4.6	5.98	7.93	5.79	6.22	4.9	5.08	6.25	5.54	3.05	5.49	5.43	5.7
LlaMA3-8B	7.66	6.7	4.89	2.3	7.28	4.55	6.22	7.08	4.73	3.33	5.97	5.97	15.24	7.93	7.43	5.41
Phi3-3.8B	8.13	6.7	6.9	5.75	7.02	6.37	9.66	10.52	5.78	5.08	5.4	2.7	7.32	7.32	7.17	6.35
Qwen1.5-72B	10.29	12.2	6.9	7.76	5.85	4.81	6.22	9.44	8.06	9.46	6.11	8.24	1.22	3.05	6.38	7.85
Deepseek-67B	8.85	11.24	8.62	10.06	8.32	9.49	7.73	12.23	9.46	5.6	8.81	10.37	10.37	10.37	8.88	9.91
LlaMA3-70B	13.16	18.42	9.77	13.51	8.58	10.27	11.59	15.88	10.68	13.49	10.94	11.08	24.39	13.41	12.73	13.72
Mistral-Large	15.27	21.0	19.05	20.45	15.1	21.72	16.56	18.54	13.03	14.21	11.22	11.94	17.07	17.68	15.33	17.94
Deepseek-v2-236B	22.25	31.58	25.57	30.17	25.1	23.28	22.96	28.11	21.54	27.5	18.89	24.15	29.88	23.78	23.74	26.94
GPT-4-Turbo	39.14	42.0	38.38	41.46	40.4	41.06	36.64	42.16	32.83	32.83	27.63	30.07	50.61	56.1	37.95	40.81

of our evaluation mechanism and metrics, the **score rankings of different models stay in relatively stable order across metrics**. In other words, we have not observed any model that excels in determining the solution correctness (thus high in MCC) but is unable to explain the rationale behind it (e.g., low in ACC_reason).

2. **Task Difficulties**: As shown in the breakdown table, the ACC_reason metric is more discriminative than the MCC metric for competent models but vice versa for the less competent ones. This aligns with our intuition that generally more difficult questions are more discriminative for strong candidates, while weaker ones are simply incapable of solving

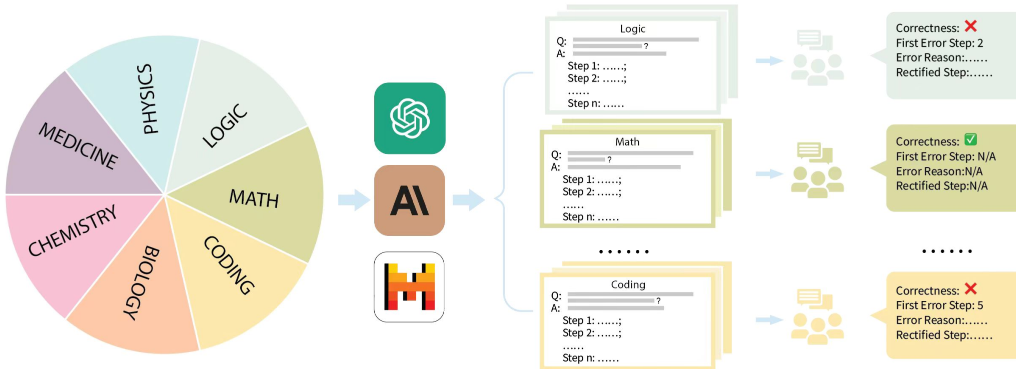

Figure 6: This is the illustration of the dataset creation pipeline of MR-Ben . We first compile a set of questions from different subjects and then collect solutions from different LLMs. For each subject, a group of domain experts is recruited to annotate each question solution pair on its solution correctness, first error step, and error reasons.

them. **This phenomenon could in part explain why in general the MR-Score is not very sensitive to minor changes in the weightings assigned to the subtasks**, since the differentiability of the subtask metrics tends to reconcile with each other under different scenarios.

3. **Differentiability and Interpretability**: The weights of the MR-Score are ultimately decided by considering both the discriminative ability and the interpretability. To best differentiate models with different evaluation results, we conducted a thorough grid search to investigate the impact of the weightings. Since the weightings calculated returned a few optimal instances, we deliberately selected the one that assigns higher scores to more difficult tasks. **We believe the current weighting ratio strikes a good balance between interpretability and differentiation**: For example, GPT-4-Turbo, Deepseek-v2-236B, and Mistral-Large achieve 86.4%, 78.5%, and 81.2% respectively in MMLU but score 43.2%, 29.4%, and 21.3% in our benchmark.

C More Discussion on Biases

To quantitatively assess the relationship between the length of solutions and their correctness, Pearson-Correlation-Coefficients were calculated and reported in Table-8 in the Appendix. The result suggests varying dynamics across disciplines regarding how solution length impacts the likelihood of correctness. For subjects such as coding, chemistry and math, longer solutions are less likely to be correct, which could suggest that complexity or elaboration in responses may lead to mistakes or incorrect reasoning. For medicine, despite being weak, there's a tendency for longer solutions to be slightly more correct, possibly due to more detailed or thorough explanations being favorable. For the other subjects, length of solution does not appear to significantly affect correctness, indicating that other factors likely play a more dominant role in determining solution quality. The overall Pearson Coefficients analysis reflects the distinct nature of problem-solving in each field of our benchmark.

Table 8: Pearson Correlation Between Solution Length and Correctness

Subject	Pearson Correlation	P-value
Medicine	0.094	0.0072
Physics	-0.061	0.111
Biology	0.009	0.783
Chemistry	-0.127	0.00018
Coding	-0.199	0.0021
Logic	0.0002	0.995
Math	-0.115	0.00049

Table 9: Evaluation Results of Models on MR-Bean in few-shot settings: This table presents a detailed breakdown of each model's performance evaluated under metric MR-Score across different subjects.

Model	k-shot	Bio.	Phy.	Math	Chem.	Med.	Logic	Coding	Avg.
Gemma-2B	0	0.1	0.0	0.0	0.1	0.0	0.0	0.7	0.1
	1	0.0	0.0	1.0	0.0	0.4	0.2	0.0	0.2
	2	0.1	0.2	0.7	0.6	0.7	0.2	0.0	**0.4**
	3	0.1	0.3	1.1	0.1	0.7	0.3	0.0	0.4
Llama3-8B	0	11.1	14.9	14.8	12.8	9.4	9.6	9.1	**11.7**
	1	11.7	8.1	7.8	12.8	7.3	10.7	5.7	9.2
	2	9.7	7.8	11.1	8.8	6.4	6.2	2.4	7.5
	3	10.0	10.7	8.3	8.2	5.5	5.3	3.0	7.3
Llama3-70B	0	19.9	15.4	15.0	17.6	14.6	13.5	28.2	17.7
	1	30.5	21.4	16.8	26.2	16.9	16.0	15.3	**20.4**
	2	27.2	19.9	16.8	22.0	15.9	17.5	19.5	19.8
	3	27.2	20.6	16.3	21.1	16.0	14.6	19.4	19.3
GPT-4-Turbo	0	44.7	42.8	44.3	44.0	38.8	34.1	53.6	43.2
	1	47.3	45.2	45.4	46.0	38.4	33.6	57.3	**44.7**
	2	46.6	42.7	44.9	43.3	42.1	35.9	53.0	44.1
	3	44.0	44.8	46.5	44.4	41.2	33.7	56.6	44.5

D Evaluation Prompt

Figure-10 is the prompt template we used to evaluate all the models in our paper. Note that with minor modifications on the following template, the evaluation results can be heavily affected. For example, by introducing a simple hint sentence "Hint: This solution is incorrect. Please focus on looking for the First Error Step and Error Reason.", the model performance can drastically improve as shown in 3. Also, by simply taking away the line of 'Solution Analysis' in the response format part of the prompt, the evaluated model will directly output the scoring result without step-wise COT analysis on the solution. This setup will lead to a near zero MR-Score performance as discussed in Section-5.

Figure-7 is the prompt we used to query language models for solution generation during the dataset compilation phase. Note that in the prompt, we specifically asked the model to analyse each option in the multiple-choice problem. This is crucial in examining if the model possesses a comprehensive understanding on the topics that the question is asking.

Figure-11 shows the prompt we used to query GPT-4 to score the error reasons returned from evaluated models. Despite the challenging nature of the original task to determine the solution correctness, it is a much easier job to determine if the error reason from the evaluated models aligns with the ground truth error reason.

Figure-8 demonstrates the prompt template we used for self-refine experiment. Note that we followed the setting of [31] without introducing any prior assumptions or knowledge. This minimum version of extra prompting would mostly rely on the capability of language models to perform self-refine procedure.

E Self Refine Analysis

In this section, we present the results of self-refine in the task level. Specifically, we are looking at the change of labelling by the evaluated models in the determination of solution correctness as shown by Figure-9. We summarize our observation below:

- **Small Models** like Gemma-2B are too limited to perform effective self-reflection.
- **Competent Models** like GPT4-Turbo are confident in their initial decisions, hardly switching their decisions during self-reflection.
- **Intermediate Models** like Llama3-70B exhibit substantial changes during self-reflection, indicating a lack of consistency in their decisions. However, its change of decisions from incorrect to correct happens to be significantly higher in locating the first error step than

Prompt for Solution Generation During Dataset Compilation

```
Please generate a step-by-step analysis for the following Question in the subject
{subject}.

Question: {df.iloc[idx]['Question']}

Choice_A: {df.iloc[idx]['Choice_A']}
Choice_B: {df.iloc[idx]['Choice_B']}
Choice_C: {df.iloc[idx]['Choice_C']}
Choice_D: {df.iloc[idx]['Choice_D']}

Here is the desired format, please analyse each candidate choice sequentially and
then jointly decide which option is the solution in the final step.
Please ensure every newline character follows a step indicator:
Step 1: [The first reasoning of the step by step analysis on the candidate
choices here]
Step 2: [The second reasoning of the step by step analysis on the candidate
choices here]
...
Step n: [Conclude your analysis and decide which choice to make here]
Solution: Choice_A/B/C/D

Please follow this format without any additional introductory or concluding
statements.
```

Figure 7: This is the prompt we used for solution generation during the dataset compilation stage. Note that besides coding, every subject question in our dataset takes the form of multiple choice problem.

in examining solution correctness and explaining the error reason, therefore boosting the overall MR-Score by a large margin. We believe the lack of consistency does not necessarily indicate a more robust or advanced reasoning ability, despite the increase in the evaluation results.

- **Conclusion**: Our results support the observation that LLMs generally lack effective self-refinement capabilities [31].

F Error Analysis

We provide qualitative analyses of how GPT-4 as an example model performed on our benchmark across all seven subjects. The purpose is to offer a deeper understanding of the types and causes of errors made by experimented models to inform future improvements. For each subject in the subsections below, a failure case and a success case are listed. Following the MR-Ben evaluation framework, each case demonstration consists of the following parts: (1) original questions, options, ground-truth final answers, and LLM-generated CoT solutions; (2) human annotations of step-wise error detection, explanation, and correction; (3) evaluation annotation from the experimented GPT-4 on the aforementioned LLM-generated CoT solutions; (4) scoring results of the error reason if the experimented model identifies the correct first error step.

From our analysis of sampled failure cases, several general observations are made. Firstly, the assessed model GPT-4 exhibits a widely resistant ‘false positive bias’ on our benchmark across all subjects: In cases where the LLM makes incorrect evaluations, the proportion of type I errors is much higher than type II errors. In other words, GPT-4 tends to overlook the mistakes that exist in incorrect model solutions and mislabel them as correct, while seldom actively mislabeling correct model solution steps as incorrect steps. In fact, among the 42 sampled cases we surveyed spanning the seven subjects, all failure cases (size = 21) belong to the type I error category. We provide two possible explanations for such bias: (a) **input bias**: the implemented LLMs are instructed-tuned, and are inherently biased to follow the prompt input. Therefore, even when the models are asked to fairly judge these CoT solution steps in the prompt input in binary labels, it is likely their labeling threshold is affected and biased towards positive judgments. This is a common issue in using LLMs as generation evaluators and may be mitigated by adjusting the prompt design or other debiasing methods [42, 79]; (b) **self-preference bias**: it has drawn recent attention that state-of-the-art models display self-preference bias: the phenomenon in which an LLM inherently favors their own generated output over texts from other LLMs and humans [43, 54]. Therefore, the experiment results of LLMs that are under the same family of the three sampled models (GPT-3.5-Turbo-0125 [50], Claude2 [5], and Mistral-Medium [32]) may be affected. With the increasingly extensive use of self-evaluation and LLM-as-judge methods, we call for future researchers' attention to the potential issue.

Secondly, the MR-Ben benchmark revealed many intricate cases where the assessed model GPT-4 reached a correct final answer through incorrect solution steps, challenging the models' multi-step reasoning capabilities to a greater scale. As shown in the failure cases in math, physics, biology, etc., our benchmark evaluation is able to identify step errors that the sample model made in the solution steps even when its final answer matches the final ground-truth choice. While such step errors can be trivial in terms of generating the correct final answer in the demonstrated failure cases, they can become significant in just slightly nuanced questions, as mentioned in the error analysis section of MMLU [27]. In contrast, our framework, by decomposing the question and model solutions, remains relatively immune to the nuances in question framing. This highlights an important significance of our MR-Ben benchmark in that it is not only elaborate but also robust compared to previous benchmarks.

Lastly, there are subtle nuances of model performance in different reasoning paradigms manifest in the case demonstrations of specific subjects. They are interpreted case by case by the captioned figures listed below.

E.1 Mathematics

Subject: High School Math
Question: What is the smallest whole number that has a remainder of 1 when divided by 4, a remainder of 1 when divided by 3, and a remainder of 2 when divided by 5?
Options: [A.] 67 [B.] 43 [C.] 57 [D.] 37
Ground Truth Answer: D

——————————————- **Sampled CoT** ——————————————

Sampled Model: Claude2
Model Solution Steps:
Step 1: Check Choice A (67) When 67 is divided by 4, the remainder is 3, not 1. So 67 is not the correct answer.
Step 2: Check Choice B (43) When 43 is divided by 4, the remainder is 1. When 43 is divided by 3, the remainder is 1. When 43 is divided by 5, the remainder is 3, not 2. So 43 is not the correct answer.
Step 3: Check Choice C (57) When 57 is divided by 4, the remainder is 1. When 57 is divided by 3, the remainder is 0, not 1. So 57 is not the correct answer.
Step 4: Check Choice D (37) When 37 is divided by 4, the remainder is 1. When 37 is divided by 3, the remainder is 1. When 37 is divided by 5, the remainder is 2.
Step 5: Choice D (37) satisfies all the requirements and is the correct answer.
Solution: Choice_D
Solution: Choice D

——————————————- **Human Annotation** ——————————————

Model Solution Correctness: incorrect
Model Solution First Error Step: Step 2
Model Solution Error Reason:
The model's calculation is incorrect. In step 2, the model incorrectly calculated the remainder of 43 divided by 4 as 1 instead of 3. The model assumes that when 43 is divided by 4, the remainder is 1, but it should be 3. There is a miscalculation of the remainder here.
Model Solution Rectified First Error Step: Step 2: Check option B (43): When 43 is divided by 4, the remainder is 3, not 1. When 43 is divided by 3, the remainder is 1. When 43 is divided by 5, the remainder is 3, not 2. Therefore, 43 is not the correct answer.

——————————————-**Model Evaluation Result**——————————————-

Evaluated Model: gpt-4-turbo
Evaluation Raw Response: Solution Analysis: Step 1: The step checks the remainder when 67 is divided by 4 and finds that it yields a remainder of 3, not 1. This is a correct verification based on the criteria specified in the question.
Step 2: Checks the remainder of 43 when divided by 4, 3, and 5. The findings (1, 1, and 3 respectively) are verified correctly and match the expected remainders except for the one with 5, correctly identifying that the condition is not met for division by 5.
Step 3: The step calculates the remainders for 57 being divided by 4 and 3. The remainder when divided by 3 does not match the specified conditions (expected was 1, result was 0). Hence, the claim that 57 isn't correct is valid.
Step 4 and Step 5: Computes and verifies the remainders for 37 when divided by 4, 3, and 5, matching all the expected remainders accurately, and concludes that 37 meets all conditions specified in the problem.
The answer selection of Choice D for 37 is based on accurate verification that fulfills the problem's constraints.
Solution Correctness: correct
First Error Step: N/A
Error Reason: N/A

——————————————**Error Reason Scoring**——————————————-

Need Error Reason Review: False

Figure E.1-1. A failure case in the math subject. This is a question where the sampled LLM (Claude2) reached a correct answer through incorrect multi-step reasoning. In 'Evaluation Results', the model fails to identify an error in step 2 of model solutions. As noted by the human-annotated 'Model Solution Error Reason', Step 2 checks the remainder when 43 is divided by 4 and incorrectly calculates the remainder as 1 while the correct calculation is 3. The failure case showcases one of the strengths of our benchmark in probing the intricacies of multi-step reasoning by examining process-level errors.

Subject: high school math
Question: Alex needs to borrow $\$10,000$ from the bank. The bank gives him two options. 1. A ten-year loan with an annual interest rate of 10% compounded quarterly, with the condition that at the end of 5 years, Alex must make a payment equal to half of what he owes. The other half continues to accrue interest, and at the end of the ten years, Alex will pay off the remaining balance. 2. A ten-year loan with a simple annual interest rate of 12%, with just one lump-sum payment at the end of the ten years.
Find the positive difference between the total amounts Alex has to pay back under the two schemes. Round your answer to the nearest dollar.
Options: [A.] 16401 [B.] 16400 [C.] 382 [D.] 383
Ground Truth Answer: C
———————————————- **Sampled CoT** ———————————————
Sampled Model: GPT3.5
Model Solution Steps:
Step 1: Calculate the total amount Alex has to pay back under option 1.
Step 2: Calculate the total amount Alex has to pay back under option 2.
Step 3: Find the positive difference between the total amounts Alex has to pay back under the two schemes.
Step 4: Compare the positive difference calculated in step 3 with the given choices.
Step 5: Round the calculated positive difference to the nearest dollar if necessary.
Step 6: Determine which choice matches the calculated positive difference. Solution: Choice A
Solution Correctness: incorrect
Model Solution First Error Step: 6
Model Solution Error Reason: According to the logic provided earlier, the model should give a correct match in the end, but the final answer given is wrong. In this step, the model selected the wrong answer, which could be due to errors in the deduction process that were not listed.
Model Solution Rectified First Error Step: Step 6: Determine which option matches the calculated positive difference. Answer: Option D.
———————————————-**Model Evaluation Result**———————————————-
Evaluated Model: gpt-4-turbo
Evaluation Raw Response:
Solution Analysis:
- Step 1: Neutral - This step claims to calculate the total amount Alex has to pay back under option 1 but does not provide any actual calculations or mathematical process.
- Step 2: Neutral - Similar to step 1, this step lacks any substantial content regarding how calculations are carried out for the repayment under option 2.
- Step 3: Neutral - This step mentions finding the positive difference between the total amounts, though no specific values or calculations are shown.
- Step 4: Neutral - Comparing the positive difference to given choices depends on values obtained in previous steps, which are not provided.
- Step 5: Neutral - The mention of rounding the difference falls in line with the problem's instructions but is non-specific due to lack of previous calculations.
- Step 6: Incorrect - Determining the choice without showing calculations leads to an unsupported conclusion.
Solution Correctness: incorrect
First Error Step: Step 6
Error Reason: The first error step is Step 6 where a decision is made on which choice matches the calculated positive difference without providing any calculations or reasons for the decision. The rectified reasoning should involve calculating the payback amounts for both options explicitly, finding their difference, rounding it if necessary, and then correctly selecting the matching option based on those calculated and verified figures.
Need Error Reason Review: True
———————————————-**Error Reason Scoring**———————————————-
Error Reason Correctness Analysis: - The incorrect solution directly jumps to the final choice without showing any calculations or logical reasoning that leads to that decision. This skips essential steps that would normally validate the conclusion drawn. - The ground truth error reasons indicate that the error lies in the final selection of the answer without considering the calculation errors or logical missteps in previous steps that may have led to an incorrect conclusion. - Essentially, the error is not in the methodology but in the final step's execution, where an incorrect choice is selected without showing the work that would justify this selection.
Error Reason Analysis: - The student correctly identifies that the error occurs in the final step, where a decision is made without showing calculations or reasoning. This observation aligns with the ground truth that the final step incorrectly concludes without apparent justification. - The student emphasizes the need for explicit calculations and a clear demonstration of the reasoning process that leads to the selection of the correct option. This suggestion correctly identifies the lack of shown work and logical deduction in the provided incorrect solution. - The student's explanation focuses on the absence of explicit calculations and reasoned decision-making in the final step, which accurately reflects the actual error in the solution.
Error Reason Correctness: Correct

Figure E.1-2. success case in the math subject. The assessed model identifies the key mistake in Step 4 regarding the faulty equation $f(8) = \frac{1}{2}f(10) = \frac{1}{2}2f(8)$ and points out the incorrect and baseless cancellation leading to $f(8) = f(0)$. Its error reason also correctly suggests that the proper method involves using the transformation properties iteratively in steps of 2 units backward from $f(10)$ to reach $f(0)$.

E.2 Physics

Subject: college_physics
Question: Protons used in cancer therapy are typically accelerated to about $0.6c$. How much work must be done on a particle of mass m in order for it to reach this speed, assuming it starts at rest?
Options: [A.] $0.25mc^2$ [B.] $0.60mc^2$ [C.] $0.67mc^2$ [D.] $1.25mc^2$
Ground Truth Answer: A
—————————— **Sampled CoT** ——————————
Sampled Model: Claude2
Model Solution Steps:
Step 1: The work-energy theorem states that the net work done on an object equals its change in kinetic energy.
Step 2: The kinetic energy of an object with mass m moving at speed v is $\frac{1}{2}mv^2$.
Step 3: Since the proton starts from rest, its initial kinetic energy is 0.
Step 4: The final speed v is given as $0.6c$.
Step 5: Plugging this into the kinetic energy equation gives:

$$\text{Final KE} = \frac{1}{2}m(0.6c)^2$$

Step 6: Setting the change in kinetic energy equal to the net work done:

$$\text{Net work} = \text{Final KE} - \text{Initial KE} = \frac{1}{2}m(0.6c)^2 - 0 = 0.36mc^2$$

Step 7: Of the choices given, $0.36mc^2$ is closest to $0.25mc^2$.
Solution: Choice A
Solution Correctness: incorrect
Model Solution First Error Step: 2
Model Solution Error Reason: This step incorrectly uses the classical kinetic energy formula $KE = \frac{1}{2}mv^2$ to calculate the kinetic energy of a particle moving at speeds close to the speed of light. When the particle's speed approaches the speed of light, the relativistic kinetic energy formula $KE = (\gamma - 1)mc^2$ should be used, where m is the relativistic mass of the particle, m_0 is the rest mass of the particle, and c is the speed of light.
In physics, when the velocity of an object approaches the speed of light, we need to use the principles of special relativity to calculate. In this case, the kinetic energy of the particle is no longer the classical kinetic energy formula but requires the use of the relativistic kinetic energy formula.
Model Solution Rectified First Error Step: Step 2: The relativistic kinetic energy of an object with mass m and velocity v is K.

Step 2: The kinetic energy of an object with mass m moving at velocity v is $mc^2\left(\frac{1}{\sqrt{1-\frac{v^2}{c^2}}} - 1\right)$.
————————**Model Evaluation Result**————————
Evaluated Model: gpt-4-turbo
Evaluation Raw Response:
Solution Analysis:
- Step 1: Correct. The work-energy theorem properly states that the net work done on an object is equal to its change in kinetic energy.
- Step 2: Incorrect. This is where the solution makes a fundamental error by using the classical formula for kinetic energy which isn't valid at significant fractions of the speed of light (like $0.6c$).
- Step 3: Correct. The initial kinetic energy being zero since the particle starts from rest is a valid statement.
- Step 4: Correct. The final speed of the proton is appropriately given as $0.6c$.
- Step 5: Incorrect as a consequence of Step 2, applying the incorrect equation.
- Step 6: Incorrect as it follows from the incorrect application of the kinetic energy formula and ends up with a wrong expression for net work done.
- Step 7: Incorrect reasoning in comparing the wrongly calculated work with the answer choices, leading to the wrong selection.

Solution Correctness: incorrect First Error Step: Step 2 Error Reason: Step 2 uses the non-relativistic formula for kinetic energy, $\frac{1}{2}mv^2$, which is not accurate for speeds approaching the speed of light (such as $0.6c$). The relativistic kinetic energy formula should be used instead, which is $KE = (\gamma - 1)mc^2$ where γ is the Lorentz factor, $\gamma = \frac{1}{\sqrt{1-\frac{v^2}{c^2}}}$. Using this formula correctly accounts for the relativistic effects significant at high velocities.
Solution Correctness: incorrect
First Error Step: Step 2
Error Reason: Step 2 uses the non-relativistic formula for kinetic energy, $\frac{1}{2}mv^2$, which is not accurate for speeds approaching the speed of light (such as $0.6c$). The relativistic kinetic energy formula should be used instead, which is $KE = (\gamma - 1)mc^2$ where γ is the Lorentz factor, $\gamma = \frac{1}{\sqrt{1-\frac{v^2}{c^2}}}$. Using this formula correctly accounts for the relativistic effects significant at high velocities.
Need Error Reason Review: True
Error Reason Correctness Analysis: - At speeds approaching that of light, relativistic effects become significant. Thus, the classical mechanics approach is no longer accurate for describing the physics of the situation. - The correct formula to calculate kinetic energy at relativistic speeds includes the Lorentz factor, γ, which accounts for the effects of time dilation and length contraction at high velocities. - The relativistic kinetic energy (KE) formula is $KE = (\gamma - 1)m_0c^2$, where m_0 is the rest mass of the particle, c is the speed of light, and γ is the Lorentz factor, defined as $\gamma = \frac{1}{\sqrt{1-\frac{v^2}{c^2}}}$. - This formula diverges from the classical $KE = \frac{1}{2}mv^2$ when v becomes a significant fraction of c. - Therefore, the incorrect step in the solution is using the non-relativistic kinetic energy formula, which fails to account for the relativistic effects that are significant at a velocity of $0.6c$.
Error Reason Analysis: - The student correctly identifies that the non-relativistic kinetic energy formula is inaccurate for speeds approaching the speed of light. - The student accurately notes the requirement to use the relativistic kinetic energy formula, which includes the Lorentz factor (γ). - The student provides the correct relativistic kinetic energy formula $KE = (\gamma - 1)mc^2$ and describes γ correctly as $\gamma = \frac{1}{\sqrt{1-\frac{v^2}{c^2}}}$. - Overall, the student's explanation properly reflects the nature of the error in the solution and identifies the appropriate relativistic formula to use.
Error Reason Correctness: Correct

Figure E.2-1. A failure case in the physics subject. It is another occurrence where the sampled LLM (Claude2) reached a correct answer even though its model solution contains an erroneous step. Physics formulas require strictly case-sensitive expressions to represent different concepts, as noted in 'Model Solution Error Reason', which models did not differentiate, thus making mistakes in explaining the relationship between the orbital period T and the satellite mass m.

Subject: college_physics
Question: Protons used in cancer therapy are typically accelerated to about $0.6c$. How much work must be done on a particle of mass m in order for it to reach this speed, assuming it starts at rest?
Options: [A.] $0.25mc^2$ [B.] $0.60mc^2$ [C.] $0.67mc^2$ [D.] $1.25mc^2$
Ground Truth Answer: A

——————————— **Sampled CoT** ———————————

Sampled Model: Claude2
Model Solution Steps:
Step 1: The work-energy theorem states that the net work done on an object equals its change in kinetic energy.
Step 2: The kinetic energy of an object with mass m moving at speed v is $\frac{1}{2}mv^2$.
Step 3: Since the proton starts from rest, its initial kinetic energy is 0.
Step 4: The final speed v is given as $0.6c$.
Step 5: Plugging this into the kinetic energy equation gives:

$$\text{Final KE} = \frac{1}{2}m(0.6c)^2$$

Step 6: Setting the change in kinetic energy equal to the net work done:

$$\text{Net work} = \text{Final KE} - \text{Initial KE} = \frac{1}{2}m(0.6c)^2 - 0 = 0.36mc^2$$

Step 7: Of the choices given, $0.36mc^2$ is closest to $0.25mc^2$. Solution: Choice A
Solution Correctness: incorrect
Model Solution First Error Step: 2
Model Solution Error Reason: This step incorrectly uses the classical kinetic energy formula $KE = \frac{1}{2}mv^2$ to calculate the kinetic energy of a particle moving at speeds close to the speed of light. When the particle's speed approaches the speed of light, the relativistic kinetic energy formula $KE = (\gamma - 1)mc^2$ should be used, where m is the relativistic mass of the particle, m_0 is the rest mass of the particle, and c is the speed of light.
Model Solution Rectified First Error Step: Step 2: The kinetic energy of an object with mass m moving at velocity v is $mc^2\left(\frac{1}{\sqrt{1-\frac{v^2}{c^2}}} - 1\right)$.

——————————**Model Evaluation Result**——————————

Evaluated Model: gpt-4-turbo
Evaluation Raw Response:
Solution Analysis:
- Step 1: Correct. The work-energy theorem properly states that the net work done on an object is equal to its change in kinetic energy.
- Step 2: Incorrect. This is where the solution makes a fundamental error by using the classical formula for kinetic energy which isn't valid at significant fractions of the speed of light (like $0.6c$).
- Step 3: Correct. The initial kinetic energy being zero since the particle starts from rest is a valid statement.
- Step 4: Correct. The final speed of the proton is appropriately given as $0.6c$.
- Step 5: Incorrect as a consequence of Step 2, applying the incorrect equation.
- Step 6: Incorrect as it follows from the incorrect application of the kinetic energy formula and ends up with a wrong expression for net work done.
- Step 7: Incorrect reasoning in comparing the wrongly calculated work with the answer choices, leading to the wrong selection.

Solution Correctness: incorrect First Error Step: Step 2 Error Reason: Step 2 uses the non-relativistic formula for kinetic energy, $\frac{1}{2}mv^2$, which is not accurate for speeds approaching the speed of light (such as $0.6c$). The relativistic kinetic energy formula should be used instead, which is $KE = (\gamma - 1)mc^2$ where γ is the Lorentz factor, $\gamma = \frac{1}{\sqrt{1-\frac{v^2}{c^2}}}$. Using this formula correctly accounts for the relativistic effects significant at high velocities.
Solution Correctness: incorrect
First Error Step: Step 2
Error Reason: Step 2 uses the non-relativistic formula for kinetic energy, $\frac{1}{2}mv^2$, which is not accurate for speeds approaching the speed of light (such as $0.6c$). The relativistic kinetic energy formula should be used instead, which is $KE = (\gamma - 1)mc^2$ where γ is the Lorentz factor, $\gamma = \frac{1}{\sqrt{1-\frac{v^2}{c^2}}}$. Using this formula correctly accounts for the relativistic effects significant at high velocities.
Need Error Reason Review: True
Error Reason Correctness Analysis: - At speeds approaching that of light, relativistic effects become significant. Thus, the classical mechanics approach is no longer accurate for describing the physics of the situation. - The correct formula to calculate kinetic energy at relativistic speeds includes the Lorentz factor, γ, which accounts for the effects of time dilation and length contraction at high velocities. - The relativistic kinetic energy (KE) formula is $KE = (\gamma - 1)m_0c^2$, where m_0 is the rest mass of the particle, c is the speed of light, and γ is the Lorentz factor, defined as $\gamma = \frac{1}{\sqrt{1-\frac{v^2}{c^2}}}$. - This formula diverges from the classical $KE = \frac{1}{2}mv^2$ when v becomes a significant fraction of c. - Therefore, the incorrect step in the solution is using the non-relativistic kinetic energy formula, which fails to account for the relativistic effects that are significant at a velocity of $0.6c$.
Error Reason Analysis: - The student correctly identifies that the non-relativistic kinetic energy formula is inaccurate for speeds approaching the speed of light. - The student accurately notes the requirement to use the relativistic kinetic energy formula, which includes the Lorentz factor (γ). - The student provides the correct relativistic kinetic energy formula $KE = (\gamma - 1)mc^2$ and describes γ correctly as $\gamma = \frac{1}{\sqrt{1-\frac{v^2}{c^2}}}$. - Overall, the student's explanation properly reflects the nature of the error in the solution and identifies the appropriate relativistic formula to use.
Error Reason Correctness: Correct

Figure E.2-2. A success case in the physics subject. The evaluator GPT-4 correctly identified the first error step and provided appropriate and rational reasons.

E.3 Biology

Subject: college_biology
Question: Which of the following statements most detracts from the claim that the human appendix is a completely vestigial organ?
Options:
-[A.] The appendix can be surgically removed with no immediate ill effects.
-[B.] The appendix might have been larger in fossil hominids.
-[C.] The appendix has a substantial amount of defensive lymphatic tissue.
-[D.] Individuals with a larger-than-average appendix leave fewer offspring than those with a below-average-sized appendix.
Ground Truth Answer: C
—————————— **Sampled CoT** ——————————
Sampled Model: Claude2
Model Solution Steps:
Step 1: Choice A states that the appendix can be surgically removed without ill effects. This suggests the appendix may not serve an important function, supporting the claim that it is vestigial.
Step 2: Choice B states the appendix may have been larger in fossil hominids. This suggests the appendix may have served a more important function in our evolutionary ancestors, weakening the claim that it is vestigial.
Step 3: Choice C states the appendix has substantial lymphatic tissue. This suggests the appendix may play an immunological role, weakening the claim that it is completely vestigial.
Step 4: Choice D relates appendix size to reproductive fitness. This does not directly address whether the appendix is vestigial.
Step 5: Choice C provides the strongest evidence against the appendix being completely vestigial, by suggesting it has an immunological function.
Solution: Choice C
Solution Correctness: incorrect
Model Solution First Error Step: 2
Model Solution Error Reason: Step 2 incorrectly assumes that the fact that the appendix might have been larger in fossil hominids weakens the claim that the appendix is a vestigial organ. If the appendix was larger in fossil hominids, it might indicate that it had some function in the past, but now these functions have diminished or become unimportant, which supports the claim that the appendix is a vestigial organ. This option only indicates the function of the appendix in fossil hominids and cannot strongly prove the function of the appendix in modern humans.
Model Solution Rectified First Error Step: Step 2: Option B indicates that the appendix might have been larger in fossil hominids. This suggests that the appendix might have had more important functions in our evolutionary ancestors, further proving the claim that it is a vestigial organ.
Step 2: Option B indicates that the appendix might have been larger in fossil hominids. However, this does not explain the function of the appendix in modern humans, so this option has limitations.
——————————**Model Evaluation Result**——————————
Evaluated Model: gpt-4-turbo
Evaluation Raw Response:
Solution Analysis: - Step 1: This step is correct. It accurately identifies that the ability to remove the appendix without immediate ill effects supports the idea that the appendix might be vestigial.
- Step 2: This step is also correct. It suggests that if the appendix was larger in the past, it may have had a more significant function in fossil hominids, which provides a basis for it not being entirely vestigial in humans today.
- Step 3: This step is correct. It presents the argument that the appendix's immunological function suggests it may serve a purpose and is not completely vestigial.
- Step 4: This step is neutral. It accurately states that Option D does not directly address whether the appendix is vestigial, which doesn't detract from nor support the idea of it being vestigial.
- Step 5: This step is correct in concluding that considering the provided options, the immunological role indicated by C (substantial amount of lymphatic tissue) provides the strongest evidence against the appendix being completely vestigial.
Solution Correctness: correct
First Error Step: N/A
Error Reason: N/A
Need Error Reason Review: False

Figure E.3-1. A failure case of the biology subject. It is another occurrence where the sampled LLM (Claude2) reached a correct answer through incorrect multi-step reasoning. The evaluator GPT-4 did not spot the knowledge and logic reasoning error in step 2 about Choice B, but made a similar incorrect analysis.

Subject: college_biology
Question: Natural selection is based on all of the following except **Options:** [A.] genetic variation exists within populations. [B.] the best-adapted individuals tend to leave the most offspring. [C.] individuals who survive longer tend to leave more offspring than those who die young. [D.] individuals adapt to their environments and, thereby, evolve. **Ground Truth Answer:** D

——————————— **Sampled CoT** ———————————

Sampled Model: GPT3.5
Model Solution Steps:
Step 1: Genetic variation within populations is a fundamental concept of natural selection as it provides the raw material for evolution to occur.
Step 2: The best-adapted individuals leaving the most offspring is a key aspect of natural selection as it leads to the passing on of advantageous traits to subsequent generations.
Step 3: Individuals who survive longer tend to leave more offspring than those who die young is directly related to the concept of natural selection, as individuals with advantageous traits are more likely to survive and reproduce.
Step 4: Individuals adapting to their environments and evolving is the core principle of natural selection, as organisms that are better suited to their environment are more likely to survive and reproduce. Solution: Choice A: genetic variation exists within populations.
Solution Correctness: incorrect
Model Solution First Error Step: 4
Model Solution Error Reason: The model that describes individuals adapting to their environments and evolving is incorrect as a core principle of natural selection, because it is populations, not individuals, that are the basic units of evolution. The result of natural selection leads to species adapting to environments and evolving, but it is not based on the factor that individuals adapt to their environments and thereby evolve.
Model Solution Rectified First Error Step: Step 4: Individuals adapting to their environments and evolving is not a core principle of natural selection, because individuals adapting to environments to increase their chances of survival and reproduction cannot be termed as evolution. Solution: Option D: Individuals adapt to their environments and thereby evolve. Step 4: Individuals adapting to their environments and evolving is a result of natural selection, Solution: Option D.

——————————**Model Evaluation Result**——————————

Evaluated Model: gpt-4-turbo
Evaluation Raw Response:
Solution Analysis: - Step 1: This step is correct as it accurately identifies that genetic variation within populations is essential for natural selection because it provides the necessary diversity on which selection can act. - Step 2: This step is correct and highlights that the best-adapted individuals generally reproduce more, which is fundamental to natural selection because it ensures the spread of beneficial traits. - Step 3: This step is correct, establishing that individuals who live longer have more opportunities to reproduce, contributing to the mechanism of natural selection through differential survival. - Step 4: This step is incorrect. It misstates the process of natural selection by suggesting that individuals adapt to their environments and thereby evolve. Natural selection acts on existing variations within a population and doesn't involve organisms individually adapting or evolving in response to an environment.
Solution Correctness: incorrect First Error Step: Step 4 Error Reason: Step 4 incorrectly implies that individual organisms adapt to their environments and evolve as a result, which misunderstands natural selection. In reality, natural selection operates on a population level based on existing genetic variations, and individuals do not adapt or evolve during their lifetimes. Evolutionary changes can occur in the population over generations as those with beneficial traits have better survival and reproductive success. The corrected reasoning for this step would be: "The misconception that individuals adapt and thereby evolve is not a principle of natural selection, as natural selection acts on heritable traits that are beneficial prior to and independently from environmental pressures, not post-adaptation by individuals."
Solution Correctness: incorrect
First Error Step: Step 4
Error Reason: Step 4 incorrectly implies that individual organisms adapt to their environments and evolve as a result, which misunderstands natural selection. In reality, natural selection operates on a population level based on existing genetic variations, and individuals do not adapt or evolve during their lifetimes. Evolutionary changes can occur in the population over generations as those with beneficial traits have better survival and reproductive success. The corrected reasoning for this step would be: "The misconception that individuals adapt and thereby evolve is not a principle of natural selection, as natural selection acts on heritable traits that are beneficial prior to and independently from environmental pressures, not post-adaptation by individuals."
Need Error Reason Review: True
Error Reason Correctness Analysis: 1. Step-by-Step Reasoning: - The original incorrect step implies that individual organisms are capable of adapting and evolving within their own lifetimes as a response to environmental pressures. This is a fundamental misunderstanding of how natural selection works. - Natural selection operates at the level of populations over generations, not individuals. It is the differential survival and reproduction of individuals with certain heritable traits that leads to evolution. - The correct understanding is that while individuals with certain traits may have better survival and reproduction rates, these traits must already exist within the genetic makeup of the population. The population evolves over time as these traits become more common, not individuals adapting or evolving on their own.
2. Student Error Reason Analysis: - The student identifies the core issue with the incorrect step, noting that it incorrectly attributes the process of adaptation and evolution to individual organisms rather than populations. - The student correctly points out that natural selection is based on pre-existing genetic variation within a population and that individuals themselves do not adapt or evolve during their lifetimes. - The student's explanation correctly aligns with the principle that evolutionary changes are a consequence of differential reproductive success among individuals with different heritable traits, and these changes accumulate over generations within a population. - The student accurately clarifies the misconception by explaining that natural selection acts on heritable traits beneficial prior to any environmental changes and that the process does not involve post-adaptation evolution by individuals.
3. Final Decision: Correct

Figure E.3-2. A success case of the biology subject.

E.4 Medicine

Question UUID: 997088bb-200d-44b5-9833-a47b29e9a88b
Subject: college_medicine
Question:
A 63-year-old woman is in the hospital recovery room 4 hours after elective left hemicolectomy for colon carcinoma at the splenic flexure. She has a preoperative written directive for no blood products through transfusion for religious reasons. Medical history is significant for hypertension and coronary artery disease. The nurse tells you the patient's blood pressure has progressively declined since the operation. Vital signs now are temperature 35.8°C (96.4°F), pulse 130/min, respirations 20/min, and blood pressure 80/50 mm Hg. Physical examination discloses a slightly distended abdomen with an intact incision. ECG shows sinus tachycardia. Urine output has been 10 mL during the past 2 hours. Hematocrit is 30%; preoperative hematocrit was 41%. The patient has received 4 L of intravenous crystalloid in the recovery room. Reported operative blood loss was 200 mL. Drainage from the nasogastric tube is clear. Damage to which of the following structures is most likely responsible for these findings?
Options:
A: Epigastric artery
B: Liver
C: Middle colic artery
D: Spleen
Ground Truth Answer: D
Ground Truth Answer Modified: C
——————————— **Sampled CoT** ———————————
Sampled Model: mistral-medium
Model Solution Steps:
Step 1: Assess the vital signs and hematocrit
The patient's vital signs show hypotension (low blood pressure), tachycardia (rapid heart rate), and hypothermia (low body temperature). The urine output is also low, indicating possible renal hypoperfusion. The significant drop in hematocrit (from 41% to 30%) suggests acute blood loss anemia.
Step 2: Evaluate the intraoperative and postoperative fluid management
The reported operative blood loss is only 200 mL, which is not enough to cause such a significant drop in hematocrit and hemodynamic instability. The patient received 4 L of intravenous crystalloids in the recovery room. However, this amount of fluid should have compensated for the blood loss and maintained blood pressure stability unless there is ongoing or occult blood loss.
Step 3: Assess the abdominal examination for signs of internal bleeding
The slightly distended abdomen and clear drainage from the nasogastric tube do not suggest an acute mechanical obstruction or ileus. The intact incision and absence of overt bleeding also do not indicate an external source of hemorrhage.
Step 4: Consider possible injury to abdominal structures during the operation
Since the patient had a left hemicolectomy, the potential structures that might have been injured include the epigastric artery, middle colic artery, spleen, or liver.
Step 5: Analyze each choice to determine the most likely cause of the patient's condition
Choice_A: Epigastric artery - Injury to the epigastric artery is unlikely to cause significant bleeding during a left hemicolectomy, as these vessels are not in the immediate operative field.
Choice_B: Liver - Although liver injury can lead to hemorrhage, it would typically present with an acute onset of hypotension and peritoneal signs. The patient's initial vital signs were stable, and there is no evidence of peritonitis.
Choice_C: Middle colic artery - This artery is in the operative field and could be injured during a left hemicolectomy. Major bleeding from this artery could lead to the patient's hemodynamic instability and hematocrit drop.
Choice_D: Spleen - Splenic injury could lead to significant hemorrhage, but this is an uncommon complication of a left hemicolectomy due to the spleen's anatomic location.
Step 6: Integrate all findings and determine the most likely cause
The patient's hemodynamic instability, low hematocrit, and minimal reported blood loss suggest ongoing or occult internal bleeding. The intact incision, clear nasogastric tube drainage, and slightly distended abdomen do not support external bleeding or mechanical obstruction. Among the possible injured structures, the middle colic artery is the most likely cause of the patient's condition.
Solution: Choice_C: Middle colic artery
Solution Correctness: incorrect
Model Solution First Error Step: 2
Model Solution Error Reason:
The analysis mentions that the patient has received 4 liters of intravenous crystalloid in the recovery room, yet the blood pressure continues to decline, indicating ongoing or occult blood loss. The decrease in hematocrit, which might be due to factors other than blood loss such as hemodilution, has not been fully considered.
When the patient woke up from the surgery and was transferred to the recovery room, the doctor had

already administered 4 liters of crystalloid solution intravenously, which is equivalent to a significant hydration. However, despite this hydration, the patient's blood pressure has not returned to normal and continues to drop. This could mean that there is ongoing or stealthy blood loss inside the patient's body, causing the blood pressure to decrease.

Model Solution Rectified First Error Step:

Step 2: The patient has received 4 liters of intravenous crystalloid in the recovery room, but the blood pressure continues to decline. The reported operative blood loss was 200 mL, and the hematocrit dropped from 41% to 30%, which may be partly due to the hemodilution effect of the intravenous injection. Considering the patient's hemodynamic status, the possibility of ongoing or occult internal bleeding cannot be ruled out.

Step 2: After being transferred to the recovery room, the medical team administered 4 liters of crystalloid solution intravenously, which is equivalent to a significant amount of hydration. However, despite this, the patient's blood pressure continues to decline. The reported blood loss during the surgery was 200 mL, which usually would not cause a drop in blood pressure. Also, the patient's hematocrit dropped from 41% to 30%, which may be partly because the infused fluid diluted the blood.

——————————**Model Evaluation Result**——————————

Evaluated Model: gpt-4-turbo

Evaluation Raw Response:

Solution Analysis:

Step 1: Assess the vital signs and hematocrit - Correct. This step accurately summarizes vital signs and hematocrit values suggesting hypotension, tachycardia, hypothermia, and potential acute blood loss.

Step 2: Evaluate the intraoperative and postoperative fluid management - Correct. Observations are correct in stating the discrepancy between reported blood loss and observed symptoms, raising questions about additional or occult bleeding.

Step 3: Assess the abdominal examination for signs of internal bleeding - Correct. The abdominal examination findings are evaluated logically to rule out external bleeding sources and acute mechanical obstructions.

Step 4: Consider possible injury to abdominal structures during the operation - Correct. Considering potential injuries to relevant abdominal structures based on the type of surgery performed is a logical step.

Step 5: Analyze each choice to determine the most likely cause - Correct. Each option is analyzed based on anatomical relevance and likelihood of contributing to the observed clinical picture, leading to a rational elimination of less likely causes.

Step 6: Integrate all findings and determine the most likely cause - Correct. This step synthesizes the information from previous steps and identifies the middle colic artery as the most likely source of ongoing internal bleeding, which is consistent with the location of the surgery and the clinical symptoms.

Solution Correctness: correct

First Error Step: N/A

Error Reason: N/A

Solution Correctness: correct

First Error Step: N/A

Error Reason: N/A

Need Error Reason Review: False

Figure E.4-1. A failure case in the medicine subject. The question is relatively lengthy, highlighting the potential importance of multi-step reasoning for decomposition and analysis. Before diving into the comparison of four options, the sample model solution first provided the initial four steps to assess important information in the question context (e.g., patient status and treatments). The model solution contains an error in step 2 analyzing the patient's fluid management which the evaluator GPT-4 fails to identify.

Subject: college_medicine
Question:
Which of the following diagnostic techniques is of no value for the diagnosis of neural tube defects?
Options:
A: Amniocentesis
B: Chorion villus sampling (CVS)
C: Maternal serum screening
D: Ultrasonography
Ground Truth Answer: B
Ground Truth Answer Modified: B
——————————— **Sampled CoT** ———————————
Sampled Model: GPT3.5
Model Solution Steps:
Step 1: Amniocentesis involves analyzing the amniotic fluid surrounding the developing fetus for genetic abnormalities. It can detect neural tube defects and is commonly used for prenatal diagnosis.
Step 2: Chorion villus sampling (CVS) is another prenatal diagnostic technique that involves taking a sample of the placental tissue to analyze the genetic composition of the fetus. It can also detect neural tube defects.
Step 3: Maternal serum screening is a blood test that measures certain substances in the mother's blood to assess the risk of certain birth defects, including neural tube defects. It is a valuable tool for screening but may not definitively diagnose neural tube defects.
Step 4: Ultrasonography is a prenatal imaging technique that can visualize the developing fetus in the uterus. It can detect neural tube defects by visualizing the spine and brain of the fetus.
Solution: Choice_C - Maternal serum screening is of no value for the diagnosis of neural tube defects.
Solution Correctness: incorrect
Model Solution First Error Step: 2
Model Solution Error Reason:
The model incorrectly assumes that Chorionic Villus Sampling (CVS) can detect neural tube defects. CVS is primarily used for the detection of chromosomal abnormalities, not for diagnosing structural birth defects like neural tube defects.
Chorionic Villus Sampling is a medical procedure that involves analyzing fetal cells in placental tissue to detect chromosomal abnormalities and genetic diseases. Although this technique can provide important information about the genetic condition of the fetus, its primary application does not include the direct diagnosis of neural tube defects. Therefore, describing the main purpose of CVS as directly used for diagnosing neural tube defects is inaccurate.
Model Solution Rectified First Error Step:
Step 2: Chorionic Villus Sampling (CVS) is a prenatal diagnostic technique primarily used for detecting chromosomal abnormalities, such as Down syndrome, by analyzing placental tissue. It is not typically used to detect neural tube defects.
Step 2: Chorionic Villus Sampling (CVS) involves collecting placental tissue and analyzing fetal cells, and is not directly used for diagnosing neural tube defects.
————————**Model Evaluation Result**————————
Evaluated Model: gpt-4-turbo
Evaluation Raw Response:
Solution Analysis:
Step 1: Correct - This explanation accurately describes the process and utility of amniocentesis in detecting genetic abnormalities, including neural tube defects.
Step 2: Incorrect - Chorion villus sampling (CVS) primarily looks at chromosomal information and is more effective in diagnosing chromosomal abnormalities rather than neural tube defects, which are best detected using other methods like amniocentesis, maternal serum screening, or ultrasonography.
Step 3: Correct - The description of maternal serum screening is accurate. While the test is valuable for screening, it is not definitive for diagnosis, hence its utility in assessing risk rather than confirming neural tube defects.
Step 4: Correct - Ultrasonography is correctly described as a visualization method capable of detecting neural tube defects through imaging.
Solution: Incorrect - The conclusion that maternal serum screening is of no value for the diagnosis of neural tube defects is incorrect. Though it is not definitive for diagnosis, it is still a valuable screening tool and does provide relevant information to suggest the presence of neural tube defects.
Solution Correctness: incorrect
First Error Step: Step 2
Error Reason: The error in Step 2 pertains to the oversimplified or wrong explanation regarding the capability of CVS to detect neural tube defects. CVS is rather specific for detecting genetic and chromosomal abnormalities rather than structural defects like neural tube issues which are better detected by amniocentesis or ultrasonography. A rectified response could state that while CVS is valuable for chromosomal analysis, it does not typically detect neural tube defects, which are better assessed by other diagnostic methods outlined here.
Solution Correctness: incorrect
First Error Step: Step 2
Error Reason: The error in Step 2 pertains to the oversimplified or wrong explanation regarding the capability of CVS to detect neural tube defects. CVS is rather specific for detecting genetic and chromosomal abnormalities rather than structural defects like neural tube issues which are better detected by amniocentesis or ultrasonography. A rectified response could state that while CVS is valuable for chromosomal analysis, it does not typically detect neural tube defects, which are better assessed by other diagnostic methods outlined here.
Need Error Reason Review: True
Error Reason Correctness: Correct

Figure E.4-2. A success case of the medicine subject. The evaluator GPT-4 correctly identified the first error step and provided appropriate and rational reasons.

E.5 Chemistry

Subject: college_chemistry
Question:
Calculate the magnetic moment (μ_I) of a ^{13}C nucleus.
Options:
A: 6.1445×10^{-27} J T^{-1}
B: 3.1445×10^{-27} J T^{-1}
C: 9.1445×10^{-27} J T^{-1}
D: 1.1445×10^{-28} J T^{-1}
Ground Truth Answer: C
Ground Truth Answer Modified: A
———————————— **Sampled CoT** ————————————
Sampled Model: Claude2
Model Solution Steps:
Step 1: The magnetic moment (μ_I) of a nucleus depends on its spin quantum number (I) according to the following formula:
$\mu_I = \gamma I \hbar$
Where γ is the gyromagnetic ratio and $\hbar$ is the reduced Planck constant.
Step 2: For a ^{13}C nucleus, $I = \frac{1}{2}$.
Step 3: The gyromagnetic ratio for a ^{13}C nucleus is 6.728×10^7 T^{-1}s^{-1}.
Step 4: Substituting the values for I and γ into the formula:
$\mu_I = (6.728 \times 10^7 \text{ T}^{-1}\text{s}^{-1})(\frac{1}{2})\hbar$
Step 5: The value of the reduced Planck constant $\hbar$ is 1.05457×10^{-34} J·s.
Step 6: Performing the calculation:
$\mu_I = (6.728 \times 10^7 \text{ T}^{-1}\text{s}^{-1})(\frac{1}{2})(1.05457 \times 10^{-34} \text{ J·s})$
$= 3.1445 \times 10^{-27}$ J T^{-1}
Step 7: Comparing to the answer choices, the magnetic moment calculated matches Choice B.
Solution: Choice B
Solution Correctness: incorrect
Model Solution First Error Step: 1
Model Solution Error Reason:
The formula given in the steps is incorrect, the correct formula should be $\mu_I = \gamma L \hbar$, where $L = (I(I+1))^{1/2}$.
The formula provided in the steps is incorrect, the correct expression should be $\mu_I = \gamma L \hbar$, where $L = I(I+1)$.
Model Solution Rectified First Error Step:
Step 1: The magnetic moment (μ_I) of the nucleus is determined by the following formula based on its spin quantum number (I): $\mu_I = \gamma L \hbar$, where $L = (I(I+1))^{1/2}$.
Step 1: The nuclear magnetic moment (μ_I) can be calculated using the following formula based on its spin quantum number (I): $\mu_I = \gamma L \hbar$, where L represents $L = I(I+1)$.
————————————**Model Evaluation Result**————————————
Evaluated Model: gpt-4-turbo-2024-04-09
Evaluation Raw Response:
Solution Analysis:
- Step 1: This step is correct. It states the relationship between the nuclear magnetic moment (μ_I), the gyromagnetic ratio (γ), and the reduced Planck constant ($\hbar$).
- Step 2: This step is correct. The nuclear spin quantum number (I) of carbon-13 (^{13}C) is indeed $\frac{1}{2}$.
- Step 3: This step is correct. The stated gyromagnetic ratio for a ^{13}C nucleus (6.728×10^7 T^{-1}s^{-1}) is the correct value for this isotope.
- Step 4: This step is correct. The given formula $\mu_I = \gamma I \hbar$ is being correctly used to substitute in the values of the gyromagnetic ratio (γ) and the spin quantum number (I).
- Step 5: This step is correct. The provided value of the reduced Planck constant ($\hbar = 1.05457 \times 10^{-34}$ J·s) is accurate.
- Step 6: This step is correct. The calculation of the magnetic moment (μ_I) is correctly performed using the provided values and the formula from step 4.
- Step 7: This step is correct. The calculated result from step 6 is accurately compared with the options given in the question and matches choice B.
Solution Correctness: correct
First Error Step: N/A
Error Reason: N/A
Solution Correctness: correct
First Error Step: N/A
Error Reason: N/A
Need Error Reason Review: False

Figure E.5-1. A failure case in the chemistry subject.

Subject: high_school_chemistry
Question:
Three 25.00 mL samples of approximately 0.10 M phenol, C_6H_5OH, $K_a = 1.0 \times 10^{-10}$ were removed from a container and placed in separate 250 mL beakers. The samples were titrated with standard potassium hydroxide, KOH, solution. Cresol red was the acid-base indicator used in the titration. The samples required 31.75, 32.38, and 41.75 mL to reach the endpoint. Which of the following might explain why one of the samples required significantly more base to reach the endpoint?
Options:
A: The indicator was added too late.
B: The wrong indicator was used.
C: There was a base contaminating the unclean beaker.
D: There was an acid contaminating the unclean beaker.
Ground Truth Answer: D
Ground Truth Answer Modified: D
—————————— **Sampled CoT** ——————————
Sampled Model: Claude2
Model Solution Steps:
Step 1: The phenol solutions were approximately 0.10 M concentration. At this concentration, the volumes of KOH required to reach the endpoint should be similar for all 3 samples.
Step 2: Two of the samples required similar volumes of KOH (31.75 mL and 32.38 mL). However, the third sample required significantly more (41.75 mL).
Step 3: Choice A suggests the indicator was added too late. This would not change the volume of KOH required, only the observed color change.
Step 4: Choice B suggests the wrong indicator was used. This may affect the observed color change, but not the volume of KOH required.
Step 5: Choice C suggests a contaminating base in one sample. A contaminating base would react with the phenol, requiring more KOH to reach the endpoint.
Step 6: Choice D suggests a contaminating acid. An acid contaminant would be neutralized by the KOH and not affect the volume required.
Step 7: Therefore, the explanation that a contaminating base required more KOH to reach the endpoint in one sample is choice C.
Solution: Choice_C
Solution Correctness: incorrect
Model Solution First Error Step: 5
Model Solution Error Reason:
The statement in the erroneous step that more KOH is needed to reach the endpoint is inaccurate. The reaction of phenol with a basic contaminant would lead to a decrease in the amount of KOH used, not an increase.
In the description of the erroneous step, the statement that more KOH is needed to reach the endpoint is inaccurate. In fact, the reaction of phenol with a basic contaminant would result in a decrease in the consumption of KOH, not an increase.
Model Solution Rectified First Error Step:
Step 5: Option C indicates that there is a contaminated base in one of the samples. The contaminated base reacts with phenol, reducing the amount of KOH needed to reach the endpoint.
Step 5: Option C points out that the sample contains a contaminated base. This contaminated base reacts with phenol, leading to a reduction in the amount of KOH needed to reach the endpoint.
——————————**Model Evaluation Result**——————————
Evaluated Model: gpt-4-turbo-2024-04-09
Evaluation Raw Response:
Solution Analysis:
- Step 1 is correct. It establishes the expectation for the volumes of KOH required based on the concentration of phenol.
- Step 2 is correct. It points out the observed discrepancy in the volumes of KOH required, which prompts the question.
- Step 3 is correct. Adding the indicator too late would not alter the required volume of KOH, only the point at which the change is observed.
- Step 4 is correct. Using the wrong indicator may affect the endpoint's observation but not the actual chemically required volume of KOH.
- Step 5 is incorrect. The explanation here misunderstands the effect of a contaminating base. A contaminating base in a sample would actually reduce the amount of KOH required to reach the endpoint because it would partially neutralize the acid (phenol) present before the KOH is added, not require more KOH as stated.
- Step 6 is also incorrect but the identified first error is in Step 5. A contaminating acid would indeed require more KOH to reach the endpoint because it adds to the total amount of acid that must be neutralized.
- Step 7 reaches an incorrect conclusion based on incorrect reasoning presented in Step 5. Solution Correctness: incorrect First Error Step: Step 5 Error Reason: The error in reasoning is that a contam-

inating base would lead to an increased consumption of KOH. In reality, a contaminating base would have already neutralized some of the acid present, thus requiring less KOH to reach the endpoint of the titration. The corrected reasoning should suggest that a contaminating acid (not base) in the unclean beaker would explain why more KOH was required to reach the endpoint, pointing toward Choice D as the correct explanation rather than Choice C.
Solution Correctness: incorrect
First Error Step: Step 5
Error Reason: The error in reasoning is that a contaminating base would lead to an increased consumption of KOH. In reality, a contaminating base would have already neutralized some of the acid present, thus requiring less KOH to reach the endpoint of the titration. The corrected reasoning should suggest that a contaminating acid (not base) in the unclean beaker would explain why more KOH was required to reach the endpoint, pointing toward Choice D as the correct explanation rather than Choice C.
Need Error Reason Review: True
Error Reason Correctness Analysis:
Scoring Model: gpt-4-turbo-2024-04-09
Scoring Raw Response:
1. Step-by-Step Reasoning:
- The error identifies the misconception about the effect of a contaminating base in one of the samples.
- It is incorrectly stated that a contaminating base would lead to an increased consumption of KOH to reach the endpoint.
- In reality, if a base contaminant had reacted with the phenol, it would have neutralized some of the acid, thereby reducing the amount of KOH necessary to reach the endpoint.
- The correct understanding is that a contaminating acid, not a base, would require additional KOH to neutralize, thus explaining the increased volume of KOH needed for one of the samples.
2. Student Error Reason Analysis:
- The student correctly identifies the error in reasoning regarding the effect of a contaminating base on the consumption of KOH.
- They accurately point out that a contaminating base would result in less, not more, KOH needed to reach the titration endpoint because the base would have neutralized some of the acid present.
- The student correctly redirects the reasoning towards a contaminating acid being the cause for the increased KOH volume required, aligning with the true reason for the observed discrepancy.
- This explanation correctly shifts from Choice C (a contaminating base) to Choice D (a contaminating acid) as the valid reason for the anomaly in the titration results.
3. Final Decision: Correct
Annotation Analysis:
- The error identifies the misconception about the effect of a contaminating base in one of the samples.
- It is incorrectly stated that a contaminating base would lead to an increased consumption of KOH to reach the endpoint.
- In reality, if a base contaminant had reacted with the phenol, it would have neutralized some of the acid, thereby reducing the amount of KOH necessary to reach the endpoint.
- The correct understanding is that a contaminating acid, not a base, would require additional KOH to neutralize, thus explaining the increased volume of KOH needed for one of the samples.
Error Reason Analysis:
- The student correctly identifies the error in reasoning regarding the effect of a contaminating base on the consumption of KOH.
- They accurately point out that a contaminating base would result in less, not more, KOH needed to reach the titration endpoint because the base would have neutralized some of the acid present.
- The student correctly redirects the reasoning towards a contaminating acid being the cause for the increased KOH volume required, aligning with the true reason for the observed discrepancy.
- This explanation correctly shifts from Choice C (a contaminating base) to Choice D (a contaminating acid) as the valid reason for the anomaly in the titration results.
Error Reason Correctness: Correct

Figure E.5-2. A success case in the chemistry subject.

E.6 Logic

Question UUID: 57a04b12-b958-4a52-95a6-caf1e5261966
Subject: logic
Question:
A highway management office decided to recruit 10 auxiliary road managers to solve the problem of shortage of formal managers, but this proposal aroused opposition from some people. The person believes that with the addition of these 10 road auxiliary managers, there will be more road violations and disciplinary violations will be found, and more formal managers will be needed to deal with these problems in the later stage. This will lead to a shortage of already tense formal managers. Which of the following, if true, will most weaken that person's point of view?
Options:
A: The newly recruited auxiliary road managers may not be able to do their job with due diligence
B: There are many road violations, and the problem of disciplinary violations can be solved on the spot, without delaying to post-processing
C: Auxiliary road managers can also deal with road violations and disciplinary violations later
D: The increase of road auxiliary managers will effectively reduce the occurrence of road violations and disciplinary violations on this section of the road
Ground Truth Answer: C
Ground Truth Answer Modified: C
——————————— **Sampled CoT** ———————————
Sampled Model: GPT3.5
Model Solution Steps:
Step 1: If the newly recruited auxiliary road managers may not be able to do their job with due diligence, it could potentially lead to more road violations and disciplinary issues, supporting the opposition's point of view.
Step 2: If there are many road violations, and the problem of disciplinary violations can be solved on the spot without delay, it weakens the argument that more formal managers will be needed to handle the issues later.
Step 3: If auxiliary road managers can also deal with road violations and disciplinary violations later, it weakens the argument that more formal managers will be required to handle these problems.
Step 4: If the increase of road auxiliary managers will effectively reduce the occurrence of road violations and disciplinary violations, it completely weakens the opposition's point of view that more formal managers will be needed to handle the expected increase in violations.
Solution: Choice_D
Solution Correctness: incorrect
Model Solution First Error Step: 4
Model Solution Error Reason:
Option D does not completely weaken the opponent's point of view. Because it only says that increasing personnel can effectively reduce violations, but the amount of reduction is uncertain, and the impact on later work is also uncertain. Therefore, the final solution reasoning result is incorrect.
This does not completely weaken the opponent's point of view. Because it lacks the specific number of how increasing personnel can effectively reduce violations, it does not provide the most reliable evidence.
Model Solution Rectified First Error Step:
Step 4: If the added auxiliary road managers can effectively reduce the occurrence of road violations and disciplinary violations, this to some extent weakens the opponent's point of view. But because it only says that increasing personnel can effectively reduce violations, but the amount of reduction is uncertain, and the impact on later work is also uncertain. Therefore, in comparison, option C is the most weakening option. Solution: Option C.
Step 4: If the added auxiliary road managers can effectively reduce the occurrence of road violations and disciplinary violations, this weakens the opponent's point of view, that is, more formal managers are needed to deal with the expected increase in violations. But it lacks the specific number of how increasing personnel can effectively reduce violations, it does not provide the most reliable evidence. Option C states that even if violations are discovered, auxiliary personnel can also solve the problem, so it will not increase the workload of formal personnel. Therefore, it most weakens the point of view. Solution: Option C.
———————————**Model Evaluation Result**———————————
Evaluated Model: gpt-4-turbo-2024-04-09
Evaluation Raw Response:
Solution Analysis:
- Step 1 explains Option A in which the auxiliary road managers may not perform well, potentially increasing issues rather than reducing them. This step is neutral as it provides background to Option A but does not directly weaken the opposition's viewpoint.
- Step 2 discusses Option B, suggesting that immediate resolution of violations reduces the need for formal managers later. This reasoning correctly identifies how this option weakens the argument. This step is correct.
- Step 3 examines Option C, indicating that auxiliary road managers being able to handle violations later could also weaken the need for formal managers. This is a valid point, and this step is correct.
- Step 4 analysis Option D and concludes that it directly weakens the opposition's point by reducing the occurrence of violations. This analysis is correct with sound reasoning.
- The selected solution (Choice D) directly addresses the concern about increasing violations requiring more formal managers by positing that violations would decrease. Solution Correctness: correct First Error Step: N/A Error Reason: N/A
Solution Correctness: correct
First Error Step: N/A
Error Reason: N/A
Need Error Reason Review: False

Figure E.6-1. A failure case in the logic subject.

Question UUID: 17f4f79c-e8db-4514-b047-fbb2f905e7f6
Subject: logic
Question:
Living gifts refer to the donation of money to children, grandchildren and other relatives when the donor decides to live, or to use it for vacations and the establishment of trust funds. Among the following options, which is a living gift?
Options:
A: Mr. Wang made a will before his death, leaving 90% of his property to his only granddaughter.
B: Chairman Li handed over 80% of his property and the entire group to his daughter, and then lived in the mountain village by himself.
C: Grandma Wang asked her grandson to get married in her own house and live in a nursing home by herself.
D: Two years after Mr. Li disappeared, his son inherited his property. One day, he suddenly appeared at home.
Ground Truth Answer: B
Ground Truth Answer Modified: B
——————————— **Sampled CoT** ———————————
Sampled Model: GPT3.5
Model Solution Steps:
Step 1: Mr. Wang leaving 90% of his property to his only granddaughter in his will is not a living gift as it occurs after his death.
Step 2: Chairman Li handing over 80% of his property to his daughter and then living in a mountain village is not a living gift as it also occurs before his death.
Step 3: Grandma Wang asking her grandson to get married in her own house and living in a nursing home by herself is not a living gift, but more of a familial arrangement.
Step 4: Two years after Mr. Li disappeared, his son inherited his property, which suggests it was not a planned living gift.
Conclusion: The only option that fits the definition of a living gift is Choice_A: Mr. Wang made a will before his death, leaving 90% of his property to his only granddaughter.
Solution Correctness: incorrect
Model Solution First Error Step: 2
Model Solution Error Reason:
The model incorrectly assumes that Chairman Li handing over 80% of his property to his daughter is not a living gift. According to the definition of living gifts, these gifts are given when the donor is alive, and option B describes Chairman Li giving his property to his daughter during his lifetime, which fits the definition of a living gift. Therefore, the judgment in step 2 is wrong.
Model Solution Rectified First Error Step:
Step 2: It should be pointed out that option B describes a living gift because Chairman Li transferred his property to his daughter during his lifetime, which is in line with the definition of living gifts.
Step 2: Option B describes a case of living gifts because Chairman Li transferred his property to his daughter during his lifetime, which is consistent with the definition of living gifts, that is, the donation is made when the donor is alive.
————————————**Model Evaluation Result**————————————
Evaluated Model: gpt-4-turbo
Evaluation Raw Response:
Solution Analysis:
Step 1: This step correctly identifies that Mr. Wang's will, which comes into effect after his death, does not qualify as a living gift. The reasoning is sound because living gifts are defined as donations made while the donor is still alive.
Step 2: This step misinterprets the definition of a living gift. Chairman Li handing over 80% of his property to his daughter and then living elsewhere matches the definition of a living gift since it is done while he is alive. Thus, the conclusion drawn in this step is incorrect.
Step 3: The analysis here is correct in determining that Grandma Wang's action doesn't directly involve a transfer of money or property as a gift to live; instead, it's about a living arrangement. However, it could still be considered a form of living gift indirectly, but the explanation does not explore this possibility. The step is more neutral due to lack of exploration.
Step 4: The identification that Mr. Li's post-disappeared inheritance does not constitute a living gift is correct, as it relies on the absence and presumed death of Mr. Li.
Conclusion: The conclusion drawn is incorrect, as it identifies Option A as the correct answer, which was established in the first step as not fitting the definition of a living gift due to it taking effect after death.
Solution Correctness: incorrect
First Error Step: Step 2
Error Reason: The first error occurs in Step 2, where the solution incorrectly states that Chairman Li handing over property to his daughter is not a living gift because it "occurs before his death." This contradicts the very definition of a living gift, which is specified to be the donation of money, property, etc., to relatives while the donor is still alive. Chairman Li's action of transferring his property to his daughter while he is alive perfectly aligns with the concept of a living gift. The corrected reasoning for Step 2 would affirm that Chairman Li's action is indeed a living gift, directly reflecting the definition provided in the question.
Solution Correctness: incorrect
First Error Step: Step 2
Error Reason: The first error occurs in Step 2, where the solution incorrectly states that Chairman Li handing over property to his daughter is not a living gift because it "occurs before his death." This contradicts the very definition of a living gift, which is specified to be the donation of money, property, etc., to relatives while the donor is still alive. Chairman Li's action of transferring his property to his daughter while he is alive perfectly aligns with the concept of a living gift. The corrected reasoning for Step 2 would affirm that Chairman Li's action is indeed a living gift, directly reflecting the definition provided in the question.
Need Error Reason Review: True
Error Reason Correctness: Correct

Figure E.6-2. A success case in the logic subject.

E.7 Coding

Subject: coding
Question:
Write a Python function according to the function name and the problem description in the docstring below.

```
from typing import List
def maximum_size_after_removal(nums1:  List[int], nums2:  List[int]):
### You are given two memory quantities nums1 and nums2 whose subscripts start from 0, and their lengths
are both even n.
You must delete n / 2 elements from nums1 and n / 2 elements from nums2.
After deletion, you insert the remaining elements from nums1 and nums2 into the set s.
Returns the maximum number of possible collections
###
>>> maximumSizeafterRemoval([3,4], [1,2])
2
>>> maximumSizeafterRemoval([1,2,1,2], [1,1,1,1])
2
```

——————————— **Sampled CoT** ———————————

Sampled Model: GPT3.5
Model Solution Steps:

```
def maximum_size_after_removal(nums1: List[int], nums2: List[int]):
    # Sort the lists and get the maximum collection
    nums1.sort()
    nums2.sort()

    # Determine the length of the lists
    n = len(nums1)

    # Calculate the number of elements to delete which is n / 2
    remove_count = n // 2

    # Create a set from the concatenated lists removing elements as required
    s = set(nums1[remove_count:] + nums2[remove_count:])

    # Return the maximum number of elements in the resulting collection set
    return len(s)

    # Test cases
    print(maximum_size_after_removal([3, 4], [1, 2])) # 2
    print(maximum_size_after_removal([1, 2, 1, 2], [1, 1, 1, 1])) # 2
```

Solution Correctness: incorrect
Model Solution First Error Step:
Return the maximum number of elements in the resulting collection set
Model Solution Error Reason:
The provided code has a problem because it does not appropriately select which elements to remove from 'nums1' and 'nums2' to maximize the size of the unique set of remaining elements. The code simply trims the first ' $n/2$ ' elements from each array, which does not necessarily lead to the maximum possible unique set size. Especially, if the smaller elements are frequently repeated and just trimmed away, this method could result in not maximizing the size of the final set.
Model Solution Rectified First Error Step:
Find the combination of elements to remove that maximizes the size of the resulting set

———————————**Model Evaluation Result**———————————

Evaluated Model: gpt-4-turbo
Evaluation Raw Response:
Solution Analysis:
1. The code starts by sorting the 'nums1' and 'nums2' lists. Sorting is neutral in this context since ensuring order is not necessary but doesn't inherently cause an error.
2. It then determines the length 'n' of 'nums1' (assumed to be equal to 'nums2' since both are said to be of even length 'n'). This is correct logic because the length is necessary to compute the number of elements to be removed.
3. The code calculates the number of elements to be deleted, 'remove_count' as 'n // 2'. Given that 'n' is always even (as stipulated by the question), this is a correct computation ensuring integers for indexing.
4. It then creates a set 's' by concatenating 'nums1' and 'nums2' from indices 'remove_count' onwards. This directs removal of the first 'n // 2' elements from each list and thus working on the latter half of both lists.
5. Finally, it returns the size of set 's' which represents the number of unique elements in the resulting collected elements. This operation is correctly used by 'len(s)'. The provided implementation perfectly resonates with the objective of the algorithm as explained by the docstring.
Solution Correctness: correct
First Error Step: N/A
Error Reason: N/A
Need Error Reason Review: False

Figure E.7-1. A failure case in coding. The sampled code fails to identify the key point that the set s is unindexed, hence duplicate elements will be removed. The assessed GPT-4 fails to annotate the error when evaluating the code solution, only blindly accepting the code as logically correct and computable.

Question: Write a Python function according to the function name and the problem description in the docstring below.

```
from typing import List
def arrange_ark_pairs(ark\_deck: List[int]) -> int:
    """
Legend tells of a great Ark built by Noah to survive an immense flood
that would cover the Earth. To preserve the natural world, Noah invited
animals to join him on the Ark, inviting them in pairs so that each
species could continue in the new world. As the animals boarded the Ark,
they were assigned places in a linear formation across the Ark's deck.
However, in the rush to board before the rain began, the animal pairs
became separated across the 2n available spaces. Each animal is known
by a unique identifier, and the list of these identifiers as they are
arranged on the Ark is given by an integer array ‘arkDeck‘ where
‘arkDeck[i]‘ represents the animal occupying the ith space. The pairs
were meant to board in order, with the first pair being (0, 1), the
second pair (2, 3), and so on, up to the last pair being (2n - 2, 2n - 1)
.
Your task is to help Noah figure out the minimum number of exchanges
necessary to reposition the animals so that each pair is resting side
by side. An exchange is the act of two animals, regardless of their
species, standing up from their places and switching spots on the deck.
    >>> arrange_ark_pairs([0,1,3,2])
    0
    >>> arrange_ark_pairs([0,3,2,1])
    1
    """
```

Ground Truth Answer:
Sampled Model: GPT3.5
Model Solution Steps:

```
from typing import List

def arrange_ark_pairs(ark\_deck: List[int]) -> int:
    swaps = 0  # Initialize a variable to count the number of swaps
                                        needed
    n = len(ark\_deck) // 2  # Calculate the number of pairs based on the
                                        deck size

    # Iterate over the pairs and check if they are in order or not
    for i in range(n):
        if ark\_deck[2*i] == 2*i and ark\_deck[2*i+1] == 2*i+1:
            continue  # If the pair is in order, move to the next pair
        else:
            # If the pair is not in order, find the position of the
                                                second member of the
                                                pair in the deck
            for j in range(2*i+1, len(ark\_deck)):
                if ark\_deck[j] == 2*i+1:
                    # Swap the positions of the two members of the pair
                    ark\_deck[2*i+1], ark\_deck[j] = ark\_deck[j], ark\
                                                        _deck[2*i+1]
                    swaps += 1  # Increment the swap count by 1
                    break

    return swaps

# Test cases
print(arrange_ark_pairs([0,1,3,2]))  # Output: 0 (No swaps needed as
                                      pairs are in order)
print(arrange_ark_pairs([0,3,2,1]))  # Output: 1 (1 swap needed to
                                      correct the pair positions)
```

Model Solution Correctness: incorrect
Model Solution First Error Step: continue # If the pair is in order, move to the next pair
Model Solution Error Reason: The model attempts to verify whether adjacent pairs meet the requirements. The 'if' statement defaults to the fact that the animals must stand in a specific position according to the number, but the animals only required to be in pairs. For example, '[0,1,2,3]' is a reasonable pair, and '[2,3,0,1]' is also a reasonable pair. One solution is to only check whether the adjacent numbers are pairs.

Model Solution Rectified First Error Step:

```
if ark\_deck[2 * i] == ark\_deck[2 * i + 1] + 1 or ark\_deck[2 * i] ==
                                       ark\_deck[2 * i + 1] - 1:
```

————————————————Model Evaluation Result————————————————

Evaluated Model: gpt-4-turbo
Evaluation Result:

Solution Analysis:
1. The function signature and the import statement are correct and set the right context for the problem.
2. Initializing ‘swaps‘ to ‘0‘ is correct as it prepares to count the number of swaps necessary, thus contributing to solving the problem.
3. Calculating ‘n‘ as ‘len(ark_deck) // 2‘ correctly determines the number of pairs, given the problem's constraints.
4. The ‘for‘ loop starting with ‘for i in range(n):‘ sets up iteration over each pair correctly
. 5. The conditional ‘if ark_deck[2*i] == 2*i and ark_deck[2*i+1] == 2*i+1:‘ is incorrect. The logic assumes that the pairs being in the correct order is solely dependent on their indices being even and odd sequentially, which does not align with the problem statement regarding how pairs should be identified and arranged.
6. The ‘else‘ block starting iteration from ‘2*i+1‘ to look for the pair is a reasonable approach to find and swap the misplaced member of a pair.
7. The swapping logic inside the ‘else‘ block and incrementing the ‘swaps‘ counter by 1 if a swap is made is logically sound, contributing to achieving the task.
8. Returning ‘swaps‘ correctly provides the solution to the problem based on the accumulated value.
9. The provided test cases confirm the solution's approach under specific conditions but might not sufficiently test all edge cases.
Solution Correctness: incorrect

First Error Step: $ifark\_deck[2*i] == 2*iandark\_deck[2*i+1] == 2*i+1:$
solution correctness: incorrect
error reason: This line incorrectly assumes that the correct order of pairs is determined by checking if the indices and values match such that both members of the pair have indices and values as even and odd sequentially (i.e., 2i and 2i+1). The problem statement describes that each pair is identified by their values being sequential (but not necessarily matching their positions sequentially as even or odd) because the identifiers themselves define the pairs (e.g., (0, 1), (2, 3), etc.), not their positions on the deck. A correct approach involves comparing each element to find its pair based on value, not position, considering the unique identifier nature of the animals. A possible correction involves removing or significantly modifying the condition to ensure it checks for pairs based on their values and not their positions in the deck:
- The solution should instead locate the partner of every animal by calculating the expected partner's identifier using an equation derived from its own identifier, considering the identifiers determine the pairs. This involves correcting the logic to find and swap the animals correctly, which might remove or alter this conditional check entirely in favor of an approach that considers the values of the identifiers rather than their positions in the array.
Need Error Reason Review: True

————————————————Error Reason Scoring————————————————

Evaluated Model: gpt-4-turbo
Evaluation Raw Response:
1. Step-by-Step Reasoning:
- The ground truth error is that the conditional check in the code incorrectly assumes that animals must be in sequential order by their identifiers on the deck (i.e., (0, 1) must be at positions 0 and 1, (2, 3) at positions 2 and 3, etc.).
- The provided solution iterates over the elements of ‘ark_deck‘ in pairs and checks if the values at positions ‘2*i‘ and ‘2*i+1‘ are exactly ‘2*i‘ and ‘2*i+1‘.
- However, the problem statement specifies that pairs only need to be side by side and does not mandate a specific order on the deck. This means that as long as each pair's identifiers are adjacent, the order is acceptable.
- The error is in the condition ‘if ark_deck[2*i] == 2*i and ark_deck[2*i+1] == 2*i+1:‘. This condition fails to check if the adjacent numbers are pairs regardless of their order on the deck.
- The rectified step suggests an alternative approach that verifies if adjacent elements are part of the same pair by checking their values, not their positions.

2. Student Error Reason Analysis:
- The student correctly identifies the first error step in the solution as the condition that checks if the indices and values match in a specific sequential pattern.
- The student explains that the solution should check for pairs based on their values rather than their positions. This is aligned with the ground truth that the pairs only need to be adjacent, not in a specific sequential position on the deck.
- The student's proposed correction to remove or modify the condition to ensure it checks for adjacent numbers being pairs based on their identifiers is also correct. This implies a logic change that focuses on the value of the animal identifiers rather than their positions in the array.

3. Final Decision: Correct
Annotation Analysis:
- The ground truth error is that the conditional check in the code incorrectly assumes that animals must be in sequential order by their identifiers on the deck (i.e., (0, 1) must be at positions 0 and 1, (2, 3) at positions 2 and 3, etc.).
- The provided solution iterates over the elements of ‘ark_deck‘ in pairs and checks if the values at positions ‘2*i‘ and ‘2*i+1‘ are exactly ‘2*i‘ and ‘2*i+1‘.
- However, the problem statement specifies that pairs only need to be side by side and does not mandate a specific order on the deck. This means that as long as each pair's identifiers are adjacent, the order is acceptable.
- The error is in the condition ‘if ark_deck[2*i] == 2*i and ark_deck[2*i+1] == 2*i+1:‘. This condition fails to check if the adjacent numbers are pairs regardless of their order on the deck.
- The rectified step suggests an alternative approach that verifies if adjacent elements are part of the same pair by checking their values, not their positions.

Error Reason Analysis:
- The student correctly identifies the first error step in the solution as the condition that checks if the indices and values match in a specific sequential pattern.
- The student explains that the solution should check for pairs based on their values rather than their positions. This is aligned with the ground truth that the pairs only need to be adjacent, not in a specific sequential position on the deck.
- The student's proposed correction to remove or modify the condition to ensure it checks for adjacent numbers being pairs based on their identifiers is also correct. This implies a logic change that focuses on the value of the animal identifiers rather than their positions in the array.

Error Reason Correctness: Correct

Figure E.7-2. A success case in the coding subject. The model evaluates the error part correctly, and its error reason annotation aligns with human annotations.

G Computational Resources Used

In this paper, all experiments are either performed on open-source models with local inference or closed-source models with API calls. For local inference, we are using A800 machines with 8 GPUs to run the inferences. The total evaluation time on our 6k benchmark on the 70B language models typically takes around 2 hours using fast inference libraries such as vllm. For smaller language models such as Phi-3 or Gemma, the compute time is smaller.

H Annotation Guidelines

Below we provide the original annotation guidelines distributed to annotators of distinctive subjects included in the MR-Ben benchmark: math, biology, physics, chemistry, logic, medicine, and coding. The guidelines serve as the primary training material and instructions for annotators to complete the labeling tasks, specified with detailed descriptions, requirements, and standards.

Annotation Guideline: Math Subject

1 Background

With the increasingly enhanced capabilities of large language models such as ChatGPT, existing models can break down complex problems and solve them step by step. However, while their solutions may often be seemingly correct, they can contain errors in understanding, calculation, and reasoning logic. In this annotation task, annotators will evaluate model-generated step-by-step solutions to questions in the mathematics subject (including university-level and high-school-level mathematics), and determine whether the model has obtained the correct answer based on the correct understanding and reasoning logic.

2 Annotation Task

Given a question and a step-by-step solution generated by the model, the annotator needs to read the question and model-generated solutions carefully. There will be different solutions from multiple models for one question, and they need to be annotated independently. Each annotation task is divided into four sub-tasks.

2.1 Sub-task 1

The first sub-task is to decide whether the model-generated step-by-step solution is correct. Even if the model's final answer matches the ground-truth answer, the model may arrive at the correct answer through incorrect understanding and reasoning. Therefore, regardless of whether the final answers match the ground-truth choice, annotators should read through each model solution carefully and complete all subtasks based on prudent judgments.

2.2 Sub-task 2

The second sub-task is to identify the first incorrect step in the model-generated solution if the solution is incorrect. Specifically, each step can be interpreted as correct, neutral, or incorrect.

- **Correct** means both the calculation and understanding are correct, and the step is essential for reaching the correct result.
- **Neutral** refers to the current step possibly being a background statement about the question, a factual/formula description, or an exploratory step generated by the model. There are no obvious errors in this step, but whether this step leads to the final correct answer is not obvious.
- **Incorrect** step refers to obvious errors in calculation, understanding, or other possible errors in reasoning, regardless of whether this error affects the final calculation result.

Please note that we only need to find the first 'incorrect' step.

2.3 Sub-task 3

The third sub-task is to describe the reason for the error in the first incorrect step. Annotators should carefully analyze the possible reasons for the error in the incorrect step and provide a rational analysis. Please try to incorporate the logic of the model's solution and analyze the reasons for this error from

a holistic perspective. This error reason may serve as a reference for other annotators or models in the future, so please be as detailed, comprehensive, and clear as possible.

2.4 Sub-task 4

The fourth and final sub-task is to rewrite the first incorrect step. Based on the error reason you provided, correct the incorrect step that you identified. The rewritten step should naturally integrate into the original context of the model solution, with only the incorrect part modified when necessary.

3 Annotation Data Format

At the end of this guideline, Figure 1 shows an annotation example of one model-generated solution for a high-school mathematics question, which includes the following items:

Question, **Options**, **Model_Solution_Steps** and **Ground_Truth_Answer**: They are what annotators should carefully read and analyze.

Model_Solution_Correctness is the first sub-task item to be annotated. It is defined as whether the model solution is completely correct for all steps. Please only answer "Yes" or "No" in this column. Please do not fill in other variants such as correct, incorrect, positive, negative, etc.

Model_Solution_First_Error_Step_Num is the second sub-task item to be annotated. It is defined as the first incorrect step specified in the model solution. If the annotation of the previous item is "No", please enter the numerical step number with quotation marks here, such as "1", "2", etc.; If the result of the first item annotation is "Yes", fill in "None" here. Please note:

1. Do not fill in other variants such as "one", "two", or "none", "no", etc.
2. Step numbers should be written in double quotation marks; do not just write numbers.

Model_Solution_Error_Reason is the third sub-task item to be annotated. Please provide a detailed error analysis for the identified first incorrect step here. For example, whether the model refers to incorrect factual knowledge / correct facts but incorrect comprehension, calculation / incorrect contextual logic links / improper conditions or assumptions, etc. If the annotation of the first item "Model_Solution_Correctness" is "Yes", fill in "None" here.

Rectified_Model_Solution_First_Error_Step is the fourth and last sub-task item to be annotated. Please correct the incorrect step based on the error reason you provided. If the annotation of the first item "Model_Solution_Correctness" is "Yes", fill in "None" here. Note that we are not responsible for the consequential steps following the first incorrect step; we only need to correct the identified first error here.

4 Annotation Regulations & Quality Control

All questions are to be annotated according to the format requirements. If issues arise with the annotation format from sampling inspection, annotators should make modifications promptly.

If the annotations of error reasons and step corrections are too brief, perfunctory, or unclear, re-annotation is required to ensure the annotation quality. For each batch of submitted files, if any of the following occurred, all questions of the same batch will be sent back for re-examination and re-annotation:

1. the annotation error rate is higher than 5% after random sampling, or
2. any of the requirements in this guideline are found violated.

5 Additional Notes for Math Subject

5.1 Look-Up Source

The mathematics subject of this task includes annotating questions in the difficulty level of American university and high school mathematics. When encountering unfamiliar or unclear mathematical terms and concepts, please use search engines and conduct factual inquiries from authoritative sources to assist with your annotation. Please do not use any language models as your annotation assistance tool. Contact the annotation management team whenever in doubt.

5.2 Math Expressions Format

The writing of mathematical formulas and special symbols is in the LaTeX language.

1. For convenience in question interpretation, annotators can use website tools such as `https://www.latexlive.com/` to convert and view the content enclosed in the symbols in the question. Since latexlive.com uses a single backslash instead of a double backslash, please change the two backslashes (\\) inside the mathematical formula in the question (e.g., $\\frac{23}{3}$) to a single backslash (\\) (e.g., $\frac{23}{3}$) when previewing the formula.
2. Correspondingly, when filling in the annotations, annotators need to convert and write the mathematical formulas: please change the single backslash (\) back to a double backslash (\\) and enclose the mathematical formula with. Examples of specific filling can be seen in the sample question annotation.

6 Common Questions

Q1: When a step repeats the incorrect option should I mark it as the first incorrect step?

No, if a step only describes what an incorrect option is stating or repeats the exact words of the incorrect option without analysis, it does not constitute an incorrect step. But if the step includes wrong analysis/judgment about the incorrect option, then it is considered incorrect.

Q2: If previous steps so far have only stated relevant facts without specific analysis, how should I handle these steps?

Similarly, if the steps focus solely on factual statements about the options or related background descriptions, these steps should be understood as neutral steps unless these steps contain incorrect understanding, incorrect facts, etc. The first incorrect step should be the one that involves incorrect analysis or reasoning.

Q3: What should I do if I encounter difficulty comprehending the question options or model solutions?

When encountering unfamiliar terms, look up textbook sources or use search engines to find reliable explanations first. Contact the annotation management team whenever in doubt.

Q4: The model's answer gives two answers, one of which is correct. How should I decide?

Annotate it as "No." Then follow the standard process to annotate the incorrect step, error reason, and the correction step.

Q5: It's inconvenient to read the JSON file without line breaks.

If you are using platforms like Visual Studio Code to annotate files in JSON format, you can install plugins like 'json editor' to assist in reading. However, take caution against modifying any content other than the required subtasks and do not manually change the data structure.

7 Annotation Example

Figure 1 below is a model solution of a sample question to be annotated; Figure 2 shows an example annotation for the same model solution.

```
{
  "Question_UUID": "ef1e7d22-346e-4383-92b6-0cca25834f7d",
  "Subject": "hs_math",
  "Question": "Given that $a(a+2b) = \\frac{104}3$, $b(b+2c) = \\frac{7}{9}$, and $c(c+2a) = -7$, find $|a+b+c|$.",
  "Options": "A: \\frac{23}{3} \nB: \\frac{5}{3} \nC: \\frac{16}{3} \nD: \\frac{10}{3} ",
  "Ground_Truth_Analysis": "",
  "Ground_Truth_Answer": "C",
  "Model_Solution_Steps": [
    "Step 1: Express a, b, and c in terms of the other variables using the given equations.",
    "From $a(a+2b) = \\frac{104}{3}$, we get $a = \\frac{104}{3(a+2b)}$.",
    "From $b(b+2c) = \\frac{7}{9}$, we get $b = \\frac{7}{9(b+2c)}$.",
    "From $c(c+2a) = -7$, we get $c = -\\frac{7}{c+2a}$.",
    "Step 2: Substitute the expressions for a, b, and c found in Step 1 into the equation for c (since it has the fewest terms).",
    "Substituting a into the equation for c gives:",
    "$c = -\\frac{7}{c + 2(\\frac{104}{3(a+2b)})}$.",
    "Step 3: Simplify the equation for c from Step 2.",
    "First, clear the fraction by multiplying both sides of the equation by the denominator:",
    "$-7(c+2(\\frac{104}{3(a+2b)})) = c(c + 2(\\frac{104}{3(a+2b)}))$.",
    "Expanding and simplifying further, we obtain:",
    "$-\\frac{21c}{3} - \\frac{1428}{9(a+2b)} = c^2 + \\frac{208c}{3(a+2b)}$.",
    "Step 4: Express c in terms of a and b by solving the quadratic equation found in Step 3.",
    "To do this, first rearrange the equation to the standard quadratic form:",
    "$c^2 + \\left(\\frac{208}{(3(a+2b)} + \\frac{21}{3}\\right)c + \\frac{1428}{9(a+2b)} = 0$.",
    "Then, solve for c using the quadratic formula.",
    "Step 5: Substitute the expression for a found in Step 1 into the equation for c from Step 4, then solve for c in terms of b.",
    "Recall that $a = \\frac{104}{3(a+2b)}$. Thus, after some simplification, we obtain:",
    "$c^2 + \\left(\\frac{208}{(3(\\frac{104}{3a}+2b)} + \\frac{21}{3}\\right)c + \\frac{1428}{9(\\frac{104}{3a}+2b)} = 0$.",
    "Solving for c in terms of b yields two possible solutions. Choose one of them and continue the analysis.",
    "Step 6: Substitute the expressions for a and c found in Steps 1 and 5, respectively, into the equation for b from Step 1, then
solve for b.",
    "Recall that $b = \\frac{7}{9(b+2c)}$. Substitute the expressions for a and c from Steps 1 and 5 into this equation, then solve
for b.",
    "Step 7: Finally, substitute the solutions for a, b, and c found in Steps 1, 6, and 5 (respectively) into the expression
$|a+b+c|$.",
    "Evaluate the absolute value of the sum of a, b, and c using the found solutions from the previous steps.",
    "Step 8: Compare the result from Step 7 with the given Choice_A, Choice_B, Choice_C, and Choice_D to find the matching option.",
    "The correct choice should be equal to the value obtained from the previous step.",
    "Solution: Choice_A"
  ],
  "Sampled_Model": "mistral-medium",
  "Model_Solution_Correctness": "TBD",
  "Model_Solution_First_Error_Step_Num": "TBD",
  "Model_Solution_Error_Reason": "TBD",
  "Rectified_Model_Solution_First_Error_Step": "TBD"
}
```


Figure 1: A task example of a model-generated solution for a high-school mathematics question (to be annotated).


```
{     .
      .
      .
  "Model_Solution_Correctness": "No",
  "Model_Solution_First_Error_Step_Num": "3",
  "Model_Solution_Error_Reason": "Step 3 has an elimination arithmetic error when eliminating fractions
by multiplying the denominator: eliminating fractions by multiplying the denominator $-7(c +
2(\\frac{104}{3(a+2b)})) = c(c + 2(\frac{104}{3(a+2b)}))) $ yields $ -7 = c(c + 2(\\\frac{104}
{3(a+2b)}))$, not $-7(c + 2(\\frac{104}{3(a+2b)})) = c(c + 2(\frac{104}{3(a+2b)})))$.",
  "Rectified_Model_Solution_First_Error_Step": "Simplify the $c$ equation in step 2. First, eliminate
the fraction by multiplying the denominator: $-7 = c(c + 2(\\frac{104}{3(a+2b)}))$. Expanding and
simplifying further, we get: $- = c^2 + \\\frac{208c}{3(a+2b)}$."
}
```


Figure 2: An example of standard annotation (with the previous items omitted to save space).

Annotation Guideline: Physics Subject

1 Background

With the increasingly enhanced capabilities of large language models such as ChatGPT, existing models can break down complex problems and solve them step by step. However, while their solutions may often be seemingly correct, they can contain errors in understanding, calculation, and reasoning logic. In this annotation task, annotators will evaluate model-generated step-by-step solutions to questions in the physics subject (including university-level and high-school-level physics), and determine whether the model has obtained the correct answer based on the correct understanding and reasoning logic.

2 Annotation Task

Given a question and a step-by-step solution generated by the model, the annotator needs to read the question and model-generated solutions carefully. There will be different solutions from multiple models for one question, and they need to be annotated independently. Each annotation task is divided into four sub-tasks.

2.1 Sub-task 1

The first sub-task is to decide whether the model-generated step-by-step solution is correct. Even if the model's final answer matches the ground-truth answer, the model may arrive at the correct answer through incorrect understanding and reasoning. Therefore, regardless of whether the final answers match the ground-truth choice, annotators should read through each model solution carefully and complete all subtasks based on prudent judgments.

2.2 Sub-task 2

The second sub-task is to identify the first incorrect step in the model-generated solution if the solution is incorrect. Specifically, each step can be interpreted as correct, neutral, or incorrect.

- **Correct** means both the calculation and understanding are correct, and the step is essential for reaching the correct result.
- **Neutral** refers to the current step possibly being a background statement about the question, a factual/formula description, or an exploratory step generated by the model. There are no obvious errors in this step, but whether this step leads to the final correct answer is not obvious.
- **Incorrect** step refers to obvious errors in calculation, understanding, or other possible errors in reasoning, regardless of whether this error affects the final calculation result.

Please note that we only need to find the first 'incorrect' step.

2.3 Sub-task 3

The third sub-task is to describe the reason for the error in the first incorrect step. Annotators should carefully analyze the possible reasons for the error in the incorrect step and provide a rational analysis. Please try to incorporate the logic of the model's solution and analyze the reasons for this error from

a holistic perspective. This error reason may serve as a reference for other annotators or models in the future, so please be as detailed, comprehensive, and clear as possible.

2.4 Sub-task 4

The fourth and final sub-task is to rewrite the first incorrect step. Based on the error reason you provided, correct the incorrect step that you identified. The rewritten step should naturally integrate into the original context of the model solution, with only the incorrect part modified when necessary.

3 Annotation Data Format

At the end of this guideline, Figure 1 shows an annotation example of one model-generated solution for a physics question, which includes the following items:

Question, **Options**, **Model_Solution_Steps** and **Ground_Truth_Answer**: They are what annotators should carefully read and analyze.

Model_Solution_Correctness is the first sub-task item to be annotated. It is defined as whether the model solution is completely correct for all steps. Please only answer "Yes" or "No" in this column. Please do not fill in other variants such as correct, incorrect, positive, negative, etc.

Model_Solution_First_Error_Step_Num is the second sub-task item to be annotated. It is defined as the first incorrect step specified in the model solution. If the annotation of the previous item is "No", please enter the numerical step number with quotation marks here, such as "1", "2", etc.; If the result of the first item annotation is "Yes", fill in "None" here. Please note:

1. Do not fill in other variants such as "one", "two", or "none", "no", etc.
2. Step numbers should be written in double quotation marks; do not just write numbers.

Model_Solution_Error_Reason is the third sub-task item to be annotated. Please provide a detailed error analysis for the identified first incorrect step here. For example, whether the model refers to incorrect factual knowledge / correct facts but incorrect comprehension, calculation / incorrect contextual logic links / improper conditions or assumptions, etc. If the annotation of the first item "Model_Solution_Correctness" is "Yes", fill in "None" here.

Rectified_Model_Solution_First_Error_Step is the fourth and last sub-task item to be annotated. Please correct the incorrect step based on the error reason you provided. If the annotation of the first item "Model_Solution_Correctness" is "Yes", fill in "None" here. Note that we are not responsible for the consequential steps following the first incorrect step; we only need to correct the identified first error here.

4 Annotation Regulations & Quality Control

All questions are to be annotated according to the format requirements. If issues arise with the annotation format from sampling inspection, annotators should make modifications promptly.

If the annotations of error reasons and step corrections are too brief, perfunctory, or unclear, re-annotation is required to ensure the annotation quality. For each batch of submitted files, if any of the following occurred, all questions of the same batch will be sent back for re-examination and re-annotation:

1. the annotation error rate is higher than 5% after random sampling, or
2. any of the requirements in this guideline are found violated.

5 Additional Notes for Physics Subjects

5.1 Look-Up Source

The physics subject of this task includes annotating questions in the difficulty level of American university and high school physics. When encountering unfamiliar or unclear terms and concepts,

please use search engines and conduct factual inquiries from authoritative sources to assist with your annotation. Please do not use any language models as your annotation assistance tool. Contact the annotation management team whenever in doubt.

5.2 Physics Expressions Format

The writing of physics formulas and special symbols is in the LaTeX language.

1. For convenience in question interpretation, annotators can use website tools such as `https://www.latexlive.com/` to convert and view the content enclosed in the symbols in the question. Since latexlive.com uses a single backslash instead of a double backslash, please change the two backslashes (\\) inside the physics formula in the question (e.g., $\\frac{23}{3}$) to a single backslash (\) (e.g., $\frac{23}{3}$) when previewing the formula.
2. Correspondingly, when filling in the annotations, annotators need to convert and write the physics formulas: please change the single backslash (\) back to a double backslash (\\) and enclose the physics formula with. Examples of specific filling can be seen in the sample question annotation.

6 Common Questions

Q1: When a step repeats the incorrect option should I mark it as the first incorrect step?

No, if a step only describes what an incorrect option is stating or repeats the exact words of the incorrect option without analysis, it does not constitute an incorrect step. But if the step includes wrong analysis/judgment about the incorrect option, then it is considered incorrect.

Q2: If previous steps so far have only stated relevant facts without specific analysis, how should I handle these steps?

Similarly, if the steps focus solely on factual statements about the options or related background descriptions, these steps should be understood as neutral steps unless these steps contain incorrect understanding, incorrect facts, etc. The first incorrect step should be the one that involves incorrect analysis or reasoning.

Q3: What should I do if I encounter difficulty comprehending the question options or model solutions?

When encountering unfamiliar terms, look up textbook sources or use search engines to find reliable explanations first. Contact the annotation management team whenever in doubt.

Q4: The model's answer gives two answers, one of which is correct. How should I decide?

Annotate it as "No." Then follow the standard process to annotate the incorrect step, error reason, and the correction step.

Q5: It's inconvenient to read the JSON file without line breaks.

If you are using platforms like Visual Studio Code to annotate files in JSON format, you can install plugins like 'json editor' to assist in reading. However, take caution against modifying any content other than the required subtasks and do not manually change the data structure.

7 Annotation Example

Figure 1 below is a model solution of a sample question to be annotated; Figure 2 shows an example annotation for the same model solution.

```
{
 "Question_UUID": "db1affc0-bdd0-4c4c-8792-0c228d04fb12",
 "Subject": "high_school_physics",
 "Question": "A body moving in the positive x direction passes the origin at time t = 0. Between t = 0 and t = 1 second, the body has a constant speed of 24 meters per second. At t = 1 second, the body is given a constant acceleration of 6 meters per second squared in the negative x direction. The position x of the body at t = 11 seconds is",
 "Options": "A: 99m \nB: 36m \nC: -36m \nD: -99m ",
 "Ground_Truth_Analysis": "",
 "Ground_Truth_Answer": "C",
 "Ground_Truth_Answer_Modified": "C",
 "Sampled_Model": "mistral-medium",
 "Model_Solution_Steps": [
   "Step 1: During the time interval t = 0 to t = 1 second, the body has a constant speed of 24 m/s in the positive x direction. The displacement during this time is (24 m/s)*(1 s) = 24 m.",
   "",
   "Step 2: At t = 1 second, the velocity becomes zero as the body momentarily comes to rest due to the instantaneous application of a constant acceleration of 6 m/s² in the negative x direction.",
   "",
    "Step 3: For the subsequent time period from t = 1 second to t = 11 seconds, the body's motion is governed by constant acceleration in the negative x direction. Using the kinematic equation x = x₀ + v₀t + ½at² with x₀ = 0, v₀ = 0, a = -6 m/s², and t = 10 seconds (since 11 - 1 = 10), we find x = -300 m.",
   "",
   "Step 4: However, the displacement calculated in Step 3 needs to be added to the displacement found in Step 1, as both displacements are in the same direction. Thus, the final position x is (24 m) + (-300 m) = -276 m.",
   "",
   "Step 5: None of the candidate choices match the calculated value in Step 4. Since the answer must be one of the provided choices, there seems to be a mistake or error in the question or the available choices. We cannot determine the correct solution based on the given information.",
   "",
   "Solution: Unable to determine a solution using the given choices."
 ],
 "Model_Solution_Correctness": "TBD",
 "Model_Solution_First_Error_Step": "TBD",
 "Model_Solution_Error_Reason": "TBD",
 "Model_Solution_Rectified_First_Error_Step": "TBD"
}
```


Figure 1: A task example of a model-generated solution for a high-school-level physics question (to be annotated).


```
 "Model_Solution_Correctness": "incorrect",
 "Model_Solution_First_Error_Step": "2",
 "Model_Solution_Error_Reason": "According to the definition of acceleration: acceleration is a physical quantity that describes how quickly the velocity of an object changes. Combined with the problem statement that \"the object is given a constant acceleration of 6 meters per second squared in the negative x direction,\" it is known that this acceleration only causes the velocity to change gradually and does not make the velocity zero or bring the object to an instant stop.\n"
 "Model_Solution_Rectified_First_Error_Step": "Step 2: At t = 1 second, due to a constant acceleration of -6 meters per second squared applied instantaneously in the negative x direction, the object's velocity first decreases gradually along the positive x direction until it reduces to zero, then increases in the negative x direction.\n"
```


Figure 2: An example of standard annotation (with the previous items omitted to save space).

Annotation Guideline: Biology Subject

1 Background

With the increasingly enhanced capabilities of large language models such as ChatGPT, existing models can break down complex problems and solve them step by step. However, while their solutions may often be seemingly correct, they can contain errors in understanding, calculation, and reasoning logic. In this annotation task, annotators will evaluate model-generated step-by-step solutions to questions in the biology subject (including university-level and high-school-level biology), and determine whether the model has obtained the correct answer based on the correct understanding and reasoning logic.

2 Annotation Task

Given a question and a step-by-step solution generated by the model, the annotator needs to read the question and model-generated solutions carefully. There will be different solutions from multiple models for one question, and they need to be annotated independently. Each annotation task is divided into four sub-tasks.

2.1 Sub-task 1

The first sub-task is to decide whether the model-generated step-by-step solution is correct. Even if the model's final answer matches the ground-truth answer, the model may arrive at the correct answer through incorrect understanding and reasoning. Therefore, regardless of whether the final answers match the ground-truth choice, annotators should read through each model solution carefully and complete all subtasks based on prudent judgments.

2.2 Sub-task 2

The second sub-task is to identify the first incorrect step in the model-generated solution if the solution is incorrect. Specifically, each step can be interpreted as correct, neutral, or incorrect.

- **Correct** means both the calculation and understanding are correct, and the step is essential for reaching the correct result.
- **Neutral** refers to the current step possibly being a background statement about the question, a factual/formula description, or an exploratory step generated by the model. There are no obvious errors in this step, but whether this step leads to the final correct answer is not obvious.
- **Incorrect** step refers to obvious errors in calculation, understanding, or other possible errors in reasoning, regardless of whether this error affects the final calculation result.

Please note that we only need to find the first 'incorrect' step.

2.3 Sub-task 3

The third sub-task is to describe the reason for the error in the first incorrect step. Annotators should carefully analyze the possible reasons for the error in the incorrect step and provide a rational analysis. Please try to incorporate the logic of the model's solution and analyze the reasons for this error from

a holistic perspective. This error reason may serve as a reference for other annotators or models in the future, so please be as detailed, comprehensive, and clear as possible.

2.4 Sub-task 4

The fourth and final sub-task is to rewrite the first incorrect step. Based on the error reason you provided, correct the incorrect step that you identified. The rewritten step should naturally integrate into the original context of the model solution, with only the incorrect part modified when necessary.

3 Annotation Data Format

At the end of this guideline, Figure 1 shows an annotation example of one model-generated solution for a high-school biology question, which includes the following items:

Question, **Options**, **Model_Solution_Steps** and **Ground_Truth_Answer**: They are what annotators should carefully read and analyze.

Model_Solution_Correctness is the first sub-task item to be annotated. It is defined as whether the model solution is completely correct for all steps. Please only answer "Yes" or "No" in this column. Please do not fill in other variants such as correct, incorrect, positive, negative, etc.

Model_Solution_First_Error_Step_Num is the second sub-task item to be annotated. It is defined as the first incorrect step specified in the model solution. If the annotation of the previous item is "No", please enter the numerical step number with quotation marks here, such as "1", "2", etc.; If the result of the first item annotation is "Yes", fill in "None" here. Please note:

1. Do not fill in other variants such as "one", "two", or "none", "no", etc.
2. Step numbers should be written in double quotation marks; do not just write numbers.

Model_Solution_Error_Reason is the third sub-task item to be annotated. Please provide a detailed error analysis for the identified first incorrect step here. For example, whether the model refers to incorrect factual knowledge / correct facts but incorrect comprehension, calculation / incorrect contextual logic links / improper conditions or assumptions, etc. If the annotation of the first item “Model_Solution_Correctness” is "Yes", fill in "None" here.

Rectified_Model_Solution_First_Error_Step is the fourth and last sub-task item to be annotated. Please correct the incorrect step based on the error reason you provided. If the annotation of the first item “Model_Solution_Correctness” is "Yes", fill in "None" here. Note that we are not responsible for the consequential steps following the first incorrect step; we only need to correct the identified first error here.

4 Annotation Regulations & Quality Control

All questions are to be annotated according to the format requirements. If issues arise with the annotation format from sampling inspection, annotators should make modifications promptly.

If the annotations of error reasons and step corrections are too brief, perfunctory, or unclear, re-annotation is required to ensure the annotation quality. For each batch of submitted files, if any of the following occurred, all questions of the same batch will be sent back for re-examination and re-annotation:

1. the annotation error rate is higher than 5% after random sampling, or
2. any of the requirements in this guideline are found violated.

5 Additional Notes for Biology Subject

5.1 Look-Up Source

The biology subject of this task includes annotating questions in the difficulty level of American university and high school biology. When encountering unfamiliar or unclear terms and concepts,

please use search engines and conduct factual inquiries from authoritative sources to assist with your annotation. Please do not use any language models as your annotation assistance tool. Contact the annotation management team whenever in doubt.

6 Common Questions

Q1: When a step repeats the incorrect option should I mark it as the first incorrect step?

No, if a step only describes what an incorrect option is stating or repeats the exact words of the incorrect option without analysis, it does not constitute an incorrect step. But if the step includes wrong analysis/judgment about the incorrect option, then it is considered incorrect.

Q2: If previous steps so far have only stated relevant facts without specific analysis, how should I handle these steps?

Similarly, if the steps focus solely on factual statements about the options or related background descriptions, these steps should be understood as neutral steps unless these steps contain incorrect understanding, incorrect facts, etc. The first incorrect step should be the one that involves incorrect analysis or reasoning.

Q3: What should I do if I encounter difficulty comprehending the question options or model solutions?

When encountering unfamiliar terms, look up textbook sources or use search engines to find reliable explanations first. Contact the annotation management team whenever in doubt.

Q4: The model's answer gives two answers, one of which is correct. How should I decide?

Annotate it as "No." Then follow the standard process to annotate the incorrect step, error reason, and the correction step.

Q5: It's inconvenient to read the JSON file without line breaks.

If you are using platforms like Visual Studio Code to annotate files in JSON format, you can install plugins like 'json editor' to assist in reading. However, take caution against modifying any content other than the required subtasks and do not manually change the data structure.

7 Annotation Example

Figure 1 below is a model solution of a sample question to be annotated; Figure 2 shows an example annotation for the same model solution.

```
{
"Question_UUID": "92f293be-132c-4b43-8937-dfb324d371be",
"Subject": "high_school_biology",
"Question": "Both C4 and CAM plants employ physiological and biochemical strategies to reduce photorespiration. Which of the following is an accurate comparison of the two strategies?",
"Options": "A: C4 plants fix CO2 as the 4-carbon acid oxaloacetate, while CAM plants fix CO2 as the 5-carbon acid ribulose 1, 5-bisphosphate. \nB: C4 plants fix CO2 in the mesophyll but carry out the Calvin cycle in the bundle sheath, while CAM plants fix CO2 and carry out the Calvin cycle in the mesophyll. \nC: C4 plants fix CO2 in the bundle sheath but carry out the Calvin cycle in the mesophyll, while CAM plants fix CO2 and carry out the Calvin cycle in the bundle sheath. \nD: C4 plants store CO2 as phosphoglycolate, while CAM plants store CO2 as oxaloacetate. ",
"Ground_Truth_Analysis": "",
"Ground_Truth_Answer": "B",
"Model_Solution_Steps": [
"Step 1: C4 plants fix CO2 as the 4-carbon acid oxaloacetate, while CAM plants fix CO2 as the 5-carbon acid ribulose 1, 5-bisphosphate.",
"Step 2: This statement accurately describes the difference in the initial fixation of CO2 between C4 and CAM plants.",
"Step 3: C4 plants fix CO2 in the mesophyll but carry out the Calvin cycle in the bundle sheath, while CAM plants fix CO2 and carry out the Calvin cycle in the mesophyll.",
"Step 4: This statement accurately describes the spatial separation of initial CO2 fixation and Calvin cycle in C4 plants but inaccurately describes this process in CAM plants.",
"Step 5: C4 plants fix CO2 in the bundle sheath but carry out the Calvin cycle in the mesophyll, while CAM plants fix CO2 and carry out the Calvin cycle in the bundle sheath.",
"Step 6: This statement inaccurately describes the spatial separation of initial CO2 fixation and the Calvin cycle in C4 plants.",
"Step 7: C4 plants store CO2 as phosphoglycolate, while CAM plants store CO2 as oxaloacetate.",
"Step 8: This statement inaccurately describes how CO2 is stored in C4 and CAM plants.",
"Solution: Choice_A"
],
"Sampled_Model": "GPT3.5",
"Model_Solution_Correctness": "TBD",
"Model_Solution_First_Error_Step_Num": "TBD",
"Model_Solution_Error_Reason": "TBD",
"Rectified_Model_Solution_First_Error_Step": "TBD"
}
```

Figure 1: A task example of a model-generated solution for a high-school biology question (to be annotated).

```
"Model_Solution_Correctness": "No",
"Model_Solution_First_Error_Step_Num": "2",
"Model_Solution_Error_Reason": "The fixation of CO2 to 5-carbonate ribulose 1,5-bisphosphate by CAM plants is an incorrect statement; CO2 is fixed in oxalic acid by the enzyme PEP carboxylase in CAM plants (the same step used by C4 plants), and then it is converted to malic acid or some other type of organic acid.",
"Rectified_Model_Solution_First_Error_Step": "The description of this statement is incorrect. Although C4 plants fix CO2 to oxaloacetate 4 carbonate, photorespiration is more limited in photosynthesis in CAM plants due to the delayed kinetics of enzyme activation, and CO2 is fixed in oxalic acid mainly by PEP carboxylase in CAM plants (the same step as that used by C4 plants), and then is converted to malic acid or other kinds of organic acids."
```

Figure 2: An example of standard annotation (with the previous items omitted to save space).

Annotation Guideline: Medicine Subject

1 Background

With the increasingly enhanced capabilities of large language models such as ChatGPT, existing models can break down complex problems and solve them step by step. However, while their solutions may often be seemingly correct, they can contain errors in understanding, calculation, and reasoning logic. In this annotation task, annotators will evaluate model-generated step-by-step solutions to questions in the medicine subject, and determine whether the model has obtained the correct answer based on the correct understanding and reasoning logic.

2 Annotation Task

Given a question and a step-by-step solution generated by the model, the annotator needs to read the question and model-generated solutions carefully. There will be different solutions from multiple models for one question, and they need to be annotated independently. Each annotation task is divided into four sub-tasks.

2.1 Sub-task 1

The first sub-task is to decide whether the model-generated step-by-step solution is correct. Even if the model's final answer matches the ground-truth answer, the model may arrive at the correct answer through incorrect understanding and reasoning. Therefore, regardless of whether the final answers match the ground-truth choice, annotators should read through each model solution carefully and complete all subtasks based on prudent judgments.

2.2 Sub-task 2

The second sub-task is to identify the first incorrect step in the model-generated solution if the solution is incorrect. Specifically, each step can be interpreted as correct, neutral, or incorrect.

- **Correct** means both the calculation and understanding are correct, and the step is essential for reaching the correct result.
- **Neutral** refers to the current step possibly being a background statement about the question, a factual/formula description, or an exploratory step generated by the model. There are no obvious errors in this step, but whether this step leads to the final correct answer is not obvious.
- **Incorrect** step refers to obvious errors in calculation, understanding, or other possible errors in reasoning, regardless of whether this error affects the final calculation result.

Please note that we only need to find the first 'incorrect' step.

2.3 Sub-task 3

The third sub-task is to describe the reason for the error in the first incorrect step. Annotators should carefully analyze the possible reasons for the error in the incorrect step and provide a rational analysis. Please try to incorporate the logic of the model's solution and analyze the reasons for this error from a holistic perspective. This error reason may serve as a reference for other annotators or models in the future, so please be as detailed, comprehensive, and clear as possible.

2.4 Sub-task 4

The fourth and final sub-task is to rewrite the first incorrect step. Based on the error reason you provided, correct the incorrect step that you identified. The rewritten step should naturally integrate into the original context of the model solution, with only the incorrect part modified when necessary.

3 Annotation Data Format

At the end of this guideline, Figure 1 shows an annotation example of one model-generated solution for a medicine question, which includes the following items:

Question, **Options**, **Model_Solution_Steps** and **Ground_Truth_Answer**: They are what annotators should carefully read and analyze.

Model_Solution_Correctness is the first sub-task item to be annotated. It is defined as whether the model solution is completely correct for all steps. Please only answer "Yes" or "No" in this column. Please do not fill in other variants such as correct, incorrect, positive, negative, etc.

Model_Solution_First_Error_Step_Num is the second sub-task item to be annotated. It is defined as the first incorrect step specified in the model solution. If the annotation of the previous item is "No", please enter the numerical step number with quotation marks here, such as "1", "2", etc.; If the result of the first item annotation is "Yes", fill in "None" here. Please note:

1. Do not fill in other variants such as "one", "two", or "none", "no", etc.
2. Step numbers should be written in double quotation marks; do not just write numbers.

Model_Solution_Error_Reason is the third sub-task item to be annotated. Please provide a detailed error analysis for the identified first incorrect step here. For example, whether the model refers to incorrect factual knowledge / correct facts but incorrect comprehension, calculation / incorrect contextual logic links / improper conditions or assumptions, etc. If the annotation of the first item "Model_Solution_Correctness" is "Yes", fill in "None" here.

Rectified_Model_Solution_First_Error_Step is the fourth and last sub-task item to be annotated. Please correct the incorrect step based on the error reason you provided. If the annotation of the first item "Model_Solution_Correctness" is "Yes", fill in "None" here. Note that we are not responsible for the consequential steps following the first incorrect step; we only need to correct the identified first error here.

4 Annotation Regulations & Quality Control

All questions are to be annotated according to the format requirements. If issues arise with the annotation format from sampling inspection, annotators should make modifications promptly.

If the annotations of error reasons and step corrections are too brief, perfunctory, or unclear, re-annotation is required to ensure the annotation quality. For each batch of submitted files, if any of the following occurred, all questions of the same batch will be sent back for re-examination and re-annotation:

1. the annotation error rate is higher than 5% after random sampling, or
2. any of the requirements in this guideline are found violated.

5 Additional Notes for Medicine Subject

5.1 Look-Up Source

Annotating the medicine subject questions requires professional expertise in the related field, as annotators have been notified and assessed accordingly in the recruitment round. When encountering unfamiliar or unclear terms and concepts, please use search engines and conduct factual inquiries from authoritative sources to assist with your annotation. Please do not use any language models as your annotation assistance tool. Contact the annotation management team whenever in doubt.

6 Common Questions

Q1: When a step repeats the incorrect option should I mark it as the first incorrect step?

No, if a step only describes what an incorrect option is stating or repeats the exact words of the incorrect option without analysis, it does not constitute an incorrect step. But if the step includes wrong analysis/judgment about the incorrect option, then it is considered incorrect.

Q2: If previous steps so far have only stated relevant facts without specific analysis, how should I handle these steps?

Similarly, if the steps focus solely on factual statements about the options or related background descriptions, these steps should be understood as neutral steps unless these steps contain incorrect understanding, incorrect facts, etc. The first incorrect step should be the one that involves incorrect analysis or reasoning.

Q3: What should I do if I encounter difficulty comprehending the question options or model solutions?

When encountering unfamiliar terms, look up textbook sources or use search engines to find reliable explanations first. Contact the annotation management team whenever in doubt.

Q4: The model's answer gives two answers, one of which is correct. How should I decide?

Annotate it as "No." Then follow the standard process to annotate the incorrect step, error reason, and the correction step.

Q5: It's inconvenient to read the JSON file without line breaks.

If you are using platforms like Visual Studio Code to annotate files in JSON format, you can install plugins like 'json editor' to assist in reading. However, take caution against modifying any content other than the required subtasks and do not manually change the data structure.

7 Annotation Example

Figure 1 below is a model solution of a sample question to be annotated; Figure 2 shows an example annotation for the same model solution.


```
{
  "Question_UUID": "5ea0fc1f-7e15-4975-a982-4774ee4e737a",
  "Subject": "college",
  "Question": "Phophocreatine resynthesis during recovery from exercise is inhibited by:",
  "Options": "A: an excess of creatine. \nB: hyperventilation. \nC: an excess of oxygen. \nD: a lack of oxygen. ",
  "Ground_Truth_Analysis": "",
  "Ground_Truth_Answer": "D",
  "Model_Solution_Steps": [
    "Step 1: Phosphocreatine is used to resynthesize ATP during short bursts of intense exercise. Its resynthesis during recovery requires energy.",
    "Step 2: Choice A states that an excess of creatine inhibits phosphocreatine resynthesis. This is correct, as creatine competes with phosphocreatine for the enzyme creatine kinase which is needed for phosphocreatine resynthesis.",
    "Step 3: Choice B states hyperventilation inhibits resynthesis. This is incorrect, as hyperventilation does not affect energy metabolism. ",
    "Step 4: Choice C states an excess of oxygen inhibits resynthesis. This is incorrect, as oxygen is required for aerobic energy production during recovery.",
    "Step 5: Choice D states a lack of oxygen inhibits resynthesis. This is correct, as oxygen is needed for aerobic energy metabolism to provide energy for phosphocreatine resynthesis.",
    "Step 6: Choices A and D are correct. An excess of creatine and a lack of oxygen both inhibit phosphocreatine resynthesis during recovery from exercise.",
     "Solution: Choice A and D"
  ],
  "Sampled_Model": "Claude2",
  "Model_Solution_Correctness": "TBD",
  "Model_Solution_First_Error_Step_Num": "TBD",
  "Model_Solution_Error_Reason": "TBD",
  "Rectified_Model_Solution_First_Error_Step": "TBD"
},
```


Figure 1: A task example of a model-generated solution for a medicine question (to be annotated).

```
    "Model_Solution_Correctness": "No",
    "Model_Solution_First_Error_Step_Num": "2",
    "Model_Solution_Error_Reason": "Excess creatine is not associated with inhibition of phosphocreatine
resynthesis.",
    "Rectified_Model_Solution_First_Error_Step": "Step 2: Option A suggests that creatine excess inhibits
resynthesis. This is incorrect; creatine supplementation increases the amount of creatine pools in muscle tissue
and increases the amount of phosphocreatine, which improves the ability to resynthesize ATP during intensity
exercise."
```

Figure 2: An example of standard annotation (with the previous items omitted to save space).

Annotation Guideline: Chemistry Subject

1 Background

With the increasingly enhanced capabilities of large language models such as ChatGPT, existing models can break down complex problems and solve them step by step. However, while their solutions may often be seemingly correct, they can contain errors in understanding, calculation, and reasoning logic. In this annotation task, annotators will evaluate model-generated step-by-step solutions to questions in the chemistry subject (including university-level and high-school-level chemistry), and determine whether the model has obtained the correct answer based on the correct understanding and reasoning logic.

2 Annotation Task

Given a question and a step-by-step solution generated by the model, the annotator needs to read the question and model-generated solutions carefully. There will be different solutions from multiple models for one question, and they need to be annotated independently. Each annotation task is divided into four sub-tasks.

2.1 Sub-task 1

The first sub-task is to decide whether the model-generated step-by-step solution is correct. Even if the model's final answer matches the ground-truth answer, the model may arrive at the correct answer through incorrect understanding and reasoning. Therefore, regardless of whether the final answers match the ground-truth choice, annotators should read through each model solution carefully and complete all subtasks based on prudent judgments.

2.2 Sub-task 2

The second sub-task is to identify the first incorrect step in the model-generated solution if the solution is incorrect. Specifically, each step can be interpreted as correct, neutral, or incorrect.

- **Correct** means both the calculation and understanding are correct, and the step is essential for reaching the correct result.
- **Neutral** refers to the current step possibly being a background statement about the question, a factual/formula description, or an exploratory step generated by the model. There are no obvious errors in this step, but whether this step leads to the final correct answer is not obvious.
- **Incorrect** step refers to obvious errors in calculation, understanding, or other possible errors in reasoning, regardless of whether this error affects the final calculation result.

Please note that we only need to find the first 'incorrect' step.

2.3 Sub-task 3

The third sub-task is to describe the reason for the error in the first incorrect step. Annotators should carefully analyze the possible reasons for the error in the incorrect step and provide a rational analysis. Please try to incorporate the logic of the model's solution and analyze the reasons for this error from

a holistic perspective. This error reason may serve as a reference for other annotators or models in the future, so please be as detailed, comprehensive, and clear as possible.

2.4 Sub-task 4

The fourth and final sub-task is to rewrite the first incorrect step. Based on the error reason you provided, correct the incorrect step that you identified. The rewritten step should naturally integrate into the original context of the model solution, with only the incorrect part modified when necessary.

3 Annotation Data Format

At the end of this guideline, Figure 1 shows an annotation example of one model-generated solution for a high-school chemistry question, which includes the following items:

Question, **Options**, **Model_Solution_Steps** and **Ground_Truth_Answer**: They are what annotators should carefully read and analyze.

Model_Solution_Correctness is the first sub-task item to be annotated. It is defined as whether the model solution is completely correct for all steps. Please only answer "Yes" or "No" in this column. Please do not fill in other variants such as correct, incorrect, positive, negative, etc.

Model_Solution_First_Error_Step_Num is the second sub-task item to be annotated. It is defined as the first incorrect step specified in the model solution. If the annotation of the previous item is "No", please enter the numerical step number with quotation marks here, such as "1", "2", etc.; If the result of the first item annotation is "Yes", fill in "None" here. Please note:

1. Do not fill in other variants such as "one", "two", or "none", "no", etc.
2. Step numbers should be written in double quotation marks; do not just write numbers.

Model_Solution_Error_Reason is the third sub-task item to be annotated. Please provide a detailed error analysis for the identified first incorrect step here. For example, whether the model refers to incorrect factual knowledge / correct facts but incorrect comprehension, calculation / incorrect contextual logic links / improper conditions or assumptions, etc. If the annotation of the first item "Model_Solution_Correctness" is "Yes", fill in "None" here.

Rectified_Model_Solution_First_Error_Step is the fourth and last sub-task item to be annotated. Please correct the incorrect step based on the error reason you provided. If the annotation of the first item "Model_Solution_Correctness" is "Yes", fill in "None" here. Note that we are not responsible for the consequential steps following the first incorrect step; we only need to correct the identified first error here.

4 Annotation Regulations & Quality Control

All questions are to be annotated according to the format requirements. If issues arise with the annotation format from sampling inspection, annotators should make modifications promptly.

If the annotations of error reasons and step corrections are too brief, perfunctory, or unclear, re-annotation is required to ensure the annotation quality. For each batch of submitted files, if any of the following occurred, all questions of the same batch will be sent back for re-examination and re-annotation:

1. the annotation error rate is higher than 5% after random sampling, or
2. any of the requirements in this guideline are found violated.

5 Additional Notes for Chemistry Subject

5.1 Look-Up Source

The chemistry subject of this task includes annotating questions in the difficulty level of American university and high school chemistry. When encountering unfamiliar or unclear terms and concepts,

please use search engines and conduct factual inquiries from authoritative sources to assist with your annotation. Please do not use any language models as your annotation assistance tool. Contact the annotation management team whenever in doubt.

6 Common Questions

Q1: When a step repeats the incorrect option should I mark it as the first incorrect step?

No, if a step only describes what an incorrect option is stating or repeats the exact words of the incorrect option without analysis, it does not constitute an incorrect step. But if the step includes wrong analysis/judgment about the incorrect option, then it is considered incorrect.

Q2: If previous steps so far have only stated relevant facts without specific analysis, how should I handle these steps?

Similarly, if the steps focus solely on factual statements about the options or related background descriptions, these steps should be understood as neutral steps unless these steps contain incorrect understanding, incorrect facts, etc. The first incorrect step should be the one that involves incorrect analysis or reasoning.

Q3: What should I do if I encounter difficulty comprehending the question options or model solutions?

When encountering unfamiliar terms, look up textbook sources or use search engines to find reliable explanations first. Contact the annotation management team whenever in doubt.

Q4: The model's answer gives two answers, one of which is correct. How should I decide?

Annotate it as "No." Then follow the standard process to annotate the incorrect step, error reason, and the correction step.

Q5: It's inconvenient to read the JSON file without line breaks.

If you are using platforms like Visual Studio Code to annotate files in JSON format, you can install plugins like 'json editor' to assist in reading. However, take caution against modifying any content other than the required subtasks and do not manually change the data structure.

7 Annotation Example

Figure 1 below is a model solution of a sample question to be annotated; Figure 2 shows an example annotation for the same model solution.

```
{
  "Question_UUID": "5e3ecc23-744f-45a1-b359-4ea9f20d0f08",
  "Subject": "high_school_chemistry",
  "Question": "Which of the following is not a good measure of relative intermolecular attractive forces?",
  "Ground_Truth_Analysis": "",
  "Options": "A: Heat of fusion \nB: Boiling points \nC: Vapor pressures \nD: Heat of vaporization ",
  "Ground_Truth_Answer": "A",
  "Model_Solution_Steps": [
    "Step 1: Heat of fusion is a good measure of the strength of intermolecular forces because it is the energy required
to change a substance from a solid to a liquid state.",
    "Step 2: Boiling points are also a good measure of relative intermolecular attractive forces since substances with
stronger intermolecular forces require more energy to overcome these forces and boil.",
    "Step 3: Vapor pressures are another good measure of relative intermolecular attractive forces because substances
with weaker intermolecular forces have higher vapor pressures.",
    "Step 4: Heat of vaporization is a good measure of the strength of intermolecular forces as it is the energy required
to change a substance from a liquid to a gas state.",
    "Step 5: Considering all the choices, it can be seen that all of them are good measures of relative intermolecular
attractive forces. Therefore, the correct choice is: None."
  ],
  "Sampled_Model": "GPT3.5",
  "Model_Solution_Correctness": "TBD",
  "Model_Solution_First_Error_Step_Num": "TBD",
  "Model_Solution_Error_Reason": "TBD",
  "Rectified_Model_Solution_First_Error_Step": "TBD"
},
```


Figure 1: A task example of a model-generated solution for a high-school chemistry question (to be annotated).


```
    "Model_Solution_Correctness": "incorrect",
    "Model_Solution_First_Error_Step": "1",
    "Model_Solution_Error_Reason": "The heat of fusion is essentially the same as the remaining
three, but it is not as good as them, leaning more towards the energy required to transition from solid
to liquid, making it not a good method compared to the other three options.\n"
    "Model_Solution_Rectified_First_Error_Step": "Step 1: The heat of fusion actually measures the
energy required to disrupt the orderly arrangement of molecules in a crystal structure, rather than
just the strength of intermolecular attractive forces.\n",
```


Figure 2: An example of standard annotation (with the previous items omitted to save space).

Annotation Guideline: Logic Subject

1 Background

With the increasingly enhanced capabilities of large language models such as ChatGPT, existing models can break down complex problems and solve them step by step. However, while their solutions may often be seemingly correct, they can contain errors in understanding, calculation, and reasoning logic. In this annotation task, annotators will evaluate model-generated step-by-step solutions to questions in the logic subject, and determine whether the model has obtained the correct answer based on the correct understanding and reasoning logic.

2 Annotation Task

Given a question and a step-by-step solution generated by the model, the annotator needs to read the question and model-generated solutions carefully. There will be different solutions from multiple models for one question, and they need to be annotated independently. Each annotation task is divided into four sub-tasks.

2.1 Sub-task 1

The first sub-task is to decide whether the model-generated step-by-step solution is correct. Even if the model's final answer matches the ground-truth answer, the model may arrive at the correct answer through incorrect understanding and reasoning. Therefore, regardless of whether the final answers match the ground-truth choice, annotators should read through each model solution carefully and complete all subtasks based on prudent judgments.

2.2 Sub-task 2

The second sub-task is to identify the first incorrect step in the model-generated solution if the solution is incorrect. Specifically, each step can be interpreted as correct, neutral, or incorrect.

- **Correct** means both the calculation and understanding are correct, and the step is essential for reaching the correct result.
- **Neutral** refers to the current step possibly being a background statement about the question, a factual/formula description, or an exploratory step generated by the model. There are no obvious errors in this step, but whether this step leads to the final correct answer is not obvious.
- **Incorrect** step refers to obvious errors in calculation, understanding, or other possible errors in reasoning, regardless of whether this error affects the final calculation result.

Please note that we only need to find the first 'incorrect' step.

2.3 Sub-task 3

The third sub-task is to describe the reason for the error in the first incorrect step. Annotators should carefully analyze the possible reasons for the error in the incorrect step and provide a rational analysis. Please try to incorporate the logic of the model's solution and analyze the reasons for this error from a holistic perspective. This error reason may serve as a reference for other annotators or models in the future, so please be as detailed, comprehensive, and clear as possible.

2.4 Sub-task 4

The fourth and final sub-task is to rewrite the first incorrect step. Based on the error reason you provided, correct the incorrect step that you identified. The rewritten step should naturally integrate into the original context of the model solution, with only the incorrect part modified when necessary.

3 Annotation Data Format

At the end of this guideline, Figure 1 shows an annotation example of one model-generated solution for a logic question, which includes the following items:

Question, **Options**, **Model_Solution_Steps** and **Ground_Truth_Answer**: They are what annotators should carefully read and analyze.

Model_Solution_Correctness is the first sub-task item to be annotated. It is defined as whether the model solution is completely correct for all steps. Please only answer "Yes" or "No" in this column. Please do not fill in other variants such as correct, incorrect, positive, negative, etc.

Model_Solution_First_Error_Step_Num is the second sub-task item to be annotated. It is defined as the first incorrect step specified in the model solution. If the annotation of the previous item is "No", please enter the numerical step number with quotation marks here, such as "1", "2", etc.; If the result of the first item annotation is "Yes", fill in "None" here. Please note:

1. Do not fill in other variants such as "one", "two", or "none", "no", etc.
2. Step numbers should be written in double quotation marks; do not just write numbers.

Model_Solution_Error_Reason is the third sub-task item to be annotated. Please provide a detailed error analysis for the identified first incorrect step here. For example, whether the model refers to incorrect factual knowledge / correct facts but incorrect comprehension, calculation / incorrect contextual logic links / improper conditions or assumptions, etc. If the annotation of the first item “Model_Solution_Correctness” is "Yes", fill in "None" here.

Rectified_Model_Solution_First_Error_Step is the fourth and last sub-task item to be annotated. Please correct the incorrect step based on the error reason you provided. If the annotation of the first item “Model_Solution_Correctness” is "Yes", fill in "None" here. Note that we are not responsible for the consequential steps following the first incorrect step; we only need to correct the identified first error here.

4 Annotation Regulations & Quality Control

All questions are to be annotated according to the format requirements. If issues arise with the annotation format from sampling inspection, annotators should make modifications promptly.

If the annotations of error reasons and step corrections are too brief, perfunctory, or unclear, re-annotation is required to ensure the annotation quality. For each batch of submitted files, if any of the following occurred, all questions of the same batch will be sent back for re-examination and re-annotation:

1. the annotation error rate is higher than 5% after random sampling, or
2. any of the requirements in this guideline are found violated.

5 Additional Notes for Logic Subject

5.1 Look-Up Source

Annotating the logic subject questions requires prudent reasoning and inference abilities, as annotators have been notified and assessed accordingly in the recruitment round. When encountering unfamiliar or unclear terms and concepts, please use search engines and conduct factual inquiries from authoritative sources to assist with your annotation. Please do not use any language models as your annotation assistance tool. Contact the annotation management team whenever in doubt.

6 Common Questions

Q1: When a step repeats the incorrect option should I mark it as the first incorrect step?

No, if a step only describes what an incorrect option is stating or repeats the exact words of the incorrect option without analysis, it does not constitute an incorrect step. But if the step includes wrong analysis/judgment about the incorrect option, then it is considered incorrect.

Q2: If previous steps so far have only stated relevant facts without specific analysis, how should I handle these steps?

Similarly, if the steps focus solely on factual statements about the options or related background descriptions, these steps should be understood as neutral steps unless these steps contain incorrect understanding, incorrect facts, etc. The first incorrect step should be the one that involves incorrect analysis or reasoning.

Q3: What should I do if I encounter difficulty comprehending the question options or model solutions?

When encountering unfamiliar terms, look up textbook sources or use search engines to find reliable explanations first. Contact the annotation management team whenever in doubt.

Q4: The model's answer gives two answers, one of which is correct. How should I decide?

Annotate it as "No." Then follow the standard process to annotate the incorrect step, error reason, and the correction step.

Q5: It's inconvenient to read the JSON file without line breaks.

If you are using platforms like Visual Studio Code to annotate files in JSON format, you can install plugins like 'json editor' to assist in reading. However, take caution against modifying any content other than the required subtasks and do not manually change the data structure.

7 Annotation Example

Figure 1 below is a model solution of a sample question to be annotated; Figure 2 shows an example annotation for the same model solution.

```
{
 "Question_UUID": "f337631a-5aeb-402e-9d5b-f69052da4938",
 "Subject": "logic",
 "Question": "Nutrition education in schools once promoted daily consumption of food from each of the four food groups: milk, meat, fruit and vegetables, and breads and cereals. This recommendation was, however, dangerous to health.Each of the following, if true, provides support for the critique above EXCEPT:",
 "Options": "A: A healthy diet should include the consumption of several different fruits and vegetables daily, but the recommendation was often interpreted as satisfied by the consumption of a single serving of a fruit or vegetable. \nB: The recommendation that some food from the fruit and vegetable group be consumed daily constituted a reminder not to neglect this group, which provides needed vitamins, minerals, and fiber. \nC: The omission of fish, which contains beneficial oils, from the names of groups in the list gave erroneous impression that it is less healthy as a food than is red meat. \nD: The division into four groups gave the impression that an equal amount of each should be consumed, but milk and meat tend to contain fats that promote heart disease and cancer and should be eaten in lesser amounts. ",
 "Ground_Truth_Analysis": "",
 "Ground_Truth_Answer": "B",
 "Ground_Truth_Answer_Modified": "B",
 "Sampled_Model": "Claude2",
 "Model_Solution_Steps": [
   "Step 1: Choice A provides support for the critique that the recommendation was dangerous to health, since interpreting it as allowing consumption of a single fruit or vegetable serving per day would be unhealthy. So Choice A should be eliminated as an answer.",
   "",
   "Step 2: Choice B states that the recommendation served as a useful reminder to consume fruits and vegetables, which provide health benefits. This supports the recommendation rather than critiquing it, so Choice B should also be eliminated.",
   "",
   "Step 3: Choice C provides a critique of the recommendation by noting the omission of fish, which is a beneficial food. This supports the claim that the recommendation was dangerous to health.",
   "",
   "Step 4: Choice D critiques the recommendation by noting that it implied equal consumption of food groups that should be eaten in lesser amounts due to health dangers. This supports the claim that the recommendation was dangerous. ",
   "",
   "Step 5: Choices A, B and D all provide support for the critique of the recommendation, while only Choice C does not.",
   "",
   "Solution: Choice_C"
 ],
 "Model_Solution_Correctness": "TBD",
 "Model_Solution_First_Error_Step": "TBD",
 "Model_Solution_Error_Reason": "TBD",
 "Model_Solution_Rectified_First_Error_Step": "TBD",
}
```


Figure 1: A task example of a model-generated solution for a logic question (to be annotated).


```
 "Model_Solution_Correctness": "incorrect",
 "Model_Solution_First_Error_Step": "2",
 "Model_Solution_Error_Reason": "Option B supports the importance of consuming fruits and vegetables in the diet. This contradicts the claim that this recommendation is dangerous to health."
 "Model_Solution_Rectified_First_Error_Step": "Step 2: Option B supports the part of the recommendation about consuming fruits and vegetables, and does not provide criticism of the entire recommendation. So Option B should be the correct answer."
```


Figure 2: An example of standard annotation (with the previous items omitted to save space).

Annotation Guideline: Coding Subject

1 Introduction

With the capabilities of large language models like ChatGPT gradually increasing, existing models often manage to decompose and partially solve complex problems. However, their explanations may sometimes be misleading or contain errors in understanding, calculation, and logical reasoning. In this annotation task, annotators should review and evaluate the step-by-step solutions provided by the model to determine whether the model has correctly understood and reasoned to obtain the correct answer.

2 Annotation Task Description

Given a question and a step-by-step solution generated by the model, annotators need to carefully read the question and solution, and review the solution. Each question will have multiple different solutions from the model, which need to be reviewed independently.

2.1 Data Format

Each data entry will include:

- A coding question and description (within a block comment in the function).
- A segment of the solution generated by the model.
- A series of assert statements to help us check the generated code.

The comments in the model-generated code solution should indicate the model's thought process. We should regard comments as steps in the thought process, and the code corresponding to each comment as the model's implementation of the comment. We can run the generated code and use assert statements to test and understand the code's logic to complete the following tasks.

2.2 Tasks

- **Task 1: Determine the correctness of the step-by-step thought process.** Even if the model's final answer does not produce an error, it might have arrived at the correct answer through incorrect understanding and reasoning. Therefore, regardless of whether the final answer matches, annotators should carefully read the question and understand the code, using assert statements to identify any errors.
- **Task 2: Identify the first erroneous step.** If the solution is incorrect, annotators need to identify the first erroneous step in the step-by-step solution. Specifically, if there is a logical error in the code, find the first incorrect thought step's comment. If the code logic is correct, but the implementation contains an error, identify the first line of erroneous code. If the model failed to correctly understand the intent of the question, find the first step explaining the thought process. Note that we need to find the first erroneous step.
- **Task 3: Describe the reason for the first error.** Annotators need to analyze the error in the erroneous step and provide a reasonable analysis. Please try to incorporate the model's thought process into the analysis. This error reason will serve as a reference for other annotators or models in the future, so please be detailed, thorough, and easy to understand.

- **Task 4: Rewrite the erroneous step.** Based on the provided error reason, correct the erroneous step. If the thought process is incorrect, modify the erroneous comment. If the code contains an error, modify the corresponding code. Note that we only need to correct the first line of erroneous code. If you have any questions about this step, please contact us.

3 Annotation File and Format Description

Below is a sample annotation for a code answer (in response to a model reply).

- `Model_Solution_Steps` and `assert_statements` are the content that annotators should carefully read, analyze, and run tests on.
- `Model_Solution_Correctness` is the first item to be annotated: please only answer "yes" or "no" in this column. Do not fill in other variants such as correct, incorrect, right, wrong, etc.
- `Model_Solution_First_Error_Step_Num` is the second item to be annotated: if the first item annotation result is "no", then only answer the number of the erroneous code line in this column such as "1", "2". If the first item annotation result is "yes", leave this column blank. Note:
  - Do not fill in other variants such as "one", "two" or "none", "no".
  - The step number should be written within quotation marks "", do not write just the number. (Line numbers can be enabled in the compiler, for example, in VSCode notebook, you can open the command palette and find Show Cell Line Numbers)
  - The line number starts from the first line of the code block. If there is no code block symbol, start counting from the first line of import or def.
- `Model_Solution_Error_Reason` is the third item to be annotated: please fill in this column with a detailed error analysis. Explain in a three-step method: 1. What was the model's original solution trying to do, and where did the problem occur. 2. Give an example of why the model could not solve this step correctly. 3. Provide a potential correct solution thought, what should be done in this step. If the first item annotation result is "yes", leave this column blank.
- `Rectified_Model_Solution_First_Error_Step` is the fourth and last item to be annotated: in this column, rewrite the erroneous step correctly. Try to provide a correct step in the same style as the model so that it can directly replace the original erroneous step. If the first item annotation result is "yes", leave this column blank. Note:
  - We are not responsible for the steps after the first error. We only need to correct the error here.
  - Each question may have multiple correct answers after the first error, we only need to choose one way to modify it.
  - Please ensure the modified code has correct indentation.

4 FAQ

1. **Q: If the code has multiple code blocks, how to locate the line number?**
   - **A:** We have regenerated the model answers and ensured a stricter format, now all code should be within one code block. If you still find such cases, please contact us.
2. **Q: What if the model answer is abnormal (e.g., no code)? What if the question is incomprehensible, or the question itself seems problematic?**
   - **A:** Please write the issue in the feedback sheet below. We will follow up on the problem.
3. **Q: If the generated code lacks library and function, what should be the error line number?**
   - **A:** All are considered errors. The error step should be located below the earliest existing import line, rather than where the inner function/library function is used.

5 Annotation Example

5.1 Items to Read / Run

5.1.1 `Model_Solution_Steps`:

```
def divide_string(s: str) ->bool:
    """
    A string composed entirely of repetitive characters is known as a uniform
                                            string.

    For example, "1111" and "33" are uniform strings. By contrast, "123" is not a
                                            uniform string.

    Rule: Given a numeric string s, divide the string into some uniform substrings,
                                            such that
    there is exactly one uniform substring of length 2 and all other uniform
                                            substrings have a
    length of 3.

    If the string s can be divided according to the rule mentioned above, return
                                            true;
    otherwise, return false.

    Substrings are sequences of contiguous characters within the original string.
    >>> divide_string("000111000")
    false
    >>> divide_string("00011111222")
    true
    """
    # Check if length of s is valid
    if len(s) % 3 !=2:
        return False

    i = 0
    while i <len(s):
        # Find a substring of length 2
        if i <len(s) -2 and s[i] ==s[i+1]:
            substr_len =2
        # Find substrings of length 3
        else:
            substr_len =3

        # Check if current substring is uniform
        substr =s[i:i+substr_len]
        if len(set(substr)) !=1:
            return False

        i += substr_len

    return True
```

The key steps are:

1. Check if length of `s` is valid based on the constraints.
2. Iterate through `s`.
3. At each position, try to extract a substring of length 2 or 3.
4. Check if the substring is uniform.
5. Return `False` if any substring is not uniform.
6. If all substrings are valid, return `True` at the end.

The comments explain the reasoning and logic behind each step.

5.1.2 assert_statements:

```
assert divide_string("000111000") ==False
assert divide_string("00011111222") ==True
assert divide_string("011100022233") ==False
assert divide_string("69966996") ==False
assert divide_string("44433444") ==True
```

5.2 Items to Annotate

5.2.1 Model_Solution_Correctness:

'no'

5.2.2 Model_Solution_First_Error_Step_Num:

'23'

5.2.3 Model_Solution_Error_Reason:

"The model tries to extract substring of length 2 and 3, but the 'if' statement will skip the possibility of finding any uniform substring of length 3. For example, if the input s is '00011', the first uniform substring '000' will be incorrectly split at '00'. One possible solution is to record the length of all uniform substrings and check if their lengths satisfy the requirements. Therefore, at this step the model should find the length of the current uniform substring."

5.2.4 Rectified_Model_Solution_First_Error_Step:

```
\# Find the length of the current uniform substring
```

Three-Round Interaction Prompt Template for Self-Refine

```
Following is a question/solution pair in subject {sol['Subject']}. Your task is
to examine the solutions step by step and determine the solution correctness.
If the solution is incorrect, please further find out the first error step and
explain the error reason.
 . . . . . .

{Generated Response From Evaluated Model}


Review your previous answer and find problems with your answer

{Review Response From Evaluated Model}


Based on the problems you found, improve your answer.

Please follow the desired response format:
 . . . . . .

{Self-Refined Response From Evaluated Model}
```

Figure 8: This is the prompt we used for self-refine experiment, note that three consecutive inference calls are made in order to perform the most basic self correction.

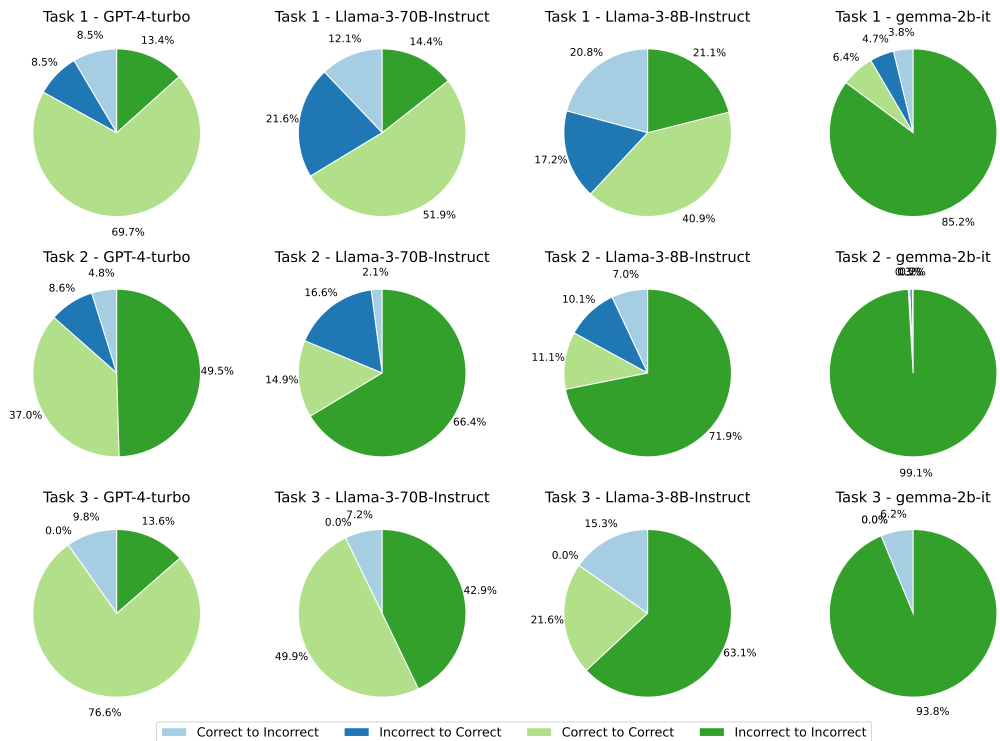

Figure 9: This is the performance breakdown for self-refine experiment in the task level, where task 1,2,3 refers to solution correctness, first error step and error reason determination.

Prompt for Response Generation

```
Following is a question/solution pair in subject {sol['Subject']}. Your task is to examine the solutions step
by step and determine the solution correctness.
If the solution is incorrect, please further find out the first error step and explain the error reason.

Following are the specific definitions of the fields:
Solution Correctness: Does the solution correctly answer the question with justifiable reasoning and selected
the corrected options?
First Error Step: For every step it can either be correct, neutral or incorrect. Correct steps are those that
possess sound logic and correct computation and lead to the correct answer.
Neutral steps are those step that are explanatory, exploring or focusing on background illustration. They
have no obvious mistakes but is not very clear if they lead to the correct answer.
Incorrect steps are those with factual errors, computation errors or understanding/logic errors. These steps
might or might not detour the reasoning path to incorrect answers.
We need to single out the first step that comes with above errors or lead to incorrect answers.
Error Reason: For the identified first error step, please specify the errors made in this step and suggest a
rectified reasoning step instead.

{k_shot_demo}
Below is the question and solution for you to solve:
Question: {sol['Question']}
Options: {sol['Options']}
Step by Step Solution: {sol['Model_Solution_Steps']}

{hint_sent}
Please follow the desired response format:
Solution Analysis: [Give a step by step analysis on the solution correctness here]
Solution Correctness: [Input 'correct'/'incorrect' here to indicate the overall correctness of the solution]
First Error Step: [Input 'Step x' here to indicate the first error step here. Input 'N/A' if the solution is
correct.]
Error Reason: [Input the error reason and the rectified reasoning of the first error step here. Input 'N/A'
if the solution is correct.]

Please follow this format without any additional introductory or concluding statements.
```

Figure 10: This is the prompt template we used to evaluate all the models. The k-shot-demo and hint-sent are either the few shot examples and solution correctness prior or empty string, depending on the experiment setup.

Prompt for Scoring Error Reasons

```
As an experienced {data['Subject']} teacher, your assistance is required to evaluate a
student's explanation regarding the error in a problem solution. The task involves a
detailed understanding of the problem, the incorrect solution provided, and the ground
truth behind the error. Your analysis should focus on whether the student's explanation
aligns with the actual error in the solution.

Please find the details below:

- Question: {data['Question']}
- Incorrect Solution Provided: {data['Model_Solution_Steps']}
- First Incorrect Step in the Solution: {data['Model_Solution_First_Error_Step']}
- Ground Truth Error Reasons: {data['Model_Solution_Error_Reason']}
- Ground Truth Rectified Steps: {data['Model_Solution_Rectified_First_Error_Step']}
- Student's Explanation of the Error: {data['Evaluation_Result']['error_reason']}

Based on this information, please provide the following:

1. Step-by-Step Reasoning: [Offer a succinct, step-by-step interpretation of the ground
truth error reason.]
2. Student Error Reason Analysis: [Analyze the student's explanation step by step,
determining its accuracy in reflecting the actual error briefly.]
3. Final Decision: [State only 'Correct' or 'Wrong' to conclude whether the student's
explanation correctly identifies the error based on your analysis.]

Please follow this format without any additional introductory or concluding statements.
```

Figure 11: This is the prompt template we used to request GPT-4 to help us score the error reasons explained by evaluated models. Note that despite the difficulties of deciding the solution correctness, it is much easier to decide the error reason correctness given the ground truth annotations.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Yes, the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of this work are discussed in § A.1.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: Our work does not involve theoretical assumptions and proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and conclusions of the paper, detailed in § 5.1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We opensourced our evaluation benchmark and the script as described in Section-1. Additionally, we have detailed the experimental setup in the paper (§ 5) , including model selection, hyperparameter settings, data selection, evaluation metrics, hardware resources, etc.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide comprehensive dataset statistics, evaluation metric descriptions, hyperparameters, and tool usage in § 5.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Our proposed method is an inference-only approach for LLM and we adopt the greedy-decoding strategy for all of our experiments, making the experiment results of each session consistent.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer:[Yes]

   Justification: We provide comprehensive experimental setup and hardware computation resources used in § G.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm that the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics, and all the authors preserve anonymity.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The broader impacts of our paper are presented in § A.2.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our dataset focuses on evaluation rather than training models. We leverage existing datasets rather than scrape from the Internet.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the assets, i.e., codes, data and models used in our paper, are properly credited and we explicitly mention and properly respect the license and terms of use.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We have submitted the anonymized dataset, codes, and corresponding documents together with the paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: The full text of instructions given to human annotators is presented in § H.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The justification is as follows: We solely engaged human annotators for the dataset, and they were not subjects of our study. Furthermore, we partnered with a legally recognized annotation company in the country, which has obtained all necessary governmental approvals to operate its annotation business.

    Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.