MR-GSM8K: A Meta-Reasoning Revolution in Large Language Model Evaluation

Anonymous ACL submission

Abstract

In this work, we introduce a novel evaluation paradigm for Large Language Models, one that challenges them to engage in meta-reasoning. This approach addresses critical shortcomings in existing math problem-solving benchmarks, traditionally used to evaluate the cognitive capabilities of agents. Our paradigm shifts the focus from result-oriented assessments, which often overlook the reasoning process, to a more holistic evaluation that effectively differentiates the cognitive capabilities among models. For example, in our benchmark, GPT-4 demonstrates a performance five times better than GPT3.5. The significance of this new paradigm lies in its ability to reveal potential cognitive deficiencies in LLMs that current benchmarks, such as GSM8K, fail to uncover due to their saturation and lack of effective differentiation among varying reasoning abilities. Our comprehensive analysis includes several state-of-the-art math models from both open-source and closed-source communities, uncovering fundamental deficiencies in their training and evaluation approaches.

1 Introduction

Pretrained on trillions of tokens and possessed with billions of parameters, today's large language model (OpenAI, 2023; Anthropic, 2023; Touvron et al., 2023) is capable of generating coherent texts and achieved super-human performances in many tasks (Bubeck et al., 2023; Hendrycks et al., 2021). With the hope of differentiating different model's cognitive ability, math questions are often selected as a proxy evaluation task. However, despite the complexity and diversity of these math problems, recent SOTA LLMs (OpenAI, 2023; Yu et al., 2023; Gou et al., 2023) have been able to achieve accuracy rates exceeding 80% (Luo et al., 2023) on multi-step math reasoning datasets like GSM8K (Cobbe et al., 2021).

Upon a detailed examination of the design principles and objectives of current math datasets, we identified several key shortcomings. Firstly, the majority of these datasets focus on result-oriented metrics, scoring accuracy based solely on the final answer, without considering the underlying reasoning process. With the chain of thought methodology (Wei et al., 2022) and its derivative techniques (Chen et al., 2022; Yao et al., 2023) emerged as the de facto standard for reasoning processes, we argue that the result-driven evaluation method may be insufficient for a comprehensive assessment of the intended cognitive and reasoning capabilities. Secondly, as a recent study (Paster, 2023) suggests that some LLMs who achieved SOTA performances in GSM8K and MATH (Hendrycks et al., 2021) benchmarks have unexpectedly low performance when facing newly released Hungarian high school exams. This raises concerns about the data contamination and potential overfitting to the benchmarks. It also challenges the efficacy of these benchmarks in differentiating model capabilities.

In response to these identified limitations, we introduced a novel meta-reasoning paradigm, namely *challenging LLMs to reason about different reasoning*. Under this paradigm (as illustrated in Figure-2), LLMs (e.g. GPT4 in the figure) are tasked to adopt a role akin to that of a teacher, assessing solutions by determining correctness, identifying potential initial errors, and providing reasons for these errors. Following this design principle, we have developed a novel benchmark named **M**eta-**R**easoning-GSM8k (e.g.**MR-GSM8k**) and proposed a corresponding novel metric called MR-Score. Our benchmark, characterized by instances manually labeled by experts and rigorously reviewed, serves as a robust tool for a qualitative and quantitative assessment of language models. Our findings indicate that most SOTA models demonstrate a significant performance decline in this more nuanced assessment. As demonstrated in Figure-1,

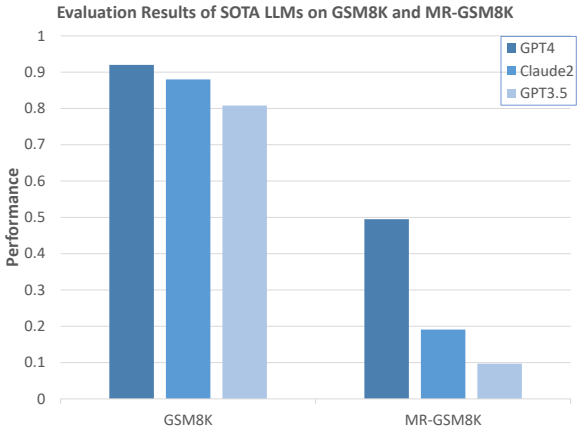

Figure 1: Comparative Performance in GSM8K and MR-GSM8k: This graph illustrates the performance of GPT-4, Claude2, and GPT3.5 in the standard GSM8K benchmark versus our novel MR-GSM8k benchmark. While these models show similar performance levels in GSM8K, a marked variance is observed in MR-GSM8k, highlighting the differentiating power of our benchmark in evaluating deeper reasoning capabilities.

although SOTA models exhibit comparable performance in GSM8K, there is a considerable variance in their effectiveness on our benchmark, with discrepancies up to fivefold.

We argue that our evaluation paradigm does not only provide a more differentiating metric that focuses on the reasoning process over mere computational outcomes, but also sheds light on fundamental deficiencies within the current evaluation and training approaches. As unveiled by our experiments in Section-4, the SOTA math models we evaluated demonstrated a few undesired properties that are otherwise undetected such as sycophancy, overfitting, lack of ontological understanding etc. In Section-5, we will demonstrate that this paradigm provides an effective transformation method that expands any existing evaluation benchmark to be more holistic and differentiating. This is particularly relevant given the non-transparency of the pretraining data of popular LLMs and potential data contamination (Balloccu et al., 2024; Yang et al., 2023).

In conclusion, our paper contributes significantly to the field in the following ways:

- Introduction of a novel evaluation principle, the accompanying open-source benchmark MR-GSM8k and metric MR-Score.
- Demonstration of effective transformation of an existing benchmark (e.g. GSM8K) and how such modification can lead to robust evaluation against potential overfitting and data contamination.
- Comprehensive experiments on several SOTA models using the MR-GSM8k benchmark and critical shortcomings in the current training and evaluation paradigms are highlighted.
- Through analysis on the cognitive levels and examinations of holistic coverage on the solution space, the need for benchmarks that go beyond surface-level evaluations is emphasized, fostering more sophisticated and nuanced AI development.

2 Related Works

Complex reasoning tasks like math problems have long been widely accepted as a great proxy to fathom the cognitive ability in language models (Sharples et al., 1989; Koncel-Kedziorski et al., 2016; Szegedy, 2020; Polu and Sutskever, 2020; Miao et al., 2020; Hendrycks et al., 2021; Cobbe et al., 2021). It demands the ability to understand the symbols and texts behind the problems, to dissect the problems into logically connected subones, then combine and arrange results into final solutions. It touches on the cognitive abilities to induce patterns out of problems, to recall corresponding formulae, apply the rules deductively and reason in abstract symbolic way.

GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) have been two popular benchmarks to evaluate the math reasoning capabilities of LLMs in the past few years. Wei et al., 2022 proposed chain of thought to approach the multi-step reasoning tasks in a step by step manner. Stanford Alpaca (Taori et al., 2023) popularized the knowledge distillation method of cloning corresponding abilities from ChatGPT (OpenAI, 2022) by asking it to generate related QA pairs. WizardMath (Luo et al., 2023) enhanced the distillation by specifying the QA difficulties in the generation process. Mammoth (Yue et al., 2023) combined chain of thought and program of thought, and finetunes its models with the answer generated by GPT-4 (OpenAI, 2023) that are either in natural language or code language. MetaMath (Yu et al., 2023) expanded the generated question types by introducing the forward/backward reasoning variations.

Despite the remarkable progress made in math reasoning, there are some evidence that shows large

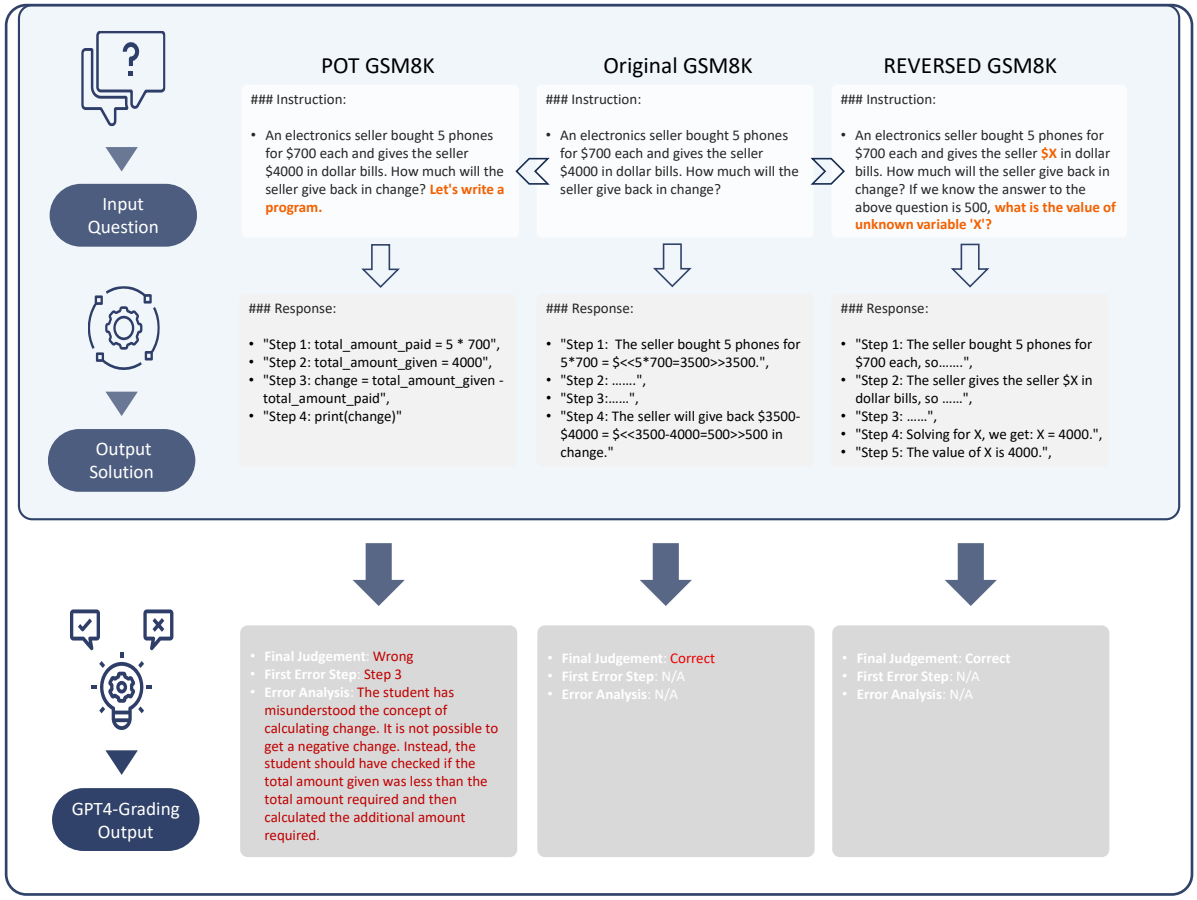

Figure 2: This figure illustrates the structure of the MR-GSM8k benchmark, which incorporates math problems from GSM8K (Cobbe et al., 2021) and introduces two additional variations: Program of Thought (POT) (Yue et al., 2023) and reversed reasoning (Yu et al., 2023). These variations serve as sophisticated proxy challenges to assess whether language models possess the ability to "reason about reasoning". Each question-solution pair within the benchmark demands that the evaluated model not only determines the solution's correctness but also identifies the first erroneous step and articulates the underlying error rationale. Note despite the simplicity of the problem, GPT4 fails to predict the correctness of the solution of original GSM8K problem and its POT variation (highlighted in red in the grading outputs).

language models might not have really mastered the reasoning and even not understanding what it generated. For example, Dziri et al., 2023 evaluated a few compositional reasoning problems and found that LLMs fail to generalize to questions with different complexity than the questions the model was trained on. Arkoudas, 2023 used a collection of 21 reasoning tasks to systematically demonstrated that, despite occasional flashes of analytical brilliance, GPT4 is still severely limited when it comes to reasoning. Huang et al., 2023a and Yen and Hsu, 2023 also found that ChatGPT is very limited in judging the solution correctness of math problems, however our work focuses on the construction of a qualitative and quantitative evaluation framework and focus on the discussion of the evaluation principle and deficiencies of current training paradigm.

3 Evaluation Framework

3.1 Dataset Construction

As illustrated in Figure-2, given a GSM8K question and its solution (e.g. the upper light blue part in the figure), the evaluated model (e.g. the lower white part) is tasked to predict the correctness of the solution. If the solution is incorrect, the model is expected to further locate the first-error-step and elucidate the error-reason.

Note that each test problem is combined with two variations which requires code solution (Yue et al., 2023) and reversed reasoning (Yu et al., 2023). The variations types are chosen specifically due to their significance in expanding the reasoning methodologies in LLMs. The "Program of Thought"(e.g. code solution), a concept proposed by Madaan et al., 2022 and empirically proven ef-

Question Types	Correct	Incorrect	Total
Original	692	726	1418
POT	113	109	222
Reverse	622	738	1360
Total	1427	1573	3000

Table 1: This table presents the composition of the MR-GSM8k benchmark. It categorizes questions into "Original" (based on GSM8K), "POT" (Program of Thought), and "Reverse" types, with counts for both correct and incorrect reasoning processes. Due to the difficulties of labelling coding solutions, the POT types of problems are labelled by the author manually therefore smaller in size.

fective in math reasoning by Yue et al., 2023; Gou et al., 2023, represents a robust reasoning framework. The reasoning process in code language is inherently more structured and hierarchically abstracted. Its reasoning process is also less prone to error types such as calculation error. Reversed reasoning, on the other hand, is recently brought under spotlight with discussions on if language models are able to learn backward relations effectively (Berglund et al., 2023).

For each problem collected, we utilized MetaMath-7B (Yu et al., 2023) with a temperature setting of 1 to generate step by step solutions. A panel of selected annotators were then recruited to review each question-solution pair on its reasoning process and decide the *solution-correctness, first-error-step and error-reason*. Table 1 shows the statistics about the evaluation benchmark we curated. For more details regarding the definitions of the annotation fields, process design and annotation challenges please refer to Appendix-A.

3.2 Evaluation Metric

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 hereby propose a novel metric named **MR-Score** consisting of three sub-metrics.

The first one is the Matthews Correlation Coefficient (a.k.a MCC, Matthews, 1975) for the binary classification of solution-correctness.

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

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.

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

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.

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

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

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

For the weights w_1, w_2 and w_3, they are chosen empirically to be 0.2, 0.3 and 0.5 by considering the difficulties of binary solution-correctness classification task, multi-class first-error-step predictions and free-form error-reason explanations. For extended discussion on the design of MR-Score, please check out Appendix-B.

4 Experiments

In this section, we will give individual analysis on closed-source commercial language models and open-source SOTA math models as they exhibit very different patterns and properties. To demonstrate the difficulties of our benchmark, we also evaluated the performance of a 70B llama2 model finetuned on in-domain training data.

4.1 Commercial LLMs Evaluation

For this study, we specifically evaluated GPT3.5-turbo-0613, Claude2.0, GPT4-0613. Given that these models have demonstrated over 80% accuracy in single-pass settings (Luo et al., 2023), our interest lies in assessing their performance under zero-shot conditions in the MR-GSM8k benchmark.

In line with findings from Orca-2 (Mitra et al., 2023) and other researchers, we recognized that task performance is significantly influenced by system-instruction design. Thus, we conducted empirical experiments with various instruction templates, selecting the most effective one from our validation set for use across all models (see Appendix-D for more details). Another key aspect of our experimental setup involves the sampling temperature. Unlike the greedy sampling method used in the aforementioned single-pass experiments, we found that greedy decoding substantially diminishes generation quality and diagnostic performance in MR-GSM8k. Consequently, after several iterations of testing on our validation set, we set the sampling temperature empirically to be 0.5 for all models.

Table-2 presents the evaluation results for the three selected models. It's evident that GPT3.5 trails behind both Claude2 and GPT4 across most metrics. For instance, in binary correctness prediction, GPT3.5 achieves an MCC score of only 0.198, while Claude2 and GPT4 score 0.345 and 0.614, respectively. Figure-3 offers a visual representation of the incorrect solutions within MR-GSM8k, elucidating the progressively complex nature of the tasks – identifying incorrect solutions, locating the first-error-step, and articulating the error's rationale. The graph reveals a downward trend in accuracy for these successive tasks. Notably, GPT3.5 shows a mere 4.64% success rate in identifying the first-error-step and explaining the reason behind it.

Further scrutiny of False Positive Rate and False Negative Rate made by the models, as illustrated in Figure-4, uncovers distinct tendencies in each model's performance. Claude2 exhibits an impressively balanced distribution between false positives and false negatives. In contrast, GPT4 is prone to committing false negatives at a rate approximately three times higher than false positives, indicating a tendency to label correct solutions as incorrect. Conversely, GPT3.5 displays an opposite trend to GPT4, with a high false positive rate of 60.52%. This suggests that GPT3.5 is more inclined to uncritically accept the given solutions, irrespective of their actual correctness, as compared to the other two models. Possible interpretation could be made by observing the low accuracy of GPT3.5 in determining the error reasons. The overall uncertainty towards a solution combined with the potentially insufficient calibration of RLHF lead to the sycophancy behavior (Sharma et al., 2023).

Despite the fact that all three evaluated models exhibit relatively similar pass@1 rates on the GSM8K dataset (e.g., GPT4 surpassing Claude2 by only 4 percent in accuracy as per Luo et al., 2023), the performance disparity becomes stark in the context of our MR-GSM8k benchmark. GPT4 demonstrates a significantly higher proficiency, being approximately 5 times more effective than GPT3.5 and 2.5 times more so than Claude2 measured by MR-Score. This pronounced variance in evaluation metrics highlights the limitations inherent in the current design of benchmark datasets, reinforcing the argument presented in Section-1 for the necessity of a new paradigm to comprehensively evaluate the reasoning abilities of contemporary LLMs.

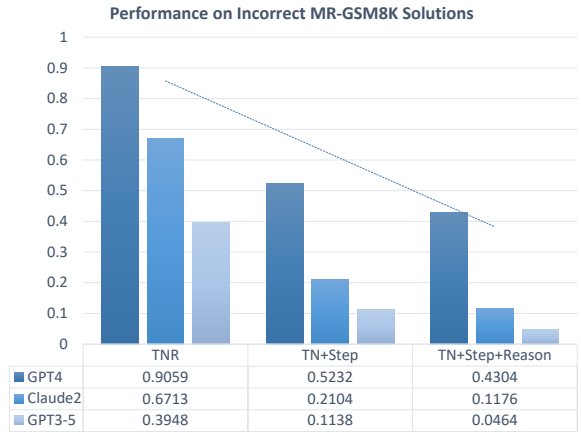

Figure 3: Performance Visualization on Incorrect MR-GSM8k Solutions: This graph depicts the performance of Claude2, GPT3.5, and GPT4 in identifying incorrect solutions in MR-GSM8k. 'TNR' denotes the True Negative Rate. 'TN+Step' refers to the ratio where models correctly identified an incorrect solution and located the first-error-step. 'TN+Step+Reason' represents the frequency of models correctly determining solution-correctness, identifying the first-error-step, and providing an accurate error-reason.

4.2 Open-sourced LLMs Evaluation

In this section, we selected several state-of-the-art open-source models fine-tuned on the 70B llama architecture (Touvron et al., 2023). Namely, they are WizardMath-70B (Luo et al., 2023),

Model	Eval Method	TPR	TNR	MCC	ACC-S	ACC-R	MR-Score
Claude2	0-shot	67.41%	67.13%	0.345	21.04%	11.76%	0.191
GPT3.5	0-shot	78.84%	39.48%	0.198	11.38%	4.64%	0.097
GPT4	0-shot	69.03%	90.59%	0.614	52.32%	43.04%	0.495
WizardMath-70B	3-shot	82.41%	2.73%	-0.250	0.38%	0.06%	0.001
Mammoth-70B	3-shot	98.81%	2.73%	0.055	0.25%	0.06%	0.012
MetaMath-70B	3-shot	91.45%	10.55%	0.034	1.40%	0.38%	0.013
llama2-70B-diag	0-shot	31.74%	73.49%	0.058	20.79%	6.29%	0.105

Table 2: Evaluation Results of Models on MR-GSM8k: This table presents a detailed breakdown of each model's performance, including True Positive Rate (TPR), True Negative Rate (TNR) and Matthews Correlation Coefficient. The ACC-S, ACC-R and MR-Score columns represent the ACC_{step}, ACC_{reason} and $MR\text{-}Score$ metrics defined in Section-3. The MR-Score here is calculated based on the manual labelling of error-reasons. Note that MCC is originally not range between 0 and 1 and MR-Score is inherently absolute, thus both are not expressed in percentage.

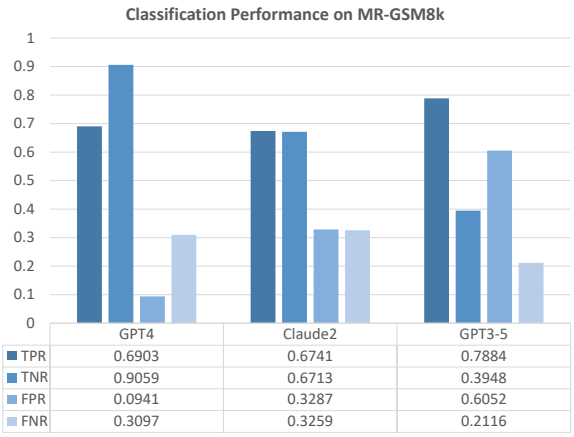

Figure 4: Error Rate Analysis of Closed-Source Models on MR-GSM8k: This figure illustrates the True Positive Rate (TPR), True Negative Rate (TNR), False Positive Rate (FPR), and False Negative Rate (FNR) of closed-source models evaluated on MR-GSM8k, offering insights into their accuracy in various aspects of solution evaluation.

Mammonth-70B (Yue et al., 2023) and MetaMath-70B (Yu et al., 2023). WizardMath-70B underwent training on augmented problems of varying difficulty levels and was further refined through a process-oriented reinforcement learning procedure. Mammoth-70B received training on a diverse range of STEM-related problems, including those from the GSM8K dataset, which were augmented with GPT4-generated code solutions. MetaMath-70B's training encompassed a substantial volume of augmented data, including rephrased and backward-transformed problems from the GSM8K set. Given that these models are specialized and not fine-tuned for general instructions, we employed three-shot in-context learning examples during inference to guide the models in adhering to the desired format and reasoning logic.

As demonstrated in Table-2, despite having exposure to similar training data, all three models failed miserably in the MR-GSM8k benchmark. This outcome highlights their lack of generalization capabilities when faced with problems similar to their training data but presented in different formats. Among these models, MetaMath displayed the most commendable performance, which is not entirely unexpected. Over half of its training dataset (approximately 240k instances) is derived directly from the GSM8K dataset, including a balanced mix of answer augmentations, question rephrasings, and backward transformations of questions. Despite having a training dataset with question types akin to those in MR-GSM8k (e.g., original and POT questions), the Mammoth model failed to demonstrate a deeper meta-understanding of the data, correctly answering only one out of 1573 incorrect solutions. WizardMath, trained using the Proximal Policy Optimization algorithm (Schulman et al., 2017) to optimize the joint reward of instruction adherence and solution process, also underperformed on our benchmark. It is noteworthy that despite given a few in-context examples, these models still occasionally fails to follow the desired format but outputs the special answer format of GSM8K, which is likely caused by overfitting on the evaluated benchmarks.

4.3 In Domain Finetuning

Given the challenges posed by the novel "reason about reasoning" task paradigm, a pertinent question arises: How much can targeted task-specific training data enhance the performance of current state-of-the-art (SOTA) models on this task? To investigate this, we considered augmenting the GSM8K training set with diagnostics data in a sim-

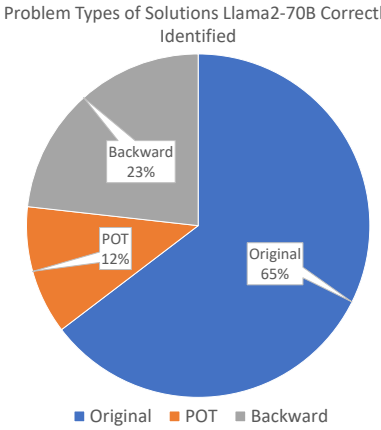

Figure 5: Problem types of incorrect solutions that llama2-70B-MR models has successfully found error step and error reason. Note the training set only includes solutions from original problem.

ilar format. However, due to the labor-intensive nature of manually annotation, we opted for a more feasible approach. Using an expert-designed procedure (as detailed in Appendix-E), we employed GPT-4 to generate the training data based solely on the original GSM8K problems, without incorporating any Program of Thought (POT) or reversed transformations.

For our base model, we utilized llama-2-70B-base, aligning with the approach of other open-source SOTA math models. We merged the GSM8K training set with the GPT-4 generated diagnostic data that composed of 5k incorrect solutions and 4k correct solutions. For fine-tuning, we adopted the Qlora method (Dettmers et al., 2023), maintaining the same hyperparameters as used for MetaMath-70B. The outcomes of this approach are detailed in the last row of Table-2. Remarkably, despite the zero-shot setting and a considerably smaller training set, the fine-tuned llama2 model outperformed all open-source models and even surpassed GPT3.5 in MR-Score.

Notably, the fine-tuned llama2 model demonstrated a tendency distinct from that of GPT3.5 and other open-source models; it was less inclined to accept solutions but to over reject solutions regardless of the correctness. As depicted in Figure-5, of the 99 questions where the model accurately predicted both correctness and the first error step, a significant portion involved questions with POT and reversed reasoning types. This is particularly noteworthy given that the model was trained exclusively on original questions.

Caution is necessary when interpreting the outcomes of in-domain fine-tuning. Although the fine-tuned model achieved results comparable to GPT3.5 post-fine-tuning, it's important to note that the overall number of correct diagnoses for incorrect solutions remains relatively low (e.g. 6.29%). This underscores the challenging nature of our MR-GSM8k benchmark, where effective diagnosis across diverse solution spaces requires a comprehensive understanding of the problem. Consequently, simple fine-tuning strategies may not yield substantial improvements in performance.

5 Discussion

5.1 What Is the Significance of Reason About Reasoning?

In this paper, we have shown that it is not sufficient to unveil the cognitive depth of the evaluated model by only looking at the computation results. What becomes equally important is the validity and logic of the reasoning process employed by the evaluated model. For the evaluated model to successfully diagnose the solution correctness, it is necessary for the model to be able to infer the correct result and also be able to counterfactually reasons along different reasoning paths and actively examines the conditions and assumptions made on different steps. It is unlikely to succeed in this paradigm without a holistic understanding and robust mastery of the underlying concepts. Therefore, the "reason about reasoning" paradigm emerges as a vital meta-evaluative tool.

Another key significance of this paradigm lies in its capability to transform any existing benchmark to be more robust and holistic. As showcased by Balloccu et al., 2024 and Yang et al., 2023, data contamination issues is becoming more and more prevalent while elusive to detect. Other than relying on collecting fresh and unseen new data, our paradigm allows easy modification on existing benchmarks and our experiments on the wide arrays of SOTA LLMs demonstrate its robustness against the potential data contamination issue.

5.2 What Insights does Reason About Reasoning Bring?

As visualized by Figure-6, we observe that recent SOTA models—regardless of whether they are trained on datasets that augment the respective benchmarks in terms of difficulty (Luo et al., 2023), coverage (Lee et al., 2023; Yue et al., 2023), reasoning types (Yu et al., 2023; Yue et al., 2023), so-

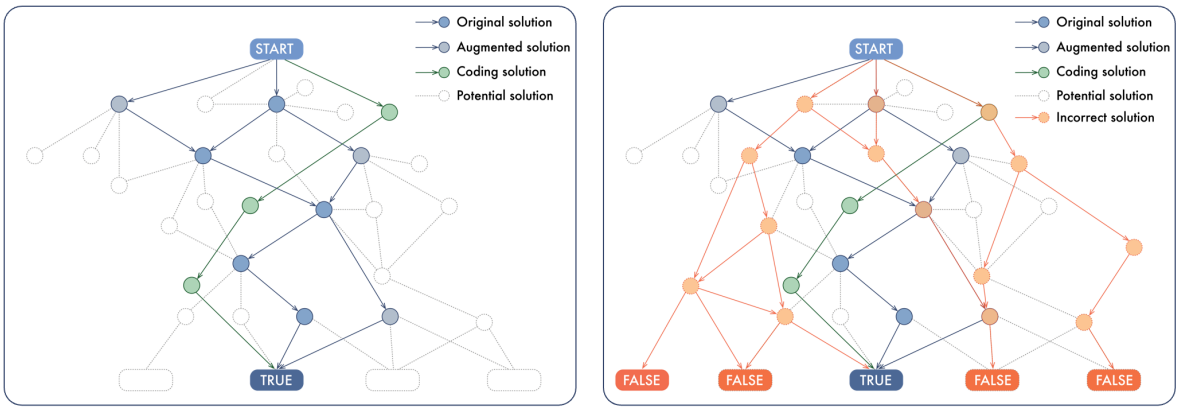

Figure 6: Analysis of Training Sets and Reasoning Path Mastery: The left side of the figure demonstrates the typical inclusion of correct reasoning paths in current math reasoning training sets, despite variations focusing on aspects such as reasoning types (e.g. program of thought) or solution diversity. The right side depicts the limitations of models trained exclusively on correct reasoning paths, showing their inability to assess the validity of alternative reasoning paths for the same problem. This highlights a critical gap in the training paradigm, where models demonstrate basic imitation skills but lack a deep understanding of the underlying logical rules, leading to a superficial grasp of reasoning processes.

lution diversities (Yuan et al., 2023), solution types (Gou et al., 2023; Yue et al., 2023), or through reinforcement learning (Uesato et al., 2022; Luo et al., 2023), are essentially trained by teaching the language models to output a few solution paths within a large search space. Although these models can generate seemingly correct solutions, their grasp of underlying rationale and principles is often superficial and unsophisticated. This is evidenced by issues such as unit inconsistency during calculations (Toh et al., 2023) which highlights a lack of fundamental ontological understanding, and the inability to discern nuanced differences between various reasoning paths for the same problems it was trained on. These core cognitive understanding abilities, vital for benchmarks to assess, have been overlooked due to the prevalent result-oriented metrics and evaluation paradigms. The meta-reasoning paradigm arguably opens up a window for the researchers to examine the "alignment" and "grounding" levels of the evaluated model. In subjects of natural science, we would expect an intelligent agent that aligns epistemically with humans to ground its logic on the rules and principles that closely matches the physical world we live in. For downstream applications, particularly in education and consulting, the critical factor for success lies in the capability to rigorously analyze different plausible solutions, offering a comprehensive and well-rounded exploration of the solution space. The inefficiencies unveiled by our paper poses an unneglectable question on the adoption of current LLMs on downstream applications and urges for a reconsideration on our current training paradigm and its limitations.

6 Conclusion

Throughout this paper, we have delved into the inadequacies of prevalent math reasoning benchmarks and introduced a pioneering evaluation paradigm that compels models to engage in meta-reasoning. Our empirical findings showcase that the benchmark, developed under this novel paradigm, stands out in its ability to differentiate between models and uncover their various deficiencies. This has been particularly evident in the struggles of state-of-the-art language models, which, when confronted with our benchmark, have exposed significant shortcomings inherent in the current training methodologies. These revelations advocate a critical reevaluation of existing training and evaluation practices in the realm of large language models.

In advocating for the widespread adoption of our 'reason about reasoning' evaluation paradigm, we urge researchers to adapt and broaden other reasoning benchmarks in a similar vein. Such transformation is vital not only for a more rigorous assessment of LLMs but also for fostering a deeper and more holistic understanding of these models' capabilities.

7 Limitations

Limitations of the Reason About Reasoning Evaluation Paradigm and MR-GSM8k Dataset

Reflecting on Goodhart's law, which states that 'When a measure becomes a target, it ceases to be a good measure,' it's evident that the 'reason about reasoning' paradigm is not immune to this phenomenon. This paradigm, like any other, can be targeted for optimization, as illustrated in Section-4.3. This is particularly pertinent for static benchmarks where in-domain augmentation and overfitting are feasible. Nonetheless, we contend that our evaluation paradigm presents a greater challenge to overfitting compared to others, owing to its demand for a comprehensive understanding of the problem within a broad error space. While we did not observe significant differences by using solutions sampled from different models, the current MR-GSM8k benchmark is constructed solely from incorrect solutions generated by the MetaMath-7B model. Future versions could incorporate solutions from humans, various models, and even across different languages, enriching its complexity and utility.

Does MR-GSM8k Mandates Human Labelling?

For the sake of rigorousness, every error reason from evaluated models with correct first error step predicted is examined manually in this work. However, this does not imply that MR-GSM8K necessitates a manual labelling for every evaluation. We would like to emphasize that MR-Score is consist of three sub-metrics and error reason is (only) one of the evaluation criteria. Similar to the translation tasks where one expression in one language might corresponds to many variations in another language, it is likewise difficult to come up with an automatic evaluator that scores the error reason perfectly. Nonetheless, this would not undermine the arguments we contend, nor would it affect the cognitive deficiencies unveiled by this metrics. To the best of our knowledge, GPT4 has been the most popular choice for being an automatic evaluator across different metrics (Zheng et al., 2023; Liu et al., 2023). In Appendix-B we empirically demonstrated that GPT4 is able to serve as a decent automatic evaluator that the final MR-Score calculated based on its labelling results is close to that of the manual labelling results.

Does Improvement on MR-GSM8k Necessarily Leads to Improvement on GSM8K?

Models	GSM8k	MR-Score
GPT4	92.0%	0.495
Claude2	88.0%	0.191
llama2-70B-MR	74.3%	0.105
GPT3.5	80.8%	0.097
MetaMath-70B	82.3%	0.013
Mammoth-70B	76.7%	0.012
WizardMath-70B	81.6%	0.001
llama2-70B-GSM8k	74.9%	N/A

Table 3: Comparison of the performances of SOTA models on GSM8K and MR-GSM8k. The GSM8K results are retrieved from corresponding paper (Luo et al., 2023; Yue et al., 2023; Yu et al., 2023). llama2-70B-GSM8K are llama2-70B model finetuned on the training set of GSM8k only. llama2-70B-MR are llama2-70B model finetuned on the GSM8k training set and its meta-reasoning augmentation by GPT4. The MR-Scores are calculated from manual labelling results.

As indicated in Table-3, the Claude2 and GPT4 models, which are stronger in MR-GSM8K, indeed perform better than GPT3.5 in GSM8K. However, despite llama2-70B-MR outperforms GPT3.5 in MR-GSM8k, its accuracy on the GSM8K test-set still trails behind that of GPT3.5. One interpretation is that those closed-source models have been through more sophisticated alignment and comprehensive instruction tuning, but our naive in-domain fine-tuning process may have only enabled our model to replicate diagnostic behaviors (Gudibande et al., 2023). This approach does not seem to enhance the model's fundamental comprehension of mathematical reasoning thus not boosting its test performance. Besides, the overall MR-Score after the finetuning remains relatively low, it would be intriguing to explore how scaling up the diagnostic data or employing more sophisticated training methodologies might alter these outcomes and observe improvements in both benchmarks in the future.

References

Anthropic. 2023. Introducing claude.

Konstantine Arkoudas. 2023. Gpt-4 can't reason. *ArXiv*, abs/2308.03762.

Simone Balloccu, Patr'icia Schmidtov'a, Mateusz Lango, and Ondvrej Duvsek. 2024. Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source llms. In *Annual Meeting of the Association for Computational Linguistics*.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2023. The reversal curse: Llms trained on "a is b" fail to learn "b is a". *ArXiv*, abs/2309.12288.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, John A. Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuan-Fang Li, Scott M. Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *ArXiv*, abs/2303.12712.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2022. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *ArXiv*, abs/2211.12588.

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. 2021. Training verifiers to solve math word problems.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *ArXiv*, abs/2305.14314.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jian, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaïd Harchaoui, and Yejin Choi. 2023. Faith and fate: Limits of transformers on compositionality. *ArXiv*, abs/2305.18654.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Tora: A tool-integrated reasoning agent for mathematical problem solving. *ArXiv*, abs/2309.17452.

Arnav Gudibande, Eric Wallace, Charles Burton Snell, Xinyang Geng, Hao Liu, P. Abbeel, Sergey Levine, and Dawn Song. 2023. The false promise of imitating proprietary llms. *ArXiv*, abs/2305.15717.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023a. Large language models cannot self-correct reasoning yet. *ArXiv*, abs/2310.01798.

Yiming Huang, Zheng-Wen Lin, Xiao Liu, Yeyun Gong, Shuai Lu, Fangyu Lei, Yaobo Liang, Yelong Shen, Chen Lin, Nan Duan, and Weizhu Chen. 2023b. Competition-level problems are effective llm evaluators. *ArXiv*, abs/2312.02143.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. Mawps: A math word problem repository. In *North American Chapter of the Association for Computational Linguistics*.

Ariel N Lee, Cole J Hunter, and Nataniel Ruiz. 2023. Platypus: Quick, cheap, and powerful refinement of llms. *arXiv preprint arXiv:2308.07317*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *ArXiv*, abs/2305.20050.

Xiao Liu, Xuanyu Lei, Sheng-Ping Wang, Yue Huang, Zhuoer Feng, Bosi Wen, Jiale Cheng, Pei Ke, Yifan Xu, Weng Tam, Xiaohan Zhang, Lichao Sun, Hongning Wang, Jing Zhang, Minlie Huang, Yuxiao Dong, and Jie Tang. 2023. Alignbench: Benchmarking chinese alignment of large language models. *ArXiv*, abs/2311.18743.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*.

Aman Madaan, Shuyan Zhou, Uri Alon, Yiming Yang, and Graham Neubig. 2022. Language models of code are few-shot commonsense learners. *ArXiv*, abs/2210.07128.

Brian W. Matthews. 1975. Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et biophysica acta*, 405 2:442–51.

Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A diverse corpus for evaluating and developing english math word problem solvers. In *Annual Meeting of the Association for Computational Linguistics*.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes, Sahaj Agrawal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, Hamid Palangi, Guoqing Zheng, Corby Rosset, Hamed Khanpour, and Ahmed Ali Awadallah. 2023. Orca 2: Teaching small language models how to reason. *ArXiv*, abs/2311.11045.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Keiran Paster. 2023. Testing language models on a held-out high school national finals exam. https://huggingface.co/datasets/keirp/hungarian_national_hs_finals_exam.

Stanislas Polu and Ilya Sutskever. 2020. Generative language modeling for automated theorem proving. *ArXiv*, abs/2009.03393.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *ArXiv*, abs/1707.06347.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Kristjanson Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott Johnston, Shauna Kravec, Tim Maxwell, Sam McCandlish, Kamal Ndousse, Oliver Rausch, Nicholas Schiefer, Da Yan, Miranda Zhang, and Ethan Perez. 2023. Towards understanding sycophancy in language models. *ArXiv*, abs/2310.13548.

Mike Sharples, David C. Hogg, Chris Hutchinson, Steve Torrance, and David J. Young. 1989. Computers and thought: A practical introduction to artificial intelligence. In *Proceedings of the Conference on Innovative Applications of Artificial Intelligence*.

Christian Szegedy. 2020. A promising path towards autoformalization and general artificial intelligence. In *International Conference on Intelligent Computer Mathematics*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca.

Vernon Toh, Ratish Puduppully, and Nancy F. Chen. 2023. Veritymath: Advancing mathematical reasoning by self-verification through unit consistency. *ArXiv*, abs/2311.07172.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, L. Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback. *ArXiv*, abs/2211.14275.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Huai hsin Chi, F. Xia, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *ArXiv*, abs/2201.11903.

Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. 2023. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. *ArXiv*, abs/2307.02477.

Shuo Yang, Wei-Lin Chiang, Lianmin Zheng, Joseph E. Gonzalez, and Ion Stoica. 2023. Rethinking benchmark and contamination for language models with rephrased samples. *ArXiv*, abs/2311.04850.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. *ArXiv*, abs/2305.10601.

An-Zi Yen and Wei-Ling Hsu. 2023. Three questions concerning the use of large language models to facilitate mathematics learning. *ArXiv*, abs/2310.13615.

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

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Chuanqi Tan, and Chang Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2023. Mammoth: Building math generalist models through hybrid instruction tuning. *ArXiv*, abs/2309.05653.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Haotong Zhang, Joseph Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *ArXiv*, abs/2306.05685.

A Annotation Details for MR-GSM8k

In this section we will give a more thorough description of the dataset construction process. We will first present the annotation procedure adopted by our annotators and then describe the challenges faced during such process.

A.1 Annotation Procedure

Given a question and its ground truth solution, the annotator is supposed to first read the question and ground truth solution carefully and make sure he/she comprehend the question thoroughly. Then the annotator is tasked to label the following fields sequentially:

Solution Correctness: Solutions that yield a final output differing from the established ground truth are automatically marked as incorrect. However, in cases where the solution's final output aligns with the ground truth, annotators are tasked with reviewing the entire reasoning path. Their objective is to ascertain whether the correct output is the result of a logical and sensible reasoning process.

First Error Step: This attribute is applicable for solutions with either an unmatched final output or a matched final output underpinned by flawed reasoning. Annotators identify the initial step in the reasoning process where the logic deviates from correctness. In line with the approach of Lightman et al., 2023, we dissected GSM8K solutions into discrete steps, each marked by a newline character, and indexed them accordingly. Each step is then categorized as positive, neutral, or negative. Positive and neutral steps represent stages in the reasoning process where the correct final output remains attainable, whereas negative steps indicate a divergence from the path leading to the correct solution.

Error Analysis: Beyond identifying the first erroneous step, annotators are also responsible for conducting an in-depth analysis of the reasoning that led to the error. This involves an examination of the solution's reasoning flow, focusing on the cause of the initial error and what the correct line of reasoning should have been at that juncture. This error analysis is subsequently compared against the reasoning errors identified by the evaluated models during testing, to assess their accuracy and validity.

A.2 Annotation Challenges

The annotation task turns out to be more difficult than we expect at the beginning. The challenge comes from several sources: First, the language barrier hinders the non-native speaker to understand the meaning of the question. Second, the labelling task requires the annotator to read the question, the ground truth solution and the model solution before judging the correctness of the reasoning process. If the reasoning process is problematic, the annotator is further required to find out the first error step and reason about why the model made such mistake. To reason along the path of the model solution and figure out why the error occurred in the perspective of the model and verbalize all the above in the error reason is quite time-consuming and an exhausting task. Third, some questions are ambiguous in its wording and allows for multiple different interpretations. The backward reasoning transformation of the original problem exacerbated this problem.

To help lower the difficulty of labelling, we translated the problems and solutions into Chinese with the help from ChatGPT (OpenAI, 2022). We noticed that ChatGPT occasionally would made some translation errors such as missing critical information and misinterpret the original text, we therefore enclosed both the original problem-solution pairs and the translated ones for reference when in doubt (See Figure-8 for a full example). The solution correctness, first error step and error analysis are then collected from the annotated dictionary and translated back to English.

A.3 Annotation Quality Control

The annotators are selected based on their labelling performance on a balanced small hold-out problems set consist of 50 questions. For every problems in the Mr-GSM8K, they have been through several examinations before used in the evaluation process: First, every question is labelled twice by different annotators. Inconsistent questions will be singled out to be labelled by the quality control labeller. Besides the double annotation, 50 percent of the annotated problems are sampled out for quality control in the second round of verification. During the evaluation process, all the questions that has a matching correctness prediction and error steps are manually examined by the authors of this work for its error reasons. Questions with incorrect error steps or reasons are cleaned up in this final stage.

Models	Step	Step+Reason/M	Step+Reason/A	MR-Score/M	MR-Score/A
GPT4	823/1573	677/1573	732/1573	0.495	0.512
Claude2	331/1573	185/1573	224/1573	0.191	0.203
llama2-70B-MR	327/1573	99/1573	139/1573	0.105	0.118
GPT3.5	179/1573	73/1573	73/1573	0.097	0.097
MetaMath-70B	22/1573	6/1573	7/1573	0.013	0.013
Mammoth-70B	4/1573	1/1573	2/1573	0.012	0.012
WizardMath-70B	6/1573	1/1573	1/1573	0.001	0.001

Table 4: Comparison of the manual labelling results and GPT4-Turbo-1106 labelling results. Step column shows the number that each evaluated models successfully located the first error steps among incorrect solutions. Step+Reason/M stands for the manual labelling results of the error reasons where its first error step is correct. Step+Reason/A corresponds to the labelling results of GPT4-Turbo-1106. llama2-70B-MR are llama2-70B model finetuned on the GSM8k training set and its meta-reasoning augmentation by GPT4.

B Design thinking of MR-Score

The MR-Score is consist of three sub-metrics corresponds to the three sequential reasoning sub-tasks. For the first solution correctness prediction, we empirically noticed that most of the evaluated language models tend to blindly classify the given solution as correct, exemplified by the low true-negative rate in Table-2. Therefore, we chose the MCC score instead of metrics like F1 or Balanced-Accuracy due to its value range. The models that have high true-positive rate but low true negative rate will have near zero score under the MCC metric. For the second and third tasks of locating first error step and elucidating error reason, we chose the simple accuracy metric. One of the reason is that locating the first error step is a multi-class classification problem and it is difficult to have large prediction bias while at the same time scores high accuracy. Similarly, the explaining error reason task is a free-form generation task that requires substantial understanding and a simple accuracy metric is enough to categorize the model behavior.

As to the weights given to the three metrics, they are crafted by considering the task complexity and the difference between manual labelling and auto labelling results of the error reason. As discussed in Section-5, we chose the GPT4-Turbo-1106 as our proxy evaluator and Table-4 is the results of auto-labelling vs our expert manual-labelling. It is clear that the final MR-Score calculated from manual labelling VS auto labelling are very close to each other, exhibiting the potential of GPT4 to serve as a delegate evaluator for our task.

Table-5 displays the confusion matrix based on GPT4's labelling of all the error reasons. Notably, GPT4 is able to achieve 82% of overall accuracy

	Pos	Neg
Pred-Pos	960/1042	218/626
Pred-Neg	82/1042	408/626

Table 5: This confusion matrix represents the accuracy of GPT4-Turbo-1106 in assessing 1668 incorrect solutions that were correctly identified with the right error step. The task for GPT4-Turbo-1106 was to evaluate the correctness of the error reason provided by the evaluated model, in comparison with the actual ground truth labelled by expert. 'Pos' and 'Neg' represent the ground truth correctness of the provided explanation, while 'Pred-Pos' and 'Pred-Neg' indicate GPT4's prediction about the correctness.

despite a substantial higher false positive rate than false negative rate. However, we still encourage large tech companies, who have the resources to bear manual labelling costs, to release open-source manual labelling results when publishing findings using MR-GSM8k for the best of rigorousness.

C Implications of In-Domain Fine-Tuning

As highlighted in Section-1, the prevalent training approach for complex reasoning tasks typically adopts an inductive learning framework. Here, language models are expected to mimic a vast array of examples, discern patterns, and ideally, generalize robustly to new problems. With the growing acceptance of the scaling law, there's been a belief within the research community that increasing

data scale and model size could lead to heightened intelligence in models. The GPT-4 evaluation report even suggested that GPT-4 might have sparked a semblance of general artificial intelligence (Bubeck et al., 2023). However, GPT-4's notable shortcomings in out-of-distribution tasks (Huang et al., 2023b; Wu et al., 2023; Arkoudas, 2023; Dziri et al., 2023) cast doubt on the emergence of such intelligence through inductive means alone. Evidently, GPT-4 struggles with deductive reasoning tasks that require applying rules or principles it hasn't explicitly learned.

The implications of performance improvements observed post in-domain fine-tuning are manifold. Firstly, the potential for data contamination necessitates careful consideration, along with timely updates to benchmarks. Secondly, the limitations of the current training paradigm underscore the significance of data quality. For instance, in educational applications of LLMs, while human-level understanding might be lacking, a model trained on a substantial corpus of student errors and correct solutions could be sufficiently adept for grading and instructional purposes. Essentially, the model could present an illusion of comprehensive understanding and appear intelligent if the training data closely mirrors real-world use cases. Thirdly, the relative failure of mere scaling prompts us to reconsider the hype around it and reflect on the need for more fundamental changes in the training paradigm.

D Prompts for Zero-Shot Scoring

Since the closed-source models such as ChatGPT and Claude have all been through large amount of diverse instruction tuning and human alignment, we assume that they should be able to follow the instruction and desired format in a zero-shot manner. Figure-7 is the prompt we used for ChatGPT, Claude and GPT4 and in-domain-finetuned llama2-70B in zero-shot setting.

The key insight here is that even though Chain-of-thought helps improving the diagnostic reasoning abilities, the expert tailored diagnosing procedure plays a crucial role. For instance, in the prompt we chose, we did not explicitly ask the model to solve the question first before diagnose because when models were prompted to solve the questions before contrasting their solutions with the provided answers, we observed a bias towards the provided solutions, adversely affecting diagnostic accuracy.

E In Domain Training Data Generation

As mentioned in Section-4, we created in-domain training data to see if in-domain finetune can boost the performance in the MR-GSM8k benchmark. This process involved presenting GPT-4 with a question and its correct solution, then instructing it to introduce an error at a randomly chosen step and complete the solution accordingly. The step-by-step analysis was subsequently generated, focusing on the fabricated error. Despite GPT-4's modest 40% accuracy in correctly identifying incorrect solutions in the test set, this procedure successfully generated accurate diagnostic training data with over 90 percent accuracy, as verified by expert annotation on a sub-sampled set. This is largely due to the expert designed procedure (Figure-9) that greatly lowered the difficulties for instruction following. Note, occasionally GPT4 will fail to fabricate a valid error due to the lack of true understanding of errors (e.g. switch the fraction 8/3 to 2 and 2/3 then claims this is an error).

F More Examples of Evaluated Models on MR-GSM8k

While GPT4 shows a dominant performance in comparison to the other models, it's important to note instances where either GPT3.5 or Claude2 successfully identify errors or provide plausible explanations but GPT4 does not. This observation suggests that even the most advanced models have limitations and specific scenarios where their performance can be outmatched.

Followings are more examples of evaluated models on MR-GSM8k. Despite the low accuracy and naive errors most of the time, they occasionally displayed certain level of ingenuity that matches their problem solving abilities. We will attach both the positive and negative examples to help spread a balanced view on both their brilliance and their simple mistakes.

Figure-10 contrasts GPT3.5 with GPT4 on a simple question where GPT4 failed to correctly diagnose while GPT3.5 did. Figure-11 provides an incorrect diagnosis example from GPT4. Figure-12 shows a correct example from MetaMath.

Prompt for Zero-Shot Scoring

```
Act as a grade school math teacher and score the following problem solution.

Question:
{data['question']}

Student Solution:
{sol_steps}

Your task involves three parts:
1. **Step-by-step Evaluation:** Go through the student solution carefully and identify
key errors and potential misunderstandings that led to the incorrect solution.
2. **Final Judgement:**  Provide an overall judgement on the correctness of the
student's solution.
3. **First Error Step:** If the solution is incorrect, generate the step number where
the first error occurs, otherwise generate N/A here
4. **Error Analysis:** If the solution is incorrect, analyse the cause and reasons for
the first error step, otherwise generate N/A here

Here's the format I want:
Step-by-step Evaluation: [Provide a step by step examination of the student solution and
identify key errors and misunderstandings here.]
Final Judgement: [Insert only **correct** or **wrong** here]
First Error Step: [Insert either N/A or the step number where the first error occurs]
Error Analysis: [Insert either N/A or the analysis of error in the first error among
solution steps]

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

Figure 7: This is the prompt we used for ChatGPT, Claude and GPT-4 under the zero-shot setting.

Sample dictionary passed to annotators


```
{
    "uuid": "3f3b593b-53e1-4808-afef-9453ba4eb6ab",
    "问题": "一个电子产品销售商以每部手机700美元的价格购买了5部手机，并给了销售商4000美元的纸币。销售商需要找回多少零钱
",
    "标准答案": "销售商以700美元 * 5 = 3500美元购买了5部手机。\n所以销售商找回的零钱为4000美元 - 3500美元 = 500美元。
\n#### 500",
    "待标注的解题过程": [
      "步骤1: 销售商以每部手机700美元的价格购买了5部手机，所以手机的总成本为5 * 700美元 = 3500美元。",
      "步骤2: 销售商给了销售商4000美元的纸币。",
      "步骤3: 为了找零，我们需要从销售商给出的金额中减去手机的总成本。",
      "步骤4: 所以，找零金额为4000美元 - 3500美元 = 500美元。",
      "步骤5: 销售商将找回500美元的零钱。",
      "步骤6: #### 500",
      "步骤7: 答案是: 500"
    ],
    "待标注的最终答案正确性": "对",
    "待标注的解题过程正确性": "待标注",
    "待标注的解题过程第一个错误步骤数": "待标注",
    "待标注的解题过程第一个错误的错误原因": "待标注",
    "英文原题": "An electronics seller bought 5 phones for $700 each and gives the seller $4000 in dollar
bills. How much will the seller give back in change?",
    "标准答案英文原文": "The seller bought the 5 phones for $700 * 5 = $<<700*5=3500>>3500.\nSo the seller
gives back $4000-$3500 = $<<4000-3500=500>>500.\n#### 500",
    "解题过程英文原文": [
      "Step 1: The seller bought 5 phones for $700 each, so the total cost of the phones is 5 * $700 = $3500.",
      "Step 2: The seller gives the seller $4000 in dollar bills.",
      "Step 3: To find the change, we need to subtract the total cost of the phones from the amount of money
given by the seller.",
      "Step 4: So, the change is $4000 - $3500 = $500.",
      "Step 5: The seller will give back $500 in change.",
      "Step 6: #### 500",
      "Step 7: The answer is: 500"
    ]
  }
```


Figure 8: To lower the requirements and difficulties for labelling, we used ChatGPT to translate the MR-GSM8k problems into Chinese. In case for the potential translation errors (missing information and wrong translation) we included the original English text for reference when in doubt.

Prompt for creating DiagGSM8K training data by GPT4

Hi ChatGPT, I need your assistance in creating a math dataset that focuses on annotated incorrect solutions. The dataset is intended for grade school math.

Provided Information:
Question: {question}
Correct Solution Steps: {steps}

Procedure:
Understand the Question and Solution: Ensure you comprehend the question and the correct solution.
Fabricate Error Step and Reason: Randomly select a step and come up with an error that is both realistic and commonly observed among students such as calculation mistake, conceptual misunderstanding, or incorrect formula application etc.
Develop the Incorrect Solution: Base on the error step and error reason, rewrite the solution, ensuring they are internally consistent but incorrect due to the initial error. **Do not mention anything about incorrectness or deliberate error in the error step in your solution**.
Independent Analysis: Provide a step-by-step analysis of the solution, including the error, as if encountering the problem for the first time. Do not reference the fact that the error was intentionally created.
Summarize the Error: Based on your analysis, summarize the error reason for the first incorrect step.

Format:
Error Step: [Identify a step number where the first error in the correct solution should occur. Any step from either the beginning, middle or the end will work.]
Error Reason: [Describe the nature of the error]
Incorrect Solution: [Present the solution, starting correctly but deviating post error step. DO NOT MENTION ANYTHING ABOUT THE INCORRECTNESS.]
Independent Step-by-Step Analysis: [Analyse the solution independently, as if unaware of the intentional error]
Independent Error Reason: [Conclude with a summary of the perceived error reason]

Please adhere strictly to this format, avoiding any additional introductory or concluding remarks.

Figure 9: This is the prompt we used for GPT4 to create in-domain training data.

Sample of GPT3-5/GPT4 evaluated results

"question": Trinity sells magazines at 11/8 of the price she bought the magazines. If she bought the magazines at $72, what is her profit?

"model_output_steps":[
"Step 1: If Trinity bought the magazines at $72, then she sells them at 11/8 * $72 = $90",
"Step 2: Her profit is the selling price minus the buying price, so her profit is $90 - $72 = $18",
"Step 3: #### 18"
]

Final Judgement by GPT3-5: wrong
First Error Step by GPT3-5 : Step 1
Error Analysis by GPT3-5 : The student made an error in Step 1 by incorrectly calculating the selling price as $90 instead of $99. This error likely occurred due to a miscalculation or misunderstanding of multiplying fractions.

Final Judgement by GPT4: Correct
First Error Step by GPT4: N/A
Error Analysis by by GPT4: N/A

Figure 10: Example of an easy problem where GPT4 fails to capture the calculation error but GPT3.5, with much less diagnosis ability, successfully captures the calculation error.

Sample of evaluated result of GPT4 from DiagGSM8K

```
"question": "Brett is 14 years old. In four years his sister Angela will be three
times as old as he is now. How old is Angela right now?"
```

```
"model_output_steps":[
      "Step 1: In four years, Brett will be 14 + 4 = 18 years old.",
      "Step 2: At that time, Angela will be three times as old as Brett is now, so
Angela will be 3 * 14 = 42 years old.",
      "Step 3: Therefore, Angela is currently 42 - 4 = 38 years old.",
      "Step 4: #### 38"
]
```

```
Final Judgement: wrong

First Error Step: Step 2

Error Analysis: The student misunderstood the problem. They were supposed to
calculate Angela's age as three times Brett's age four years from now, not Brett's
current age. As a result, they multiplied the wrong number by 3.
```

Figure 11: Example of question where GPT4 got confused on the problem and claimed an error in the correct solution with some hallucinated reason.

Sample of MetaMath-70B evaluated results

```
"question": "Maggie's oven is malfunctioning. When she sets it to 450 the actual
temperature is 468. If it's off by the same percentage for any recipe, what
temperature should she set it at if her recipe calls for 520 degrees?"
```

```
"model_output_steps":[
      "Step 1:  First, we need to find out how many degrees off the oven is. We do
this by subtracting the actual temperature from the temperature it should be: 520
degrees - 468 degrees = <<520-468=52>>52 degrees.",
      "Step 2: Now that we know the oven is off by 52 degrees, we divide that
amount by the percentage of the oven that's off to find out what the actual
temperature should be: 52 degrees / .052 = <<52/.052=1000>>1000 degrees.",
      "Step 3: #### 1000"
]
```

```
Final Judgement: wrong

First Error Step: Step 1

Error Analysis: The student made a mistake in the first step by subtracting the
actual temperature from the temperature it should be. The correct approach would
be to find the percentage of the oven that's off.
```

Figure 12: Example of MetaMath-70B successfully diagnosed the correctness of a solution from MR-GSM8k and outputted sensible error step and reason.