Over-Reasoning and Redundant Calculation of Large Language Models

Anonymous ACL submission

Abstract

Large language models (LLMs) can solve problems step-by-step. While this chain-of-thought (CoT) reasoning boosts LLMs' performance, it is unclear if LLMs *know* when to use CoT and whether those CoT are always necessary to answer the question. This paper shows that LLMs tend to generate redundant calculations and reasoning on a manually constructed math QA dataset, *GSM8K-Zero*. GSM8K-Zero is constructed such that the questions can be answered without any calculations, but LLMs, including Llama-2 models and Claude-2, tend to generate lengthy and unnecessary calculations to answer the questions. We also conduct experiments to explain why LLMs generate redundant calculations and reasonings.

1 Introduction

LLMs have shown impressive performance on various tasks (Brown et al., 2020; Ouyang et al., 2022; Touvron et al., 2023), and CoT reasonings (Wei et al., 2022; Kojima et al., 2022) further boost the LLMs' performance. To make LLMs better learn to think step-by-step, it is common to enhance the CoT ability of LLMs by adding training samples that contain CoT reasoning in instruction tuning (Longpre et al., 2023).

While step-by-step reasoning is useful, it lengthens the LLM's output. Longer outputs imply longer inference latency and cost more money on commercial LLMs. Moreover, if LLMs give long answers to questions that can be easily solved without complex reasoning, this may confuse the users and make them falsely believe that the questions are difficult and must be solved step-by-step. It is thus more desirable that LLM only triggers CoT reasoning ability when necessary, but it is unclear if LLMs know when to think step-by-step and only generate step-by-step reasoning when necessary. Prior works on CoT mostly focus on the correctness of the CoT reasoning steps or whether the reasoning steps are faithful to the question and support the final answer (Ye and Durrett, 2022; Golovneva et al., 2023). However, no past study focuses on the redundancy in CoT reasonings.

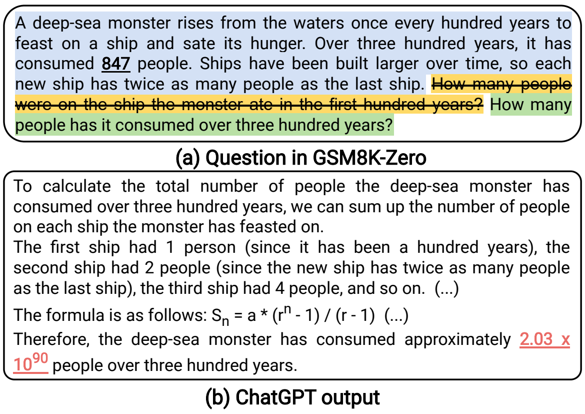

Figure 1: (a) A question in GSM8K-Zero. The blue parts are the known information, and the orange part is the original question in GSM8K and is removed in GSM8K-Zero. The green part is the new question in GSM8K-Zero. (b) The answer generated by ChatGPT.

As an initiative to study the redundancy of LLM outputs, we aim to understand the following research question: Does LLM generate redundant reasonings when they clearly need not do so? To study this question, we construct a math QA dataset, GSM8K-Zero, which contains trivial questions that can be answered without any calculations and reasoning. Using this curated dataset, we can define the redundancy of output from LLMs. We evaluate seven LLMs trained with reinforcement learning with human feedback (RLHF), and we find that LLMs tend to generate redundant calculations that complicate the responses and sometimes lead to the wrong answer. To explain our observation, we show that GPT-4 (OpenAI, 2023) and ChatGPT (OpenAI, 2022), which are widely used in gathering the preference data for training a reward model in RLHF (Guo et al., 2023; Anand et al., 2023), show a strong preference towards long answers that contain redundant calculations, even if the long answers are incorrect.

Our contributions are summarized as follows:

- To the best of our knowledge, we are the first to study the redundancy of LLM outputs.
- We show that LLMs tend to generate redundant calculations on math questions that can be answered without any calculation.
- We show that LLMs' tendency to generate long answers may stem from the imperfect reward model that prefers longer answers regardless of its correctness.

2 Dataset: GSM8K-Zero

2.1 Construction of GSM8K-Zero

To study LLMs' tendency for redundant calculations, we created **GSM8K-Zero** from GSM8K (Cobbe et al., 2021). A question in GSM8K comprises (1) the **known** information (blue parts in Figure 1) and (2) a query for an **unknown** quantity (orange parts in Figure 1). Using questions in GSM8K, we aim to create questions whose answers are directly stated in the questions and can be obtained without any calculations.

We use the following procedure to achieve this goal. The following procedure is best read with Figure 1 (a). Given a question in GSM8K, we remove the last sentence from the question that queries for an unknown variable and keep the known information. Next, we generate a question that asks the value of a known variable (green parts in Figure 1 (a)) based on the known information and append the question behind the known information. The question is generated by randomly selecting a number in the known information as the ground truth answer and using few-shot prompting to generate a question whose answer is the selected ground truth using ChatGPT. We then use GPT-4 to answer the newly generated question. If GPT-4's answer deviates from the ground truth answer, the question is discarded. We randomly select 3,500 questions from GSM8K's training set[1] and obtain 2,978 question-answer pairs after the above procedure. Based on a manual inspection of 250 random question-answer pairs by the authors, we estimate that about 85% of question-answer pairs in GSM8K-Zero are valid.

[1]In our preliminary experiment, we find that our results also hold when we use the testing set of GSM8K to construct the questions in GSM8K-Zero

2.2 Evaluating Redundancy

We define redundant outputs as **any superfluous information in LLM responses that are not required for accurately answering the question**. Measuring this redundancy is often challenging for existing datasets. However, GSM8K-Zero offers an easy way to evaluate LLM output redundancy due to its unique nature: questions can be answered without any calculations since the answers are explicitly stated within the questions. If an LLM's answer includes calculations, it is deemed redundant. We identify mathematical operators ($\times, +$, and $=$) in LLM outputs by a regular expression and say that the LLM's answer is redundant whenever mathematical operators are found.

3 Experiments

We test LLMs on GSM8K-Zero in zero-shot, as zero-shot inference closely mirrors most users' practical use of *LLMs as assistants*. Instead of leveraging advanced prompting techniques like zero-shot CoT (Kojima et al., 2022) or Plan-and-Solve (Wang et al., 2023), we present a single question to the LLM and take its response. For each question, we sample one response from the LLM. In our preliminary experiments, we find the observations in our paper are robust toward the hyperparameters used for sampling outputs from LLMs.

Our evaluation encompasses proprietary LLMs, such as GPT-4, ChatGPT, Claude-2 (Anthropic, 2023), and PaLM (`text-bison-001`) (Anil et al., 2023), and open-source ones like Llama-2-chat models of different sizes (Touvron et al., 2023). We assess LLM performance on GSM8K-Zero using two metrics: (1) **Redundancy**: Determined by the percentage of LLM answers containing numerical operators like $\times, +$, and $=$. (2) **Accuracy**: Accuracy measures how often the LLM's answer, extracted using a regular expression, aligns with the GSM8K-Zero ground truth.

3.1 Main Results

We show the LLMs' performance on GSM8K-Zero in Table 1. First, we observe almost half of the LLMs we test have an accuracy lower than 50% (second column in Table 1). Recall that the answers to the question in GSM8K-Zero can be easily extracted from the question without any calculations,

Models	Red.	Accuracy		
		Avg.	Cal. ✗	Cal. ✓
Proprietary LLMs				
GPT-4	11.7	100.0†	100.0†	100.0†
ChatGPT	47.1	79.7	96.6	60.7
Claude-2	74.7	88.4	98.8	84.8
PaLM	29.2	40.9	40.9	40.6
Open LLMs (Llama-2)				
70b-chat	80.3	54.5	87.7	46.3
13b-chat	88.3	39.9	86.0	33.8
7b-chat	88.6	41.4	80.2	36.3

Table 1: The redundancy (**Red.**) and accuracy of LLMs' responses. We report the average accuracy (**Avg.**) on all questions (second column), the accuracy for answers without calculation (Cal. ✗, third column) and with calculation (Cal. ✓, fourth column). †: The accuracy of GPT-4 is 100% by construction since we use GPT-4 to filter samples when constructing GSM8K-Zero.

Model	Redundancy	Accuracy
ChatGPT	25.7	83.6
Claude-2	40.7	88.5
Llama-2-70b-chat	54.4	73.3
Llama-2-13b-chat	45.8	65.5
Llama-2-7b-chat	32.7	68.3

Table 2: The redundancy and accuracy of answers from LLMs when allowing LLMs not to use CoT.

which makes GSM8K-Zero more like an extractive QA than a math QA. Simple as this dataset is, some LLMs still cannot perform well on this dataset.

Next, we turn our attention to the redundancy in the answers. It can be clearly seen that both proprietary and open-source LLMs generate redundant calculations and reasoning to answer the questions. ChatGPT yield unnecessary calculation in their step-by-step reasoning answers in almost half of the answers, and all Llama-2 models generate lengthy reasoning steps and redundant calculations in more than 80% of their response while they are not explicitly prompted to do so. By manually inspecting those outputs from LLMs, we find that in most cases, LLMs solve all the unknown variables in the questions, which are not asked in the questions. This is a problematic behavior for a helpful assistant since it complicates the responses and may falsely lead the users to think it is necessary to solve all the unknown variables to arrive at the final answer. We also find that the LLMs sometimes only provide the values of the unknown variables but do not answer the value asked in the question, showing that LLMs cannot follow user instructions very well in these cases.

After discussing redundancy and accuracy independently, we want to know if redundant calculation co-occurs more often with wrong answers. We separate the model outputs into two groups: one that contains calculations and another that does not have calculations, and we calculate the accuracy for the two groups. The results are shown in the two rightmost columns in Table 1. When the LLM's answers contain calculations, the accuracy drops significantly for almost all models except for PaLM. By manually inspecting the wrong answers that include calculations, we find that sometimes LLMs hallucinate variables not specified in the questions. Sometimes, LLMs make calculation errors and lead to the wrong answer. This shows that redundant calculations not only waste time and resources to generate but can also hurt the LLM's performance due to calculation errors and incorrect reasoning.

3.2 Do LLMs Know When to Use CoT?

Section 3.1 reveals that LLMs can generate redundant calculations and unnecessary CoT reasoning steps. This is possibly because, during instruction tuning, LLMs are trained to generate CoT reasoning for mathematical problems **when the input instruction does not specify how to solve the question**, forcing them to apply CoT on every question that *looks like* a mathematical question. Hence, we are curious whether LLMs can drop the CoT reasoning and calculations **when properly instructed**. To explore this possibility, we append the following instruction after the questions in GSM8K-Zero: "*If the question is simple enough, you can omit the step-by-step reasoning and just give the answer.*" Here, we only test on the LLMs that generate answers with higher redundancy in Section 3.1.

The results are shown in Table 2. We can see that when LLMs are allowed to omit step-by-step reasoning, the redundancy of the LLMs significantly drops compared with Table 1 while the accuracy significantly boosts for almost all models. The decrease in output redundancy implies that LLMs do know that some questions in GSM8K-Zero are easy enough to answer directly. However, even when they are allowed to omit step-by-step reasoning, the redundancy in these LLMs is still higher than 20%. This means that LLMs cannot always correctly infer the difficulty and whether step-by-step reasonings are necessary for the questions.

4 Why Do LLMs Generate Redundant Calculations?

After seeing that LLMs produce excessive calculations, we seek to understand why. We speculate that the reward models (RMs) in RLHF might favor more verbose outputs over concise ones, making RLHF-trained models prone to generate lengthy output even if it is redundant. To test this hypothesis, we would like to compare RM's preference between long and short answers. However, we cannot access RMs used to train ChatGPT or Llama models. As a workaround, we use ChatGPT and GPT-4 as the proxy of the RMs; we call these models *proxy RMs* in this case. To obtain the preference of the proxy RMs, we give proxy RMs some instructions, a question in GSM8K-Zero, a pair of long and short answers, and ask the model to select a better answer. We follow the instructions used in Zheng et al. (2023), which asks the proxy RMs to consider the accuracy and helpfulness of the answer. The experiment is repeated by inverting the order of the short and long answers to counteract potential position bias. Using ChatGPT or GPT-4 as the proxy RMs is reasonable, as these models should learn the preferences of their RMs during RLHF. Additionally, prior works have used ChatGPT and GPT-4 to generate the preference data to train the RMs (Anand et al., 2023), so the preference of ChatGPT or GPT-4 can reflect the preference of RMs.

We prepare the long and short answers as follows: To collect long answers, we collect ChatGPT's answers to questions in GSM8K-Zero, select those with redundant calculations, and group those answers into two: correct answers and incorrect answers, with approximately 100 samples in each group. Next, for each long answer collected, we construct a short answer counterpart by the template, "*The answer is [[ground truth]]*", where *"[[ground truth]]"* is filled in with the ground truth in GSM8K-Zero.

The preference of proxy RMs between long and short answers is shown in Figure 2. First, we observe that when both the long and short answers are correct (Figure 2 (a)), both GPT-4 and ChatGPT prefer long answers. By scrutinizing the evaluation results, we find that GPT-4 and ChatGPT frequently complain about the shorter answer to "only answer the question without any further details," while the long answer "shows more information." However, when reading the long answers, the authors find it hard to locate the answer to the question since the model outputs too much unnecessary information and complicates the problem, making the answer unhelpful. Next, when the long answer is incorrect and the short answer is correct (Figure 2 (b)), we find that ChatGPT consistently prefers lengthy but wrong answers. While GPT-4 successfully prefers the short and correct answer in 61% of the cases, GPT-4 still votes for long but wrong answers in 34% of the cases. Overall, the results in Figure 2 show that proxy RMs strongly prefer long outputs that contain redundant calculations and unnecessary reasoning, even if the final answer is wrong! If we use the proxy RMs' preference data collected in this section, it is easy to think that we will obtain RMs that favor lengthy output, eventually leading to an LLM that generates redundant calculations. We repeat the above experiment using the answers from Llama-7b-chat and observe a similar result.

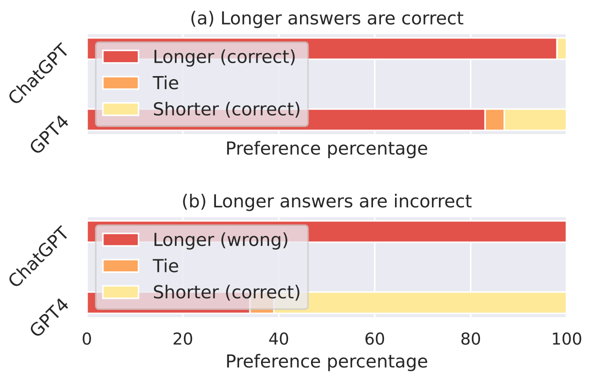

Figure 2: The preference of GPT-4 and ChatGPT between longer and shorter answers. (a) The case when the longer answers are correct. (b) The case when the longer answers are incorrect.

5 Conclusion

In this paper, we construct GSM8K-Zero to illustrate the redundancy in the output from LLMs. We show that LLMs tend to generate redundant calculations and unnecessary reasoning, sometimes leading to a wrong answer. We reveal that LLMs may not differentiate questions requiring step-by-step reasoning from simpler ones, suggesting a possible future research direction. To explain our observation, we use proxy RMs and find that these models prefer lengthy answers even if they are wrong. Through this paper, we hope future researchers can focus more on the redundancy of the outputs of LLMs and develop training techniques to teach LLMs when to think step-by-step.

Limitations

The main limitation of our paper is that we only study redundancy on a manually constructed dataset, GSM8K-Zero. The reason is that it is easier to define and calculate redundancy on GSM8K-Zero; we believe this is an ample contribution since it is a phenomenon never mentioned in the literature. While exploring redundancy on other existing datasets will be interesting, we leave it to future works.

Another limitation of our paper is that we rely on ChatGPT and GPT-4 to construct GSM8K-Zero, so noises in the constructed dataset are inevitable. We emphasize that future researchers need to keep the noises in the dataset in mind and take special caution when interpreting the results evaluated on GSM8K-Zero. To understand the noises in the dataset, the authors randomly selected 250 samples from GSM8K-Zero and reviewed them. As stated in Section 2.1, we estimate that 85% of question-answer pairs in GSM8K-Zero are valid. We present the details about our manual review of the dataset in Appendix B.2. We also discuss that our results and observations in the main content still hold when considering the noises in the dataset.

Last, since our paper is a short paper, an obvious limitation is that there is still a lot to explore, but we cannot include them in our paper. While we deem our paper's main content to be self-contained, we include some potential questions that might be raised by curious and enthusiastic readers in Appendix A (FAQs section).

Ethical Statements

We do not see our work to have possible harmful outcomes. We follow the ACL ethical guidelines when conducting the research in this paper.

References

Yuvanesh Anand, Zach Nussbaum, Brandon Duderstadt, Benjamin Schmidt, and Andriy Mulyar. 2023. Gpt4all: Training an assistant-style chatbot with large scale data distillation from gpt-3.5-turbo. https://github.com/nomic-ai/gpt4all.

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report.

Anthropic. 2023. Model card and evaluations for claude models. Accessed on October 1, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

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*.

Olga Golovneva, Moya Peng Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023. ROSCOE: A suite of metrics for scoring step-by-step reasoning. In *The Eleventh International Conference on Learning Representations*.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arxiv:2301.07597*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

OpenAI. 2022. Chatgpt: Optimizing language models for dialogue. Accessed on October 10, 2023.

OpenAI. 2023. Gpt-4 technical report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

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`.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Xi Ye and Greg Durrett. 2022. The unreliability of explanations in few-shot prompting for textual reasoning. In *Advances in Neural Information Processing Systems*.

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

A FAQs

Q1 The accuracy of PaLM does not differ for answers that include calculations and answers that do not. Why is this the case?

A1 By manually scrutinizing the outputs of `text-bison-001` that contains calculations, we observe that `text-bison-001` often first generates an Arabic number as the answer followed by some calculation as the explanation. In this case, the numeric answer of `text-bison-001` does not depend on the calculations, so even if the calculation and reasoning following the answers are wrong, they cannot affect the answer. This makes the accuracy of answers with and without calculation similar for `text-bison-001`.

Q2 This paper only studies RLFH models. What about LLMs that are not RLHF-trained? Do they also show redundancy in their outputs?

A2 Yes, non-RLHF-trained LLMs also show redundancy in their outputs on GSM8K-Zero. We use Alpaca (Taori et al., 2023) and Vicuna (Chiang et al., 2023) and find them to also generate redundant outputs in 40% of the cases. We do not report the results in the main paper since the outputs from Alpaca and Vicuna are quite messy, and it is hard to calculate the accuracy using regular expressions.

Q3 In Section 4, is it possible that the wrong and long answers generated by ChatGPT are correct, making the proxy RMs prefer those long answers? For example, when using regular expressions to calculate accuracy, there might be some cases that regular expressions cannot handle.

A4 This is highly unlikely to happen. This is because one of the authors manually reviews the long answers (100 correct and 100 wrong ones) used in Section 4. Thus, the wrong answers are assured to be wrong, and the correct answers are assured to be correct. Since the authors cannot review all the answers that contain calculations, we only randomly sample approximately 100 correct and 100 wrong answers with calculations and include them in the results in Figure 2.

B More Information about GSM8K-Zero

B.1 Dataset Cards

GSM8K-Zero is constructed from GSM8K (Cobbe et al., 2021). Since GSM8K does not include the dataset license, we are unsure what license to release GSM8K-Zero.

B.2 Manual Review by the Authors

The authors randomly sample 250 samples from GSM8K-Zero to understand the quality of the samples and whether using regular expression to calculate accuracy has a high precision. The human (author) evaluation is conducted in the following steps: First, we randomly sample 125 samples from the answers of ChatGPT that are correct together with their corresponding questions, and we sample 125 samples for the answers of ChatGPT that are incorrect together with their corresponding questions. Recall that the accuracy is calculated using regular expressions. We search for the first or last number that appears in the last sentence of the model's response, and we count the model response to be accurate if the ground truth matches the number extracted by regular expressions. While this process may falsely consider the model to be correct when the model's answer is wrong, we find that this merely happens during our manual review of 250 answers from ChatGPT. We separately sample questions that ChatGPT correctly answered and questions that ChatGPT got wrong because those two groups of questions might be systematically different.

Given a question, an answer from ChatGPT, and the ground truth answer, one of the authors labels the sample into four categories:

1. The ground truth is correct, and the answer from ChatGPT is correct

2. The ground truth is wrong, while the answer from ChatGPT is correct (matches the real ground truth)

3. The ground truth is correct, but the answer from ChatGPT is wrong

4. The question is invalid, including that ground truth is wrong, the question cannot be answered without calculation, or the question is ambiguous.

We find that for questions that ChatGPT is correct, 89% of questions are valid, and the ground truth answer is always correct. However, we find that for 7% of the questions, ChatGPT's answer is wrong, but we count it as correct due to imperfect parsing of regular expressions. For questions that ChatGPT is inaccurate, about 70% of the questions are valid, and the ground truth is wrong in 0.02% of the cases. Only in 0.04% of the cases the regular expression we use considers the answer of ChatGPT to be wrong when it is correct. Considering that ChatGPT's accuracy is about 80%, we estimate that the proportion of invalid questions in GSM8K-Zero is 14.8%.

B.2.1 Does Invalid Questions Affect the Results?

Readers may be concerned about whether the invalid questions change the observations in the main content. The short answer is no. We explain as follows: For the redundancy shown in Table 1, if the model generates CoT reasonings and calculations for those invalid questions, then the redundancy should be around 15%. But clearly, all model except GPT-4 has a redundancy much higher than 15%. Thus, LLMs can still generate a lot of redundant calculations for other valid questions. As for GPT-4, we still find that it does generate redundant calculations in some cases. We show some examples in Figure 3 and 4.

Next, for accuracy, even if LLMs are wrong for all the invalid questions, their accuracy should be around 85% if they get all the valid questions correct. However, this is clearly not the case for all LLMs except Claude-2. Next, for the rightmost column in Table 1, if we assume that all the invalid samples happen to be the samples that LLMs include calculations in the answer, the accuracy in this column should increase. However, by some simple maths, the readers can easily verify that even considering this, the accuracy for answers that contain calculations is still much lower than the accuracy of answers that do not include calculations. Thus, our observation in the paper still holds.

A deep-sea monster rises from the waters once every hundred years to feast on a ship and sate its hunger. Over three hundred years, it has consumed **847** people. Ships have been built larger over time, so each new ship has twice as many people as the last ship. How many people has it consumed over three hundred years?

(a) Question in GSM8K-Zero

The deep-sea monster consumes ships every 100 years, and over 300 years it would have consumed 3 ships. Given that each new ship has twice as many people as the last ship, it means there's a series of doubling.

Let's assume the number of people on the first ship is X.\n\nThen the number of people on the second ship (which is twice the first) would be 2X and the number of people on the third ship (twice the second ship) would be 4X.\n\nGiven that the total number of people consumed over three centuries is 847, we can write the equation:

X + 2X + 4X = 847\n\nThis simplifies to 7X = 847

To find X (the number of people on the first ship), we divide 847 by 7, so X = 121.

So, over three hundred years, the Monster has consumed 121 people on the first ship, 242 people on the second ship, and 484 people on the third ship. The total is 847 people.

(b) GPT-4 output

Figure 3: An example question that makes GPT-4 generates redundant calculations but results in a correct answer.

C Prompts

We list the prompts we use in this section.

Prompts for question generation using ChatGPT in Section 2.1 **System prompt**: You are a helpful assistant. You need to answer the questions of the user accurately. You need to strictly follow the instructions.

User prompt

Your task is to convert a declarative sentence into a question and the answer to that question should be a number. Importantly, the answer (number) to the

A deep-sea monster rises from the waters once every hundred years to feast on a ship and sate its hunger. Over three hundred years, it has consumed **847** people. Ships have been built larger over time, so each new ship has twice as many people as the last ship. How many people has it consumed over three hundred years?

(a) Question in GSM8K-Zero

The deep-sea monster consumes ships every 100 years, and over 300 years it would have consumed 3 ships. Given that each new ship has twice as many people as the last ship, it means there's a series of doubling.

Let's assume the number of people on the first ship is X.\n\nThen the number of people on the second ship (which is twice the first) would be 2X and the number of people on the third ship (twice the second ship) would be 4X.\n\nGiven that the total number of people consumed over three centuries is 847, we can write the equation:

X + 2X + 4X = 847\n\nThis simplifies to 7X = 847

To find X (the number of people on the first ship), we divide 847 by 7, so X = 121.

So, over three hundred years, the Monster has consumed 121 people on the first ship, 242 people on the second ship, and 484 people on the third ship. The total is 847 people.

(b) GPT-4 output

Figure 4: An example question that makes GPT-4 generates redundant calculations but results in a wrong answer.

question should already be included in the original sentence. If the answer need to be obtained by calculation, the question is invalid. Even simple calculation is not allowed. Keep the question as simple as possible. For example:

Example 1:
Original sentence: Alyssa, Keely, and Kendall ordered 100 chicken nuggets from a fast-food restaurant.
Answer (number only): 100
Question: How many chicken nuggets did Alyssa, Keely, and Kendall order?
Explanation: The number 100 already appeared in the original sentence, so the question fulfill the requirements.

Example 2:
Original sentence: Lilah's family gallery has 400 photos.
Answer (number only): 400
Question: How many photos are there in Lilah's family gallery?
Explanation: The number 400 already appeared in the original sentence, so the question fulfill the requirements.

Example 3:
Original sentence: {KNOWN_INFO}
Answer (number only): {ANS}
Question:

The {KNOWN_INFO} should be filled in with the known information in the original question, and the {ANS} should be filled in with the ground truth answer.

Prompts for using ChatGPT and GPT-4 as the proxy in Section 4

System prompt `Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below. You should choose the assistant that follows the user's instructions and answers the user's question better. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of their responses. Begin your evaluation by comparing the two responses and provide a short explanation. Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible. After providing your explanation, output your final verdict by strictly following this format: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie.`

User Prompt

```
[User Question]
{question}
[The Start of Assistant A's Answer]
{answer_a}
[The End of Assistant A's Answer]

[The Start of Assistant B's Answer]
{answer_b}
[The End of Assistant B's Answer]
```

D Sampling parameters of LLMs

When using LLMs to generate the answer to questions in GSM8K-Zero, we set the temperature to 0.7 and keep all the other parameters as default. We use Huggingface Transformers to run Llama-2.