Can LLMs Learn from Previous Mistakes? Investigating LLMs' Errors to Boost for Reasoning

Anonymous ACL submission

Abstract

Large language models (LLMs) have demonstrated striking reasoning capability. Recent works have shown the benefits to LLMs from fine-tuning golden-standard Chain-of-Thought (CoT) rationales or using them as correct examples in few-shot prompting. While humans can indeed imitate correct examples, learning from our mistakes is another vital aspect of human cognition. Hence, a question naturally arises: *can LLMs learn and benefit from their mistakes, especially for their reasoning?* This study investigates this problem from both the prompting and model-tuning perspectives. We begin by introducing COTERRORSET, a new benchmark with 609,432 questions, each designed with both correct and error references, and demonstrating the types and reasons for making such mistakes. To explore the effectiveness of those mistakes, we design two methods: (1) **Self-rethinking** prompting guides LLMs to rethink whether they have made similar previous mistakes; and (2) **Mistake tuning** involves finetuning models in both correct and incorrect reasoning domains, rather than only tuning models to learn ground truth in traditional methodology. We conduct a series of experiments to prove LLMs can obtain benefits from mistakes in both directions. Our two methods offer potentially cost-effective strategies by leveraging errors to enhance reasoning capabilities, which costs significantly less than creating meticulously hand-crafted golden references. We ultimately make a thorough analysis of the reasons behind LLMs' errors, which provides directions that future research needs to overcome. COTERRORSET will be published soon on `Anonymity Link`.

1 Introduction

Large language models (LLMs) (Brown et al., 2020; Zhang et al., 2022; Anil et al., 2023; Touvron et al., 2023) have demonstrated strong capabilities across various tasks and applications (Liang et al., 2022; Chang et al., 2023). To further unleash the reasoning abilities of LLMs and align their thinking process with humans, many recent studies explored Chain-of-Thought (CoT)-based prompting (Wei et al., 2022; Wang et al., 2022; Li et al., 2023a; Tong et al., 2023; Yao et al., 2023; Besta et al., 2023) to instruct LLMs to solve the given problem with human-like logic. Besides logical step-by-step thinking, another critical learning pattern of us humans is to rethink and learn from our previous mistakes so that avoid repeating the same mistakes in the future (Mercer, 2008; Reich et al., 2023). However, few studies have focused on systematically understanding what kinds of intermediate errors occur in making CoT procedures and whether LLMs can learn from those mistakes. To address these issues, we aim to explore the potential of LLMs to effectively utilize their previous mistakes to boost reasoning.

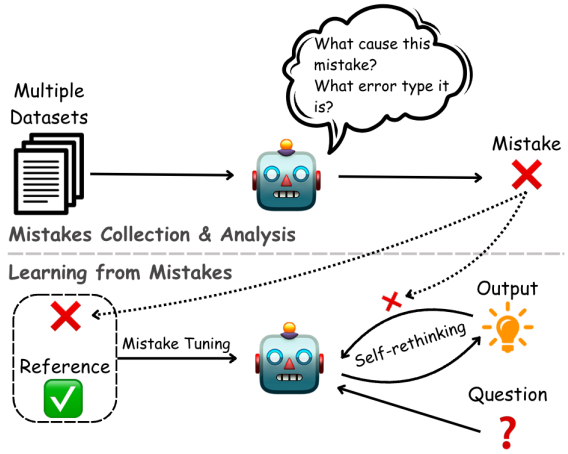


Figure 1: The overview pipeline of our work includes (1). Mistake collection and analysis (Section 3). (2) Two novel methods to instruct LLMs to learn from mistakes(Section 4 and Section 5).

To enhance the scalability and efficiency of analyzing and learning from the mistakes of LLMs, we

began by collecting a vast dataset of LLMs' reasoning outputs and built CoTErrorSet, which consists of 609,432 questions sourced from 1060 tasks across diverse domains. Each query in this set is meticulously structured, featuring both a manually curated correct reference and the incorrect rationales collected from PaLM2 (Anil et al., 2023)'s responses. Furthermore, we prompt the LLMs with the correct reference and the incorrect responses in order to make it reflect why making such mistakes. The introspective responses are also collected and subsequently utilized in our work. We employ this data for cluster analysis to identify specific details of the errors.

With our CoTErrorSet, we introduce two innovative paradigms, namely **mistake tuning** and **self-rethinking**, aimed at efficiently augmenting LLMs by leveraging their historical errors during both tuning and inference stages. Diverging from the conventional approach of only relying on correct rationales in traditional supervised fine-tuning, our mistake tuning strategy incorporates combinations of both correct references and incorrect rationales. To facilitate the learning process for LLMs, we introduce the prefixes *[CORRECT RATIONALE]* and *[INCORRECT RATIONALE]* before the corresponding rationales. Intuitively, this prompt tuning facilitates LLMs to distinguish between correct and incorrect rationales while avoiding corruption from the incorrect ones with the two separated prefixes. For self-rethinking, inspired by contrastive in-context learning (Gao and Das, 2024), we expose LLMs to both correct and incorrect rationales in demonstration samples. After obtaining the initial answer output by the LLM, we iteratively prompt it to rethink and rectify the result based on the historical mistakes. To manage computational resources and prevent potential loops, we implement a threshold, limiting the number of times the model can engage in self-rethinking and corrections. Figure 1 gives an overview pipeline of our work.

To substantiate the efficacy of our proposed methodologies and to delve into the learning capabilities of LLMs from their mistakes, we undertake experiments encompassing diverse reasoning tasks and LLMs of varying sizes. The application of our methods consistently yields performance enhancements across a spectrum of tasks, underscoring the effectiveness and broad applicability of our approaches in leveraging LLMs' mistakes during both the tuning and inference stages. Additionally, we conduct thorough analyses of the error types exhibited by LLMs, offering comprehensive insights and guidance on mitigating the most prevalent errors in these models.

In general, our contributions are as follows:

- A large-scale error set, CoTErrorSet, is constructed for scalable analysis and learning from the LLMs' mistakes.
- We novelly designed two paradigms for LLMs to utilize and learn from their previous mistakes at both fine-tuning and inference stages.
- With extensive experiments, we validate the effectiveness of our proposed methods and provide further hints based on analysis of LLMs' error types.

2 Related Work

Human-like Reasoning with LLMs. CoT (Wei et al., 2022) demonstrate the great potential of equipping LLMs with human-like reasoning capability. Following them, various logical and structural reasoning strategies (Wang et al., 2022; Zhou et al., 2022; Creswell and Shanahan, 2022; Besta et al., 2023; Li et al., 2023b; Lightman et al., 2023) are proposed to align LLMs' thinking processes with humans. These enhanced reasoning approaches have been adopted in different tasks and areas, including commonsense reasoning (Geva et al., 2021; Ahn et al., 2022), logical reasoning (Pan et al., 2023; Lei et al., 2023) and mathematical reasoning (Cobbe et al., 2021; Hendrycks et al., 2021) and achieved promising performance. In this work, we aim to investigate whether LLMs can benefit from rethinking and learning from previous mistakes, which is one of the most important learning patterns of humans.

Refined Reasoning Errors. Several studies focus on adjusting their reasoning pathways to arrive at better solutions. Huang et al. (2022) introduce self-improve that employs CoT plus self-consistency to obtain high-confidence solutions on a large set of unlabeled questions. The self-generated content is then used for fine-tuning in subsequent iterations, thereby further augmenting its reasoning capabilities. Madaan et al. (2023) propose a self-refine technique that encourages LLMs to autonomously correct their outputs without the need for external data or feedback. However, it has been argued by some researchers that LLMs face challenges in self-correcting their responses in

the absence of external feedback, and under certain conditions, such attempts might even deteriorate their performance (Huang et al., 2023). Based on that, An et al. (2023) suggest fine-tuning LLMs using pairs consisting of errors and their respective corrections generated by GPT-4 as a supervisory mechanism. Nevertheless, our work is pioneering in highlighting the impact of exposing mistake examples on in-context learning. Furthermore, our experiments reveal that in the process of model tuning, learning from mistakes can inherently enhance itself by merely being exposed to correct examples and errors, without depending on explicit corrections from teacher models.

3 A Novel Dataset: CoTErrorSet

3.1 Dataset Construction

In order to investigate whether incorrect rationales can also contribute to LLMs' reasoning performance, we introduce CoTErrorSet, a novel benchmark based on the source of CoT-Collection (Kim et al., 2023), built upon various domains, including multiple-choice QA, extractive QA, closed-book QA, formal logic, natural language inference, and arithmetic reasoning. Our dataset's question and reference are obtained from the following datasets: QASC (Khot et al., 2020), AQuA (Ling et al., 2017), GSM8K (Cobbe et al., 2021), QED (Lamm et al., 2021), StrategyQA (Geva et al., 2021), SenseMaking (Wang et al., 2019), CREAK (Onoe et al., 2021), e-SNLI (Camburu et al., 2018) and ECQA (Aggarwal et al., 2021). Each task within this collection is systematically organized to include a question and a correct reference, followed by an incorrect response and the demonstrations why making such mistakes. The errors and demonstrations are both generated from PaLM2.

CoTErrorSet diverges from traditional CoT datasets by employing PaLM2's mistakes and the reasons behind them. We utilized PaLM2 to generate rationales for each question in the dataset, focusing specifically on collecting incorrect rationales. Then we provide PaLM2 with both correct references and its incorrect answers to demonstrate and reflect why it makes such mistakes. The steps of the construction process are shown in Figure 2. This systematic collection of incorrect rationales can make CoTErrorSet a promising benchmark in providing future improvements from a different perspective. One example is shown in Table 7.

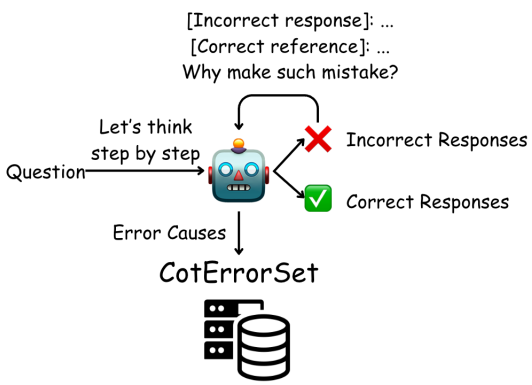


Figure 2: The pipeline to construct CoTErrorSet. By providing PaLM2 with the correct reference and the incorrect response generated by itself, we prompt it to introspect and grasp the underlying reasons for its errors.

3.2 Error Analysis with CoTErrorSet

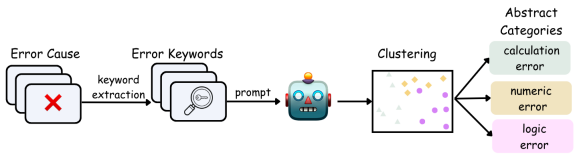


Figure 3: Our pipeline for clustering PaLM2's mistakes.

After collecting the CoTErrorSet dataset, we observe that the error types in it are very intricate and diverse. The intricacy poses obstacles to subsequent enhancement efforts. In order to tackle this issue and gain a more overarching understanding of LLMs' error types, we utilize an LLM-based unsupervised clustering approach shown in Figure 3 to match diverse error types into more general categories.

To be specific, we begin by extracting the specific error keywords from each error cause. Subsequently, we input all the extracted keywords into the LLMs and prompt them to generate more general categories that encompass the entire spectrum of error names. Following this automated clustering process, we manually review each cluster, making necessary adjustments to refine the matching results. Finally, we distill the diverse error types into several abstract categories, such as calculation error, numeric error, and logical error in domains of arithmetic reasoning and logical error, commonsense error, linguistic error, and context error in domains of commonsense reasoning. A detailed definition of each error category is shown in Appendix D. We put results and analysis in Section 8.

Step 1: CoT reasoning

- *Question:* Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?
- *Answer:* Let's think step by step.
- Response: Natalia sold 48 * 2 = 96 clips in May. Natalia sold 48+96 = 144 clips altogether in April and May.

Step 2: self-rethinking

- *Question:* Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?
- *Your output:* Natalia sold 48 * 2 = 96 clips in May. Natalia sold 48+96 = 144 clips altogether in April and May.
- *Do you make similar mistakes with the following examples:* Error Type 1: Misapplication of Algebraic Identities: ...
- Response: Yes, I make a mistake.

Step 3: correction (if they think they make mistakes)

- *So the correct answer is:*
- Response: Natalia sold 48/2 = 24 clips in May. Natalia sold 48+24 = 72 clips altogether in April and May.

Table 1: One example of interactive prompting and responses for self-rethinking. Black texts are the prompting while the red content serves as LLMs' response example.

4 Our Methodology: Self-rethinking

Self-rethinking offers an innovative approach to encourage LLMs to consider if they are repeating past errors. This method starts with an initial CoT reasoning. Following this, the model uses the provided reasoning outputs and a random selection of examples from CoTErrorSet. This step is designed to assess if the model's most recent response includes similar inaccuracies. If errors are detected, it will formulate a new rationale and undergo the evaluation process again. This cycle continues until the model deems its latest answer to be correct or it reaches a set limit of evaluation rounds. The main goal is to empower the LLM to learn from its errors introspectively and minimize the recurrence of such mistakes. One example is shown in Table 1.

The core of self-rethinking lies in the backward-checking stage. In this phase, the LLM reviews its reasoning chain, but with a specific focus on the error types it previously identified. This explicit demonstration of errors, coupled with the question, golden reference, and incorrect rationales, is instrumental in enabling the LLM to recognize specific types of mistakes it tends to make. This targeted review helps the LLM to not just correct the random errors but to consciously avoid repeating the same types of mistakes it has made in the past. The process includes a loop for error correction and confirmation. If the LLM finds that it has repeated any of the previously identified mistakes, it revisits the reasoning process to correct them. Otherwise, the last response is adopted as the final result.

Moreover, the iterative checking process should have a crucial repeating boundary, denoted as k iterations. If the LLM's error-checking and correction cycles surpass this predefined threshold and errors still persist, the process concludes under the assumption that the issue at hand or the error detection might exceed the LLM's current capabilities. This constraint prevents the LLM from being caught in an endless loop of self-rethinking, ensuring the efficiency and practicality of the reasoning process.

5 Our Methodology: Mistake Tuning

In order to fully investigate the other potential utilization of our principles, we introduce mistake tuning, which demonstrates our motivation is a broad and pioneering framework not only in the field of in-context learning. This approach is designed to finetune LLMs on the combinations of both correct rationales and incorrect mistakes. By simply appending prefixes *[CORRECT RATIONALE]* and *[INCORRECT RATIONALE]* before corresponding rationales, mistake tuning can further improve LLMs' abilities to distinguish between correct and incorrect rationale.

Mistake tuning is built upon the foundational motivations and conclusions of self-rethinking, where LLMs can learn from the implicit reasons and types of mistakes they made in CoT reasoning. This process can be formulated as:

$$p = [Q \oplus S \oplus R], \tag{1}$$

$$\mathcal{L} = -\sum_{t=1}^{|p|} logP(p_t|p_{<t}), \tag{2}$$

Where Q, S and R represent the given question, special prefix and corresponding rationale respectively. $\oplus$ represents the operation of concatenation.

Mistake tuning presents a cost-effective, straightforward, and efficient alternative. Previous work has proven pretraining with some controlled signals based on human feedback can result in LLMs' better ability to generate more satisfactory contents (Korbak et al., 2023; Keskar et al., 2019). Hence, incorporating fixed prefixes in finetuning LLMs in the field of reasoning can also help models differentiate information from golden references and mistakes. Our results also demonstrate its effectiveness for promoting LLMs' reasoning abilities without additional costs similar to annotating golden reasoning references.

6 Experiments

In this section, we conducted a series of experiments to compare the proposed self-rethinking methods with the existing approach on both arithmetic and commonsense reasoning benchmarks.

6.1 Experiment Setup

We conduct comparisons between self-rethinking and several other baselines on multiple benchmarks.

Baselines: We select the following reasoning baselines to evaluate our framework, self-rethinking's performance.

- Standard prompting (Brown et al., 2020): the basic reasoning promptings with prefixes as *question* and *answer*.
- Chain-of-Thought (CoT) (Madaan et al., 2023): a technique that enhances large language models' ability to perform complex and multi-step reasoning by guiding them through a problem-solving process step by step, significantly improving their performance on tasks that require deeper cognitive processing.
- Self-refine (Madaan et al., 2023): an approach that enables LLMs to iteratively improve their initial outputs by providing feedback to themselves and refining their responses.
- Self-consistency (Wang et al., 2022): a decoding strategy that enhances CoT prompting in LLMs by sampling multiple reasoning paths and selecting the most consistent answer.

Benchmarks: We consider the following existing math problems benchmarks designed with human rationale reference.

- GSM8K benchmark of math word problems (Cobbe et al., 2021).
- AQuA dataset of algebraic math problems (Ling et al., 2017).
- MathQA benchmark of multiple-choice math problems (Amini et al., 2019).
- Openbook benchmark modeled after open book exams for assessing human understanding of a subject (Mihaylov et al., 2018).
- LogiQA dataset sourced from expert-written questions for testing human logical reasoning (Liu et al., 2020).
- Critical Reasoning in MARB benchmark of several graduate admission tests, highlighting the reasoning to assumptions, conclusions and paradoxes in arguments (Tong et al., 2023).

Models: In order to evaluate self-rethinking's effects, we choose PaLM2 (Anil et al., 2023) and GPT4 (OpenAI, 2023) as the baseline model. PaLM2 is a dense left-to-right, decoder-only language model. It is pre-trained on a high-quality corpus of 780 billion tokens with filtered webpages, books, Wikipedia, news articles, source code, and social media conversations. GPT4 is a large-scale multi-modal state-of-the-art model that exhibits human-level performance on various tasks. We use PaLM2's TEXT-BISON-001 and GPT4's GPT-4 models provided in their APIs.

For mistake tuning, we choose two different-sized Flan T5 (Chung et al., 2022), which are specifically designed for instruction tuning strategies. This model excels in understanding and generating human-like text, demonstrating remarkable performance across a wide range of natural language processing tasks.

Training Details: All of the following experiments were designed with a common setting, employing a random seed of 42, learning rate=1e-4. Considering the vast number of data in AQuA, we only randomly select 10,000 of them to represent the differences in tuning on two different domains.

6.2 Self-rethinking Results

Table 2 presents PaLM2's evaluation results on chosen benchmarks. In this experiment, we set our method, self-rethinking's k equal to 1 to trade between the accuracy and computing resources. In order to align the commuting budget with our methods, we set the times of inference in self-consistency to 3. Our approach involves an initial

Methods	GSM8K	AQuA	MathQA	OpenbookQA	LogiQA	CR
Standard Prompting (Brown et al., 2020)	17.06	22.40	27.57	80.92	41.21	24.45
CoT (Madaan et al., 2023)	56.29	32.11	30.89	82.66	41.05	51.98
Self-refine (Madaan et al., 2023)	34.74	39.92	**54.01**	28.75	35.99	12.28
Self-consistency (Wang et al., 2022)	58.38	42.80	41.37	87.61	42.88	22.58
Self-rethinking (Ours)	**65.13**	**44.72**	43.95	**87.71**	**49.12**	**54.53**

Table 2: PaLM2's accuracy on the existing baselines and our methods, self-rethinking prompting. Self-rethinking shows consistent improvements but uses less inference time compared with self-consistency.

Methods	GSM8K	AQuA	MathQA	LogiQA
8-shot CoT	64.56	30.65	36.21	29.57
8-shot self-rethinking	**70.15**	**34.80**	**40.56**	**33.64**

Table 3: PaLM2's accuracy results on few-shot Chain-of-Thought(CoT) and our methods, self-rethinking. We select 8-shot examples from the corresponding trainset. Then we collect PaLM2's incorrect rationales of those 8 examples. The part of the original correct reference is CoT's demonstrations. Those generated incorrect rationales serve as demonstrations for the rethink stage.

Methods	GSM8K	AQuA	OpenbookQA	CR
CoT	97.93	88.98	93.21	78.92
Self-rethinking	**98.02**	**91.03**	**95.07**	**81.37**

Table 4: GPT4' results on zero-shot Chain-of-Thought (CoT) and our methods, self-rethinking.

zero-shot CoT inference, then rethinking whether this rationale has made similar errors. This leads to the final answer if no errors are found. If inaccuracies are detected, it combines a demonstration and the previously suspected erroneous answer for a third inference to arrive at the final answer. Hence, the overall inference times in our methods are between 2 and 3 times per question, which is still lower than self-consistency here.

With the considered computational settings, the self-rethinking method shows superior performance with significant improvements, especially in GSM8K, AQuA, MathQA, and LogiQA, clearly outperforming self-consistency under a similar computing cost. However, while our method surpasses CoT in performance on the MathQA dataset, it falls short of achieving self-refine results. It's important to note that this dataset is specifically tailored towards operation-based arithmetic problems rather than general questions, aiming to gauge the models' proficiency in tackling complex issues (Amini et al., 2019). This suggests that the nature of the MathQA dataset may inherently be more suitable for self-refine. In contrast to our approach, which aims to amend responses by identifying and addressing typical errors. Table 4 compares GPT4's performance of CoT and self-rethinking. The results demonstrate a notable improvement when using our self-rethinking method over CoT. These findings suggest that self-rethinking is a more effective approach for enhancing GPT-4's performance.

Table 3 presents the 8-shot examples of CoT and self-rethinking, using the PaLM2 model across four different tasks: GSM8K, AQuA, MathQA, and LogiQA. A key part of the process involved collecting PaLM2's incorrect rationales for these examples, which were then used as learning demonstrations to rethink. The results show a clear advantage of the self-rethinking method over the standard 8-shot CoT approach. These results highlight the efficacy of the self-rethinking method in improving accuracy in few-shot learning scenarios for complex problem-solving tasks.

Notably, self-refine shares our basic motivations about self-refining or self-correcting their answers but without utilizing any mistake samples. The result shows that our self-rethinking outperformed self-refine by a considerable margin across most of the datasets. This indicates the importance of our proposal for utilizing previous mistake examples. While self-refine demonstrates improvements in three arithmetic reasoning datasets, it concurrently exhibits substantial performance drops in commonsense reasoning datasets. By contrast, our self-rethinking consistently outperforms the standard method in various domains. This further implies the introduction of previous mistakes can stabilize the refinement and rethinking process.

In conclusion, our self-rethinking method achieved remarkable accuracy improvements in most tests, particularly in scenarios that demand high logical rigor and offer the opportunity to learn from errors by identifying fixed logical patterns, especially in arithmetic reasoning tasks. It indicates self-rethinking effectiveness in tasks requiring strong logic and prone to minor errors. Additionally, the self-rethinking method proves particularly beneficial in assisting LLMs in identifying and rectifying low-level mistakes or misunderstand-

ings that are within the model's capabilities but have been previously overlooked. This capability indicates that self-rethinking can serve as a valuable tool in refining the accuracy and reliability of responses in LLMs, especially in complex problem-solving contexts.

Models	Methods	GSM8K	MathQA	AQuA
Flan-T5-large (780M)	Standard finetuning	14.28	42.79	13.10
	Mistake tuning	**18.36**	**48.95**	**18.07**
Flan-T5-xl (3B)	Standard finetuning	23.81	47.24	17.81
	Mistake tuning	**24.29**	**52.22**	**20.99**

Table 5: Accuracy of Standard finetuning models (with only correct rationales) vs. our methods, mistake tuning (combined correct and incorrect rationales). Mistake tuning shows consistent and superior performance compared with only fine-tuned correct rationales.

6.3 Mistake Tuning Results

Table 5 showcases the performance of the Flan-T5 models in the context of mistake tuning, highlighting the impact of combining correct and incorrect rationales. The data presented in Table 5 reveals significant insights into the performance of Flan-T5 models under mistake tuning, which involves integrating both correct and incorrect rationales. This approach is evident across different model scales, whether it's the smaller 780M version or the larger 3B variant. Notably, in the MathQA domain, Flan-T5-large(780M) tuned by our methods demonstrates superior performance compared to PaLM2, achieving an accuracy of 48.95% versus 41.37%. This phenomenon suggests that LLMs can benefit from engaging with incorrect reasoning, thereby enhancing their problem-solving and reasoning capabilities. It extends beyond merely bolstering the model's grasp of correct CoT, to also encompassing the ability to identify and learn from incorrect rationales.

Furthermore, the expense of obtaining ground truth or hand-crafted references is significantly higher compared to generating and collecting incorrect rationales. This cost disparity underscores the practical value of our approach, offering a more cost-effective solution without compromising the quality of training data for machine learning models. All mentioned provides a direction for further work of reasoning, which involves not only enhancing the model's understanding and learning of correct CoT but also the ability to identify and learn from incorrect rationales.

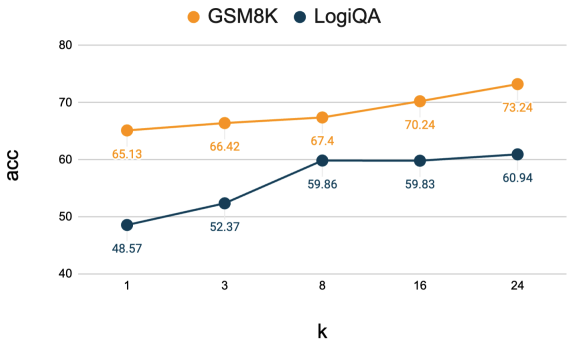


Figure 4: Accuracy of different re-thinking iterations(k). As the value of k increases, the overall prediction accuracy improves.

7 Further Studies

7.1 Hyperparameter Analysis of Rethinking Iteration Times

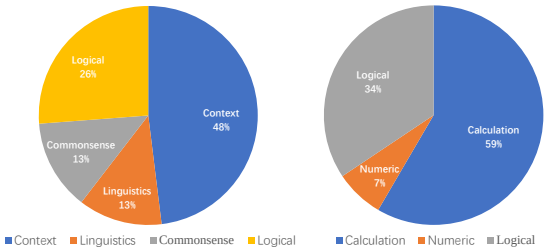
In this section, we conduct experiments to assess the impact of different rethinking iterations, denoted as k, on the performance of our framework. We evaluate it on two mainstream benchmarks in the field of mathematics and commonsense reasoning, GSM8K and LogiQA. Figure 4 represents the detailed trend under varying re-thinking times. Notably, as k increases from 1 to 24, GSM8K represents a growth of 8.11% and 12.37% in LogiQA. It is evident as k increases, both LLMs' arithmetic and commonsense reasoning accuracy exhibit an upward trend. This trend suggests a positive correlation between the number of rethinking iterations and the overall reasoning abilities. These observations indicate self-thinking's potential benefits with more inference time.

CAT.	DEM.	COR.	INC.	GSM8K	LogiQA
✓				64.30	50.21
✓	✓			62.70	48.57
✓	✓	✓		65.70	51.01
✓	✓	✓	✓	65.13	49.21

Table 6: Impact of Component Combinations. CAT. stands for the previous mistakes' type name, DEM. are the reasons for making such mistakes, and COR. and INC. mean corresponding correct and incorrect rationale examples. All components here are generated by LLM itself before reasoning.

7.2 Ablation Study on Rethinking Process

In this ablation study, we examined the impact of various component combinations in promptings to guide LLMs to self-rethinking . Table 6 shows the performance of different components. The results



(a) Commonsense Reasnoing (b) Arithmetic Reasoning

Figure 5: PaLM2's error type distribution in the commonsense and arithmetic reasoning task.

indicate that the inclusion or exclusion of different components has varying effects on PaLM2's accuracy in domains of GSM8K and LogiQA. However, the overall performance across various components is relatively similar. It performs similarly well regardless of the specific combination of components, indicating good generalizability of the method. This study suggests our method's flexibility and stability in future usage.

8 Unveiling LLM's Reasoning Errors

In this section, we delve into the detailed types and underlying reasons that lead to mistakes in LLMs's inference process. We sample mistake examples from GSM8K and LogiQA to conduct an in-depth analysis of both arithmetic and commonsense reasoning. We put some examples in Appendix C.

For commonsense reasoning, we find errors like the misinterpretation of facts or concepts usually arise due to the model's limitations in understanding and applying context accurately. This reveals current LLMs may still fall short of consistently recalling precise factual knowledge within a given context. Consequently, this underscores the imperative to advance toward the development of Retrieval-Augmented Generation(RAG) systems (Guu et al., 2020; Mallen et al., 2022), as they hold the promise of yielding more faithful and contextually aligned results. Additionally, errors stemming from logical fallacies or incorrect inferences reveal LLMs' reliance on pattern recognition over logical reasoning, sometimes leading them to make logically inconsistent or unsupported connections by the given facts.

As shown in Figure 5, the most errors made by LLMs in arithmetic reasoning are about calculation. This can be attributed to the different nature of LLMs compared to other tools like calculators. To address this issue, Chen et al. (2022)'s suggestion using Program-of-Thought (PoT) is a promising approach to instruct LLMs to generate a segment of code to solve the given problem, resulting in more accurate calculation results. Furthermore, it's important to note that logical error is also a type of error that LLMs always suffer from. Compared with calculation errors and numeric errors, the causes of logical errors are more complicated and nuanced. For instance, errors like misinterpreting given data or misapplying arithmetic operations reveal a lack of depth in understanding mathematical relationships. This can result from the model's limitations in comprehending the nuances of mathematical concepts or its inability to correctly infer the needed function from the context of the question. In the future, more fine-grained analysis and methods are needed to address such complex logical errors in arithmetic reasoning.

9 Conclusions and Future Work

In this work, we explore whether LLMs can learn from their mistakes. In order to investigate LLMs' abilities to differentiate and learn from mistakes, we introduce COTERRORSET, a novel benchmark collecting both correct and incorrect CoT rationales across various domains and designed with demonstrations for making errors. We propose two possible solutions to expose the effects of mistakes from different perspectives: self-rethinking and mistake tuning. Both of them have achieved consistent and significant improvements, which demonstrates the potential benefits of learning from reasoning errors. In the last, we conduct a comprehensive and detailed analysis of LLMs' common mistakes in both arithmetic and commonsense reasoning. The findings will provide a clear direction for future improvements.

For future work, we envision proposing corresponding algorithms or loss functions to learn implicit information from mistakes. The primary intent of this work is to provide a new paradigm so there are still a lot of improvements that can be down following this work. For example, incorporating contrastive learning to differentiate correct references and errors is intuitive to make more improvements. Also, some memorization and retrieval-augmented skills can help models benefit from mistakes similar to each question.

Limitations

In addition to the noted challenge of fine-tuning commercial LLMs, we recognize several other

specific limitations in our study that require attention. Primarily, our self-rethinking methodology may not be entirely suitable for tasks where a distinct, objective label is not readily available, such as in machine translation or dialogue generation. These areas pose a unique challenge as the correctness of outputs can often be subjective or context-dependent, making it difficult to apply our approach effectively. Moreover, our utilization of the CoTErrorSet collection for mistake tuning necessitates a ground truth label for each sample, posing a potential impediment to the applicability of our method in low-resource scenarios. In the future, we will continually improve our method and bring the concept of learning from mistakes to wider scenarios and applications. Thanks again for your thoughtful insights and informative comments.

References

Shourya Aggarwal, Divyanshu Mandowara, Vishwajeet Agrawal, Dinesh Khandelwal, Parag Singla, and Dinesh Garg. 2021. Explanations for commonsenseqa: New dataset and models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3050–3065.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. 2022. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*.

Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*.

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. 2023. Learning from mistakes makes llm better reasoner. *arXiv preprint arXiv:2310.20689*.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Michal Podstawski, Hubert Niewiadomski, Piotr Nyczyk, et al. 2023. Graph of thoughts: Solving elaborate problems with large language models. *arXiv preprint arXiv:2308.09687*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Oana-Maria Camburu, Tim Rocktäschel, Thomas Lukasiewicz, and Phil Blunsom. 2018. e-snli: Natural language inference with natural language explanations. *Advances in Neural Information Processing Systems*, 31.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2023. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*.

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

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Antonia Creswell and Murray Shanahan. 2022. Faithful reasoning using large language models. *arXiv preprint arXiv:2208.14271*.

Xiang Gao and Kamalika Das. 2024. Customizing language model responses with contrastive in-context learning. *arXiv preprint arXiv:2401.17390*.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

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. *arXiv preprint arXiv:2103.03874*.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*.

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

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8082–8090.

Seungone Kim, Se June Joo, Doyoung Kim, Joel Jang, Seonghyeon Ye, Jamin Shin, and Minjoon Seo. 2023. The cot collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning. *arXiv preprint arXiv:2305.14045*.

Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Vinayak Bhalerao, Christopher Buckley, Jason Phang, Samuel R Bowman, and Ethan Perez. 2023. Pretraining language models with human preferences. In *International Conference on Machine Learning*, pages 17506–17533. PMLR.

Matthew Lamm, Jennimaria Palomaki, Chris Alberti, Daniel Andor, Eunsol Choi, Livio Baldini Soares, and Michael Collins. 2021. Qed: A framework and dataset for explanations in question answering. *Transactions of the Association for computational Linguistics*, 9:790–806.

Bin Lei, Chunhua Liao, Caiwen Ding, et al. 2023. Boosting logical reasoning in large language models through a new framework: The graph of thought. *arXiv preprint arXiv:2308.08614*.

Dawei Li, Yaxuan Li, Dheeraj Mekala, Shuyao Li, Xueqi Wang, William Hogan, Jingbo Shang, et al. 2023a. Dail: Data augmentation for in-context learning via self-paraphrase. *arXiv preprint arXiv:2311.03319*.

Yifei Li, Zeqi Lin, Shizhuo Zhang, Qiang Fu, Bei Chen, Jian-Guang Lou, and Weizhu Chen. 2023b. Making language models better reasoners with step-aware verifier. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5315–5333.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. 2022. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*.

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

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. 2020. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. *arXiv preprint arXiv:2007.08124*.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. 2022. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. *arXiv preprint arXiv:2212.10511*.

Neil Mercer. 2008. Talk and the development of reasoning and understanding. *Human development*, 51(1):90–100.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Yasumasa Onoe, Michael JQ Zhang, Eunsol Choi, and Greg Durrett. 2021. Creak: A dataset for commonsense reasoning over entity knowledge. *arXiv preprint arXiv:2109.01653*.

OpenAI. 2023. Gpt-4 technical report.

Liangming Pan, Michael Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2023. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *arXiv preprint arXiv:2308.03188*.

Taly Reich, Alex Kaju, and Sam J Maglio. 2023. How to overcome algorithm aversion: Learning from mistakes. *Journal of Consumer Psychology*, 33(2):285–302.

Yongqi Tong, Yifan Wang, Dawei Li, Sizhe Wang, Zi Lin, Simeng Han, and Jingbo Shang. 2023. Eliminating reasoning via inferring with planning: A new framework to guide llms' non-linear thinking. *arXiv preprint arXiv:2310.12342*.

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*.

Cunxiang Wang, Shuailong Liang, Yue Zhang, Xiaonan Li, and Tian Gao. 2019. Does it make sense? and why? a pilot study for sense making and explanation. *arXiv preprint arXiv:1906.00363*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

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 preprint arXiv:2305.10601*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

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

A An example in CoTErrorSet

B ALgorithm for self-rethinking

C Reasoning Mistake Examples

D More Details about LLM-based Clustering Approach

Questions: Combine facts and answer this: Which meridian extends across Europe, the Mediterranean Sea, Africa, Asia, the Pacific Ocean, North America, and the Atlantic Ocean?
Target: The Cimarron meridian
Reference: The Cimarron meridian extends across Europe, the Mediterranean Sea, Africa, Asia, the Pacific Ocean, North America and the Atlantic Ocean.
Incorrect Rationale: The 180th meridian extends across Europe, the Mediterranean Sea, Africa, Asia, the Pacific Ocean, North America and the Atlantic Ocean.
Error Causes: Making mistakes in incorrect rationales, such as claiming the 180th meridian extends across various continents and oceans, can lead to significant misinformation and confusion. This particular error demonstrates a fundamental misunderstanding of geography, as the 180th meridian primarily runs through the Pacific Ocean and does not cross the regions listed. Such inaccuracies underscore the importance of fact-checking in educational content to prevent the spread of misconceptions. Correcting these mistakes not only clarifies the factual information but also serves as a valuable learning opportunity, emphasizing the need for accuracy and critical evaluation of information.

Table 7: An example in CoTErrorSet. The content of *Incorrect Rationale* and *Error Causes* are generated by PaLM2 as indicated in Figure 2.

Error name: Misinterpretation of Given Data
Error type: Logical

- *Example:* Natalia sold clips to 48 of her friends in April, and then she sold half as many clips in May. How many clips did Natalia sell altogether in April and May?
- *Correct Answer:* Natalia sold 48/2 = 24 clips in May. Natalia sold 48+24 = 72 clips altogether in April and May.
- *Incorrect Rationale:* Natalia sold 48 * 2 = 96 clips in May. Natalia sold 48+96 = 144 clips altogether in April and May.
- *Demonstration:* Mistaking multiplication for division led to a significant overestimate of the total clips sold.

Error type: Overlooking Details
Error type: Logical

- *Example:* Mark has a garden with flowers. He planted plants of three different colors in it. Ten of them are yellow, and there are 80% more of those in purple. There are only 25% as many green flowers as there are yellow and purple flowers. How many flowers does Mark have in his garden?
- *Correct Answer:* There are 80/100 * 10 = 8 more purple flowers than yellow flowers. So in Mark's garden, there are 10 + 8 = 18 purple flowers. Purple and yellow flowers sum up to 10 + 18 = 28 flowers. That means in Mark's garden there are 25/100 * 28 = 7 green flowers. So in total Mark has 28 + 7 = 35 plants in his garden.
- *Incorrect Rationale:* There are 80/100 * 10 = 8 more purple flowers than yellow flowers. So in Mark's garden, there are 10 + 8 = 18 purple flowers. That means in Mark's garden there are 25/100 * 18 = 4.5 green flowers. So in total Mark has 10 + 18 + 4.5 = 32.5 plants in his garden.
- *Demonstration:* Neglecting to consider both yellow and purple flowers in the green flower calculation led to a significant underestimation of the total number of flowers in Mark's garden.

Error name: Misapplication of Arithmetic Operation
Error type: Calculation

- *Example:* Weng earns $12 an hour for babysitting. Yesterday, she just did 50 minutes of babysitting. How much did she earn?
- *Correct Answer:* Weng earns 12/60 = $0.2 per minute. Working 50 minutes, she earned 0.2 x 50 = $10.
- *Incorrect Rationale:* Weng earns 12/60 = $2 per minute. Working 50 minutes, she earned 2 x 50 = $100.
- *Demonstration:* Confusing the rate per hour with the rate per minute led to a substantial overestimation of earnings.

Error name: Numerical
Error type: Numeric

- *Example:* The chicken crossed the road to get to the other side twice for the thrill of it. The first time, it had to dodge 23 speeding cars. The second time, a person tried to catch it and accidentally pulled out twice as many feathers as the number of cars the chicken had dodged. The chicken had 5263 feathers before its thrill-seeking road crossings. How many feathers did it have afterward?
- *Correct Answer:* The chicken lost 23 * 2 = «23*2=46»46 feathers on its second road crossing., it had 5263 - 46 = «5263-46=5217»5217 feathers after crossing the road twice.
- *Incorrect Rationale:* The chicken lost 23 * 2 = «23*2=46»46 feathers on its second road crossing., it had 5263 - 46 = «5263-52=5211»5211 feathers after crossing the road twice.
- *Demonstration:* 1. The correct answer is 5217, while your answer is 5211. 2. Your answer is wrong because you subtracted 52 instead of 46. 3. The type name of the incorrect answer is numerical.

Table 8: Examples of Error Types in Arithmetic Reasoning. All contents are generated by PaLM2 itself.

Error name: Logical Fallacy or Incorrect Inference
Error type: Logical

- *Example:* "When standing miles away from Mount Rushmore"
- *Correct Rationale:* Objects appear smaller when viewed from a greater distance.
- *Incorrect Rationale:* "The mountains do not look smaller when standing miles away from Mount Rushmore. They look larger." (Logical fallacy)
- *Demonstration:* 1. The correct rationale is that objects appear smaller when viewed from a greater distance, whereas the incorrect rationale states the opposite. 2. This is a logical fallacy as it contradicts a basic principle of perception. 3. The type name of the incorrect rationale is logical.

Error name: Incorrect Assumptions or Generalizations
Error type: Logical

- *Example:* "Poison causes harm to which of the following?"
- *Correct Rationale:* Poison affects living organisms.
- *Incorrect Rationale:* "Robots do not get hurt by poison." (Incorrect generalization about the effects of poison)
- *Demonstration:* 1. The correct rationale is that poison affects living organisms, but the incorrect rationale generalizes that robots are immune to poison. 2. This is an incorrect generalization because robots, being non-living entities, are not subject to biological effects. 3. The type name of the incorrect rationale is logical.

Error name: Misunderstanding Literal vs. Metaphorical Language
Error type: Linguistics

- *Example:* "When food is reduced in the stomach"
- *Correct Rationale:* Digestion involves the breakdown of food by stomach acid.
- *Incorrect Rationale:* "Choice D is incorrect because it is not a fact." (Misunderstanding metaphorical language)
- *Demonstration:* 1. The correct rationale is about the literal process of digestion, whereas the incorrect rationale misinterprets the metaphorical language. 2. This demonstrates a misunderstanding of metaphorical language. 3. The type name of the incorrect rationale is linguistics.

Error name: Factual Inaccuracy
Error type: Commonsense

- *Example:* "You can make a telescope with a"
- *Correct Rationale:* A telescope requires specific optical elements to function.
- *Incorrect Rationale:* "A telescope needs a lens and a magnifying glass is a lens, so glass is a good choice." (Factually inaccurate about how telescopes are made)
- *Demonstration:* 1. The correct rationale is that a telescope requires specific optical elements, whereas the incorrect rationale assumes any lens, like a magnifying glass, can make a telescope. 2. This shows a factual inaccuracy in understanding how telescopes are constructed. 3. The type name of the incorrect rationale is commonsense.

Error type: Misunderstanding Context or Relevance
Error type: Context

- *Example:* "an inherited characteristic found on all mammals is"
- *Correct Rationale:* Inherited characteristics in mammals include features like fur.
- *Incorrect Rationale:* "Shoes are not found on all mammals" (Misunderstanding the context of biological characteristics)
- *Demonstration:* 1. The correct rationale focuses on relevant inherited physical traits like fur. 2. This error illustrates a clear lack of understanding of the context. 3. The type name of the incorrect rationale should be context.

Table 9: Examples of Error Types in Commonsense Reasoning. All contents are generated by PaLM2 itself.

Algorithm 1 self-rethinking

Mistakes = {...}
Correct & Incorrect Examples = {...}
ErrorCounter ← 0
Prompt: Why you made the mistakes?
Mistakes ← Error Type, Demonstrations, Examples.
Stage1 Prompt: Let's think step by step.
Stage2 Prompt: Do you make the same mistakes in *Mistakes*?
while ErrorCounter < k **do**
  if Yes **then**
    go to Step2
    ErrorCounter ← *ErrorCounter* + 1
  else if No **then**
    get the answer
    break
  end if
end while
if *ErrorCounter* == k **then**
  Assume: Problem or error detection exceeds the model's capabilities.
end if
Prompt: So the final answer is:

Input	Please generate several keywords to cover all the following error types, and assign each keyword to an error type category. Output in the following format: [Specific Error Category1]: [keyword1], [keyword2] [Specific Error Category2]: [keyword3], [keyword4] Keywords: {keywords}
Output	Mathematical: {keywords cluster1} Numerical: {keywords cluster2} Arithmetic: {keywords cluster3} Calculation: {keywords cluster4}

Table 10: Detailed input and output of our LLM-based clustering method.

Error Type	Definition
Calculation	Mistakes or inaccuracies that occur during the process of performing mathematical calculations. These errors can arise from various sources and can occur at any stage of a mathematical problem-solving process.
Numeric	Numeric errors in the context of mathematical reasoning refer to inaccuracies that arise from the representation and manipulation of numerical values. These errors can occur at various stages of mathematical computations and can result from limitations in the precision of the representation of real numbers or mistakes in handling numerical data.
Logical	Logical errors involve mistakes in the overall reasoning or strategy used to solve a mathematical problem. This type of error may not be immediately apparent during the calculation process but can lead to incorrect final results. It could include using an incorrect formula or assumptions, misunderstanding the problem statement, or applying the wrong concept.
Linguistics	Errors in linguistics involve inaccuracies or mistakes in the use of language. These can include grammatical errors, misuse of vocabulary, incorrect syntax, or problems in semantics. Linguistic errors may arise from a lack of understanding of the rules of a language, misinterpretation of meaning, or the inability to effectively convey a message in a given language. Such errors can affect the clarity, coherence, and overall effectiveness of communication.
Commonsense	Commonsense errors refer to mistakes or inaccuracies that occur in the application of general world knowledge or everyday reasoning. These errors can arise from misconceptions, flawed logic, or misunderstandings of basic principles that are widely accepted as common knowledge. Commonsense errors often lead to conclusions or decisions that, upon closer examination, are illogical or inconsistent with general understanding of the world.
Context	Errors of misunderstanding context or relevance occur when there is a failure to correctly interpret or apply the relevant information in a given scenario. This type of error typically involves overlooking key aspects of a context, making inappropriate generalizations, or failing to distinguish between literal and metaphorical language. These errors can significantly alter the intended meaning or relevance of a response in reasoning tasks.

Table 11: PaLM2's Understanding and Definitions for Error Types. All contents are generated by itself after providing its mistakes and corresponding golden-standard references.