No Need for Explanations: LLMs can *implicitly* learn from mistakes in-context

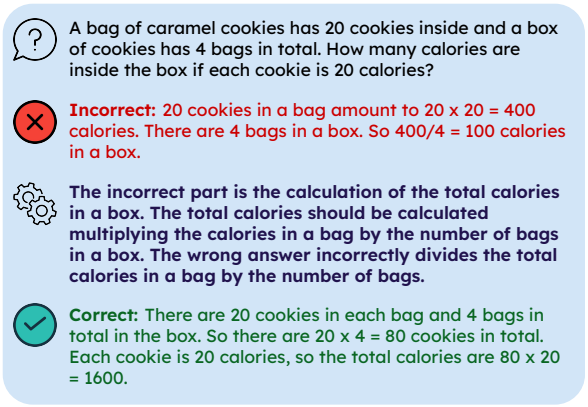
Lisa Alazraki[1]*, **Maximilian Mozes**[2], **Jon Ander Campos**[2], **Tan Yi-Chern**[2], **Marek Rei**[1], **Max Bartolo**[2]
[1]Imperial College London, [2]Cohere
lisa.alazraki20@imperial.ac.uk

Abstract

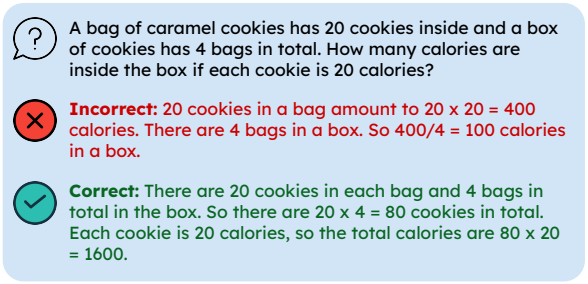
Showing incorrect answers to Large Language Models (LLMs) is a popular strategy to improve their performance in reasoning-intensive tasks. It is widely assumed that, in order to be helpful, the incorrect answers must be accompanied by comprehensive rationales, explicitly detailing where the mistakes are and how to correct them. However, in this work we present a counterintuitive finding: we observe that LLMs perform *better* in math reasoning tasks when these rationales are eliminated from the context and models are left to infer on their own what makes an incorrect answer flawed. This approach also substantially outperforms chain-of-thought prompting in our evaluations. These results are consistent across LLMs of different sizes and varying reasoning abilities. To gain an understanding of *why* LLMs learn from mistakes more effectively without explicit corrective rationales, we perform a thorough analysis, investigating changes in context length and answer diversity between different prompting strategies, and their effect on performance. We also examine evidence of overfitting to the in-context rationales when these are provided, and study the extent to which LLMs are able to autonomously infer high-quality corrective rationales given only incorrect answers as input. We find evidence that, while incorrect answers are more beneficial for LLM learning than additional diverse *correct* answers, explicit corrective rationales over-constrain the model, thus limiting those benefits.

1 Introduction

Adding incorrect answers to the training examples of an LLM has become an established strategy to improve new generations, as these models are able to learn from their and others' mistakes (Madaan et al., 2024; Shinn et al., 2024; An et al., 2023; Paul et al., 2024). Existing literature assumes that in order to learn from incorrect answers effectively, these must be accompanied by explicit corrective rationales locating each error and/or explaining how to rectify it.



(a)



(b)

Figure 1: Learning examples for (a) explicit learning and (b) implicit learning. In explicit learning prompts, corrective feedback follows the incorrect answer (in red) and explains how to derive from it the correct answer (in green). For implicit learning, corrective feedback is discarded and the model is expected to infer the differences between the incorrect and the correct answer.

Previous work finds that when mistakes are shown to the model alongside this corrective feedback—whether in the training data (An et al., 2023; Paul et al., 2024) or in context (Madaan et al., 2024; Shinn et al., 2024)—accuracy improves compared to fine-tuning or prompting with only valid question-answer pairs.

*Work done while at Cohere.

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 33180–33204
November 4-9, 2025

In this work, we investigate the ability of LLMs to learn from mistakes *implicitly*—without the aid of corrective feedback—within an In-Context Learning (ICL) setting. We construct few-shot prompts for questions designed to probe reasoning abilities, alongside incorrect and correct chain-of-thought answers. We provide the incorrect answers without any rationales that would help contextualise them, and let the model autonomously infer the patterns that render an answer wrong or correct. We refer to this strategy as 'prompting for *implicit* learning'. We compare its performance against two baselines: (i) a prompt that includes the same correct and incorrect responses, but where the latter are accompanied by corrective rationales, as in An et al. (2023) (we refer to this strategy as 'prompting for *explicit* learning'), and (ii) a prompt only including the valid step-by-step answers to the same questions, as in the original chain-of-thought implementation (Wei et al., 2024) (we refer to this prompting strategy simply as 'CoT'). We find not only that prompting for implicit learning outperforms vanilla CoT, but—remarkably—that it is also superior to the far more established strategy of prompting for explicit learning. To ensure the robustness of the results, we test all strategies extensively using seven LLMs from four distinct model families and four diverse tasks distributed across four established mathematical reasoning datasets.

Additionally, we carry out a thorough analysis to understand why implicit learning prompts outperform explicit ones. Firstly, we examine whether the greater context length that results from adding incorrect answers to the prompt contributes to the performance improvement—an investigation that was missing in prior literature on learning from mistakes. We insert additional valid question-answer pairs in the vanilla CoT setup, which matches the context length of implicit learning prompts and also provides the model with more diverse examples. We also experiment with showing two correct step-by-step answers for each question in the prompt. In both cases, we find that incorrect answers are more beneficial for LLM performance than additional correct answers. Secondly, to test whether LLMs can indeed infer the patterns that inform correct and incorrect answers implicitly, we have them generate rationales for new incorrect answers under each prompting strategy. We then perform a human evaluation study on the generated rationales. We observe that rationales produced with both implicit and explicit prompting are nearly identical in quality according to annotators. In contrast, rationales produced using only correct CoT answers are substantially lower quality. Lastly, we analyse to what extent LLMs overfit to in-context corrective feedback. We find relatively high similarity scores between the new rationales generated with explicit learning prompting and those provided in-context, indicating that the models are over-constrained by explicit learning prompts. This is further confirmed by our visual inspection of the generated rationales.

In summary, our experiments confirm the findings from previous literature that incorrect answers are particularly beneficial for LLM learning—indeed more so than additional correct answers—yet they also evidence that conditioning the model to in-context rationales, as is currently standard practice, may limit those benefits by adding unnecessary constraints. Since corrective rationales are normally produced by state-of-the-art close-sourced models (An et al., 2023), and are thus expensive to curate at scale, our findings in favour of rationale-free learning have real-world utility for researchers and developers.

Our main contributions are:

1. We investigate *implicit* learning from mistakes with LLMs and compare it to *explicit* learning that uses both mistakes and rationales. To the best of our knowledge, no such investigation has been carried out before, and existing work relies heavily on explicit corrective feedback.

2. We demonstrate that prompting for implicit learning outperforms explicit learning, as well as other strong ICL baselines. This indicates LLMs are well-suited for implicit learning.

3. Our analytical experiments show that, while incorrect answers are beneficial for LLM learning, explicit corrective rationales limit those benefits by over-constraining the model.

4. Our work brings into question the rationale behind rationales and offers a simple yet effective alternative.

2 Related Work

Incorrect answers have been leveraged in prior work to improve LLM responses in challenging tasks. The existing literature can be largely categorised into three approaches: (i) *Self-refinement*, where an LLM critiques its own erroneous generations, (ii) *External feedback*, where corrective

rationales are sourced from a distinct LLM, and (iii) *Multi-agent debate*, where two or more models take turns at providing feedback for a previously generated response.

Self-refinement. The self-refinement pipeline is well exemplified by Madaan et al. (2024). They devise a framework where an LLM first answers a question, then generates feedback for that answer, and finally outputs a new answer based on the feedback. Note that the model is not fine-tuned and each step is elicited via prompting. The refinement process can be repeated multiple times until a stopping criterion is met, to iteratively improve the final answer. In Zhang et al. (2024), not only does the LLM critique its own incorrect answer, but it also summarises what principles can be learned from it. Kim et al. (2024) and Shinn et al. (2024) adopt a similar strategy: the model executes a task and, based on the error signal received from the environment, outputs a self-reflection. This is then added to the LLM context in the next episode.

External feedback. Xu et al. (2024c) observe that self-refinement is inherently biased as LLMs tend to assess their own generations positively. Hence, Xu et al. (2024b) propose a two-model system, where a base LLM answers a question and a fine-tuned model critiques the answer. Similarly, Olausson et al. (2024) find LLM self-critique to be biased in the context of code generation, and show that utilising a second, larger model as the critic allows for more substantial improvements in the task. Tong et al. (2024) feed a corpus of questions and incorrect answers to PaLM2 (Anil et al., 2023), which outputs the type and reasons for each mistake. They show that fine-tuning Flan-T5 models (Chung et al., 2024) on the resulting rationales improves their performance. Similarly, Paul et al. (2024) use corrective feedback from a fine-tuned model as a signal to train a base LLM for producing better responses. An et al. (2023) extend the above approach by collecting LLM-generated incorrect answers and prompting GPT-4 (Achiam et al., 2024) to identify and correct the mistakes. They show that LLMs fine-tuned on this data achieve superior reasoning capabilities.

Multi-agent debate. Since LLMs benefit from a single critic model, it is reasonable to assume that using multiple critics may achieve further improvements. Indeed, Chen et al. (2024) show that a round table of LLMs attains high accuracy in reasoning tasks. In their framework, each LLM produces an answer to a question, followed by a self-critique. Then, all models carry out a multi-turn discussion, revising their answers at each turn based on the responses and self-critiques of the other LLMs. Du et al. (2024) propose a similar framework where multiple instances of an LLM generate candidate answers to a math reasoning question. Each instance then critiques the output of the other models, and uses this to update its answer. Khan et al. (2024) have two models generate different answers and debate their correctness, while the final choice is made by a third LLM witnessing the debate.

Lastly, related work that does not fall into the above categories is Chia et al. (2023)'s contrastive CoT. Using an entity recognition model, they extract and randomly shuffle numbers and equations within a golden mathematical answer to obtain its incoherent counterpart. While this setup shares some similarities with implicit learning due to the absence of a rationale to accompany the incoherent answer, the latter is inherently different from our incorrect reasoning traces. Most saliently, their analysis is not concerned with what and how much information about previous mistakes is required to improve LLM reasoning. In contrast, our investigation stems from the observation that learning from mistakes with LLMs conventionally assumes the need for explicit, fine-grained corrective feedback. We seek to answer the previously unexplored question of whether this additional feedback is actually beneficial or even necessary.

3 Prompt Construction

Let $E = \|_{n=1}^{N} e_n$ be the text sequence resulting from concatenating N in-context examples. In the typical few-shot CoT setting (Wei et al., 2024), an individual example

$$e_n^{\text{CoT}} = \left(q^{(n)}, a^{(n)}\right) \tag{1}$$

is defined by the question $q^{(n)}$ and the corresponding correct step-by-step answer $a^{(n)}$. This can be extended to

$$e_n{}^{\text{explicit}} = \left(q^{(n)}, w^{(n)}, r^{(n)}, a^{(n)}\right) \tag{2}$$

which additionally includes a wrong step-by-step answer $w^{(n)}$ and a rationale $r^{(n)}$ that explicitly identifies the errors in $w^{(n)}$ that need correcting to obtain $a^{(n)}$. This learning setup has been widely

explored in prior literature (Madaan et al., 2024; Kim et al., 2024; Shinn et al., 2024). We additionally consider examples of the form

$$e_n{}^{\text{implicit}} = \left(q^{(n)}, w^{(n)}, a^{(n)}\right) \quad (3)$$

where the explicit rationale $r^{(n)}$ is removed. Figure 1 illustrates instances of (2) and (3).

In our experiments, we set $N = 8$ for all example types. We investigate how the example formulations in the set $\mathcal{E} = \{E^{\text{CoT}}, E^{\text{explicit}}, E^{\text{implicit}}\}$ affect LLMs across different tasks: *labelling* the correctness of an entire answer or an individual reasoning step, *editing* an incorrect answer, or *solving* a new question (we further elaborate on each task in Section 4.3). We do not alter the example format by task, but we instead construct a task-specific prompt by appending an instruction, I, to the examples. I solely depends on the task and not on the type of examples preceding it. Hence, we have a set of task-specific instructions $\mathcal{I} = \{I^{\text{label}_{\text{ans}}}, I^{\text{label}_{\text{step}}}, I^{\text{edit}}, I^{\text{solve}}\}$.

We experiment with all example types for all tasks. That is, we evaluate all prompts in the set $\mathcal{P} = \{E \,||\, I \mid (E, I) \in \mathcal{E} \times \mathcal{I}\}$. Prompts are shown in Appendix A.

3.1 Generating Correct Answers

All the examples in $\mathcal{E}$ include questions and their corresponding correct answers. While questions are provided by the training set, not all datasets contain CoT-style golden answers. In those cases, we generate them by prompting GPT-4 (Achiam et al., 2024) to provide answers in a zero-shot CoT fashion, and inspect both the reasoning trace correctness and final result.

3.2 Generating Incorrect Answers

The incorrect answers necessary to construct the exemplars in E^{explicit} and E^{implicit} are not present in most datasets. To obtain them, we prompt LLMs that are no longer state-of-the-art to generate answers for the training set questions. We use LLaMA 30B (Touvron et al., 2023a), Llama 2 7B (Touvron et al., 2023b) and Llama 3 8B (Grattafiori et al., 2024). The specific model choice depends on the dataset and its difficulty (refer to Appendix C). We use few-shot CoT prompting with all models. We gather answers that are marked as wrong by automated evaluation of the final numerical result. Having discarded empty and partial answers, we simply select the first N incorrect answers in the set and pair them with the corresponding questions and their correct counterparts, obtained as detailed in Section 3.1.

3.3 Generating Corrective Rationales

We generate the corrective rationales in E^{explicit} following a strategy similar to that described in An et al. (2023): we prompt GPT-4 in a few-shot fashion, showing it questions with incorrect and correct answers, as well as rationales. Given a new question and a pair of answers, we ask the model to identify the mistakes in the incorrect answer and explain how to correct them. We use the same few-shot examples as An et al. (2023), slightly reformatted for our task. All rationales are carefully verified for correctness.

4 Experiments

4.1 Models

We study LLMs of different sizes: Command R[1] (35 billion parameters), Llama 3 70B Instruct (Grattafiori et al., 2024) (70 billion parameters), Command R+[1] (104 billion parameters), WizardLM (Xu et al., 2024a) (141 billion parameters). Note that for the Command models, we use both the original and the Refresh versions, as preliminary experiments showed significant differences in their output and results for math reasoning tasks. We also test Titan Text G1 Express[2], whose exact number of parameters has not been publicly disclosed. We note, however, that this model is substantially less capable than the others in reasoning tasks, as evidenced by the lower scores in Table 1. Therefore, we consider seven LLMs in total. We employ a greedy sampling strategy with all models. LLMs are accessed via API; further details including the inference hyperparameters and model IDs are given in Appendix B.

4.2 Datasets

Our main focus is understanding whether LLMs learn implicitly in tasks that require complex reasoning. Contemporary work investigating LLM reasoning has primarily focused on math reasoning as an early and convenient proxy for complex reasoning ability evaluation (Ahn et al., 2024; Paul et al., 2024; Ruis et al., 2025; Liu et al., 2025). Consistent with this approach, we test our method on several math reasoning benchmarks.

[1] https://cohere.com/command

[2] https://aws.amazon.com/bedrock/amazon-models

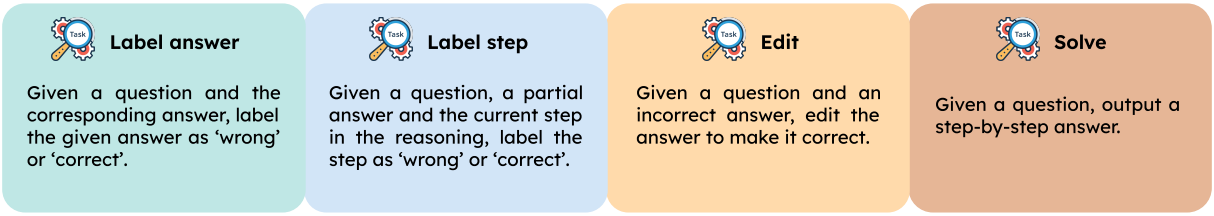


Figure 2: We evaluate LLM performance on four auxiliary tasks: (i) labelling an answer as wrong or correct, (ii) labelling an individual reasoning step, (iii) editing an incorrect answer to make it correct, and (iv) solving a new question. In the labelling tasks, we instruct the model to output a rationale justifying its choice before generating the predicted label. This is motivated by previous work showing that this approach tends to produce more robust labels (Trivedi et al., 2024; Zheng et al., 2024).

GSM8K includes grade-school-level arithmetic problems that require multiple reasoning steps to solve (Cobbe et al., 2021). All problems in GSM8K can be tackled using basic arithmetic operations (addition, subtraction, multiplication, division).

ASDiv contains diverse problems of varying difficulty (Miao et al., 2020). In addition to arithmetic operations, questions can be solved using algebra, number theory, set operations and geometric formulas. They can also require pattern identification and unit conversion.

AQuA is a dataset of algebraic word problems from postgraduate admissions tests such as GRE and GMAT, as well as new questions of similar difficulty collected through crowd-sourcing (Ling et al., 2017). Note that while the original version of the dataset is multiple-choice, here we use a more challenging open-ended version.

PRM800K (Lightman et al., 2024) is derived from the MATH dataset (Hendrycks et al., 2021), which contains challenging competition-level math problems. In PRM800K, model-generated answers to the questions in MATH are paired with human annotations providing a validation signal on intermediary reasoning steps.

These datasets cover a wide range of math domains and difficulty levels, each constituting a particular challenge. Furthermore, statistical analysis on GSM8K, ASDiv and AQuA has determined that these datasets are entirely out-of-domain with respect to one another (Ott et al., 2023), which makes this selection of evaluation datasets an appropriate test bed for our analysis.

4.3 Tasks

In addition to evaluating on diverse math reasoning datasets, we consider auxiliary tasks that can be carried out within those datasets. We illustrate them below and in Figure 2.

Labelling an answer. In this task, we have the model assign a binary label to a CoT-style answer, identifying whether it is correct or not, given the question. Previous work has found that LLM-assigned labels are more robust when they are accompanied by a model-generated rationale (Trivedi et al., 2024; Zheng et al., 2024). Hence, we require LLMs to first output a rationale explaining their choice, followed by the label. Performance in the binary labelling tasks is measured by the macro-averaged F1-score, weighted by support to account for label imbalance. The answers to be labelled are generated by running Llama 2 7B and Llama 3 8B on the test set of each dataset (refer to Appendix C for details).

Labelling a reasoning step. We leverage the step-wise reasoning annotations in PRM800K to have models score the correctness of a single reasoning step given the question and any previous context. Similar to the above setting, the LLM outputs a rationale followed by a binary label ('correct' or 'incorrect'). As the other datasets do not contain step-wise annotations, we perform this task only on PRM800K.

Editing an incorrect answer. We show the model a question and a corresponding incorrect answer, then ask it to output a new, edited answer that leads to the correct solution. Performance is measured by computing the accuracy of the numerical solution. For this task, we use the incorrect portion of the pre-generated answers obtained by running Llama 2 7B and Llama 3 8B on the test sets.

Solving a math question. We show the model a test set question and ask it to output the solution. As in the previous task, we compute the accuracy of the final numerical solution.

Model	Strategy	GSM8K			ASDiv			AQuA			PRM800K			
		$\text{label}_{\text{ans}}$	edit	solve	$\text{label}_{\text{ans}}$	edit	solve	$\text{label}_{\text{ans}}$	edit	solve	$\text{label}_{\text{ans}}$	$\text{label}_{\text{step}}$	edit	solve
Llama 3 70B Instruct	CoT	$83.8_{0.5}$	$81.3_{0.3}$	$91.8_{0.5}$	$90.1_{0.3}$	$82.7_{0.8}$	$90.8_{0.3}$	$\mathbf{66.6}_{1.1}$	$37.3_{0.4}$	$55.8_{0.6}$	$\mathbf{31.7}_{0.6}$	$49.6_{0.2}$	$20.6_{1.3}$	$43.9_{0.8}$
	Explicit	$82.5_{0.6}$	$84.2_{0.2}$	$92.8_{0.3}$	$90.0_{0.3}$	$81.4_{0.9}$	$\mathbf{91.5}_{0.1}$	$55.7_{1.1}$	$34.0_{1.1}$	$55.1_{1.0}$	$19.0_{0.4}$	$48.2_{0.4}$	$21.8_{1.6}$	$48.1_{0.4}$
	Implicit	$\mathbf{84.0}_{0.7}$	$\mathbf{84.8}_{0.1}$	$\mathbf{93.3}_{0.4}$	$\mathbf{91.4}_{0.5}$	$\mathbf{84.9}_{0.7}$	$91.1_{0.2}$	$56.6_{0.2}$	$\mathbf{37.6}_{1.3}$	$\mathbf{56.4}_{0.4}$	$19.2_{0.2}$	$\mathbf{50.0}_{0.6}$	$\mathbf{26.5}_{1.9}$	$\mathbf{48.4}_{0.6}$
Command R	CoT	$50.5_{0.8}$	$17.2_{0.8}$	$\mathbf{63.1}_{0.5}$	$53.4_{1.1}$	$49.3_{1.2}$	$\mathbf{77.6}_{0.7}$	$37.1_{0.8}$	$7.9_{1.3}$	$\mathbf{21.9}_{1.1}$	$21.4_{0.7}$	$36.3_{0.4}$	$4.7_{1.2}$	$13.3_{1.0}$
	Explicit	$57.0_{0.8}$	$25.1_{1.3}$	$56.7_{1.0}$	$\mathbf{64.1}_{1.2}$	$48.0_{1.3}$	$69.6_{1.0}$	$34.2_{1.6}$	$6.7_{1.1}$	$17.8_{2.0}$	$32.7_{0.6}$	$39.0_{0.2}$	$7.5_{0.8}$	$13.0_{0.7}$
	Implicit	$\mathbf{64.2}_{0.8}$	$\mathbf{31.1}_{1.2}$	$60.5_{1.2}$	$60.3_{0.7}$	$\mathbf{51.4}_{0.7}$	$70.1_{0.6}$	$\mathbf{39.8}_{1.3}$	$\mathbf{11.2}_{1.2}$	$19.1_{0.3}$	$\mathbf{56.0}_{0.9}$	$\mathbf{43.4}_{1.2}$	$\mathbf{8.8}_{0.4}$	$\mathbf{14.8}_{0.7}$
Command R+	CoT	$65.8_{0.7}$	$48.0_{0.3}$	$69.7_{0.5}$	$78.9_{0.6}$	$61.9_{1.1}$	$81.7_{0.6}$	$43.8_{0.9}$	$11.9_{1.2}$	$32.0_{1.6}$	$16.1_{1.1}$	$35.8_{0.4}$	$14.5_{1.1}$	$\mathbf{23.9}_{1.3}$
	Explicit	$64.3_{0.3}$	$59.8_{0.6}$	$76.0_{0.9}$	$80.4_{0.3}$	$69.3_{1.2}$	$83.9_{0.4}$	$46.5_{0.4}$	$12.5_{1.4}$	$31.1_{2.0}$	$\mathbf{59.7}_{0.8}$	$38.8_{0.2}$	$12.9_{1.0}$	$18.1_{0.6}$
	Implicit	$\mathbf{71.9}_{0.4}$	$\mathbf{62.0}_{0.8}$	$\mathbf{79.9}_{0.8}$	$\mathbf{82.6}_{0.2}$	$\mathbf{70.7}_{1.3}$	$\mathbf{85.3}_{0.3}$	$\mathbf{47.6}_{0.9}$	$\mathbf{16.8}_{1.1}$	$\mathbf{35.8}_{1.1}$	$59.5_{1.2}$	$\mathbf{39.2}_{0.2}$	$\mathbf{16.6}_{0.8}$	$21.1_{0.6}$
Command R Refresh	CoT	$55.5_{0.9}$	$52.1_{0.6}$	$78.9_{0.4}$	$54.8_{0.3}$	$64.8_{0.2}$	$84.5_{0.6}$	$47.5_{1.4}$	$8.5_{1.0}$	$35.3_{1.2}$	$68.1_{0.7}$	$39.3_{0.6}$	$11.7_{0.7}$	$30.6_{0.7}$
	Explicit	$48.7_{1.1}$	$55.9_{0.4}$	$75.9_{0.8}$	$37.9_{0.6}$	$69.2_{1.1}$	$80.9_{0.6}$	$42.4_{0.4}$	$16.5_{0.5}$	$39.1_{1.8}$	$67.3_{0.9}$	$\mathbf{55.9}_{0.6}$	$\mathbf{13.1}_{0.8}$	$30.8_{0.5}$
	Implicit	$\mathbf{62.5}_{1.0}$	$\mathbf{57.4}_{0.7}$	$\mathbf{79.2}_{0.7}$	$\mathbf{70.4}_{1.1}$	$\mathbf{72.2}_{0.3}$	$\mathbf{84.8}_{0.5}$	$\mathbf{50.7}_{0.9}$	$\mathbf{16.6}_{0.6}$	$\mathbf{40.5}_{0.8}$	$\mathbf{71.1}_{0.9}$	$53.7_{0.8}$	$11.8_{1.1}$	$\mathbf{32.1}_{0.7}$
Command R+ Refresh	CoT	$46.9_{0.9}$	$45.9_{0.7}$	$75.6_{0.9}$	$77.7_{0.1}$	$78.8_{0.5}$	$89.4_{0.4}$	$61.0_{0.4}$	$\mathbf{23.5}_{0.7}$	$44.5_{0.9}$	$54.5_{1.2}$	$\mathbf{51.9}_{0.6}$	$16.1_{1.3}$	$\mathbf{31.9}_{0.8}$
	Explicit	$40.3_{0.8}$	$57.6_{1.3}$	$82.0_{0.7}$	$64.8_{0.2}$	$76.1_{0.9}$	$89.9_{0.3}$	$53.6_{0.5}$	$20.6_{1.7}$	$43.2_{1.7}$	$73.3_{0.4}$	$51.7_{0.9}$	$15.9_{1.3}$	$26.8_{0.4}$
	Implicit	$\mathbf{47.2}_{0.8}$	$\mathbf{62.8}_{0.9}$	$\mathbf{86.3}_{0.8}$	$\mathbf{79.6}_{0.2}$	$\mathbf{81.9}_{1.0}$	$\mathbf{90.4}_{0.4}$	$\mathbf{63.3}_{1.2}$	$21.5_{1.1}$	$\mathbf{47.8}_{1.2}$	$\mathbf{73.9}_{0.8}$	$47.8_{0.7}$	$\mathbf{18.7}_{1.1}$	$29.7_{0.7}$
Titan Text G1 Express	CoT	$53.4_{0.2}$	$2.1_{0.3}$	$29.7_{0.7}$	$45.4_{0.6}$	$\mathbf{13.1}_{0.3}$	$61.2_{0.4}$	$\mathbf{54.9}_{1.4}$	$1.0_{0.2}$	$7.9_{0.6}$	$\mathbf{67.2}_{0.4}$	$46.4_{0.5}$	$3.0_{0.3}$	$5.4_{0.2}$
	Explicit	$68.7_{0.3}$	$2.8_{0.4}$	$33.2_{0.5}$	$53.0_{0.8}$	$12.9_{0.3}$	$61.1_{0.6}$	$43.2_{1.3}$	$1.2_{0.2}$	$8.8_{0.7}$	$35.5_{0.4}$	$48.9_{0.4}$	$3.0_{1.0}$	$\mathbf{7.6}_{0.4}$
	Implicit	$\mathbf{69.3}_{0.5}$	$\mathbf{3.0}_{0.4}$	$\mathbf{34.7}_{0.9}$	$\mathbf{60.2}_{0.4}$	$12.9_{0.1}$	$\mathbf{62.8}_{0.4}$	$49.6_{0.8}$	$\mathbf{1.4}_{0.1}$	$\mathbf{11.0}_{0.2}$	$45.7_{0.9}$	$\mathbf{49.7}_{1.0}$	$\mathbf{4.0}_{0.5}$	$5.8_{0.5}$
WizardLM	CoT	$85.9_{0.4}$	$72.4_{1.1}$	$91.3_{0.6}$	$93.8_{0.2}$	$80.6_{0.3}$	$90.9_{0.4}$	$76.6_{0.2}$	$41.7_{0.2}$	$59.1_{0.7}$	$68.4_{0.6}$	$61.4_{0.4}$	$\mathbf{36.8}_{1.2}$	$60.3_{0.9}$
	Explicit	$85.4_{0.5}$	$78.5_{1.3}$	$\mathbf{91.5}_{0.9}$	$93.5_{0.3}$	$82.4_{0.2}$	$\mathbf{91.4}_{0.1}$	$77.3_{0.5}$	$43.2_{0.4}$	$58.0_{1.1}$	$62.5_{0.4}$	$60.2_{0.3}$	$36.5_{1.5}$	$60.8_{0.7}$
	Implicit	$\mathbf{86.9}_{0.5}$	$\mathbf{82.3}_{0.9}$	$\mathbf{91.5}_{0.7}$	$\mathbf{95.0}_{0.2}$	$\mathbf{82.6}_{0.2}$	$90.6_{0.3}$	$\mathbf{79.5}_{0.1}$	$\mathbf{44.1}_{0.5}$	$\mathbf{62.1}_{0.6}$	$\mathbf{69.0}_{0.4}$	$\mathbf{62.6}_{0.5}$	$35.7_{1.3}$	$\mathbf{61.1}_{0.6}$

Table 1: Results of CoT prompting, *explicit* learning prompting, and *implicit* learning prompting for different LLMs on four math reasoning benchmarks. We use the benchmarks for the following tasks: (i) labelling an answer as wrong or correct ($\text{label}_{\text{ans}}$), (ii) labelling a single reasoning step as wrong or correct ($\text{label}_{\text{step}}$), (iii) editing an incorrect answer (edit), and (iv) solving a question (solve). We report the accuracy of the final numerical result for all tasks except the two labelling tasks, where we report the weighted F1-score of the binary label. Due to small variations likely resulting from dynamic batching in the APIs, we report results averaged over five runs. Confidence intervals are shown in Appendix D.

To encourage the models to output responses *conditioned* on the context, as opposed to text that merely mimics the format of the examples in it, we append the task-specific instruction after the examples. We further aid generalisation by prepending the text *'Now apply what you have learned'* to the instruction. Mao et al. (2024) show that the position of the instruction within a few-shot prompt affects the model's behaviour and performance. On the other hand, the model may still be inclined to generate responses in the format of the examples (e.g., when tasked with editing an answer, having observed examples that contain corrective rationales, the model may output a rationale before the corrected answer). To account for this possibility without unnecessarily penalising any particular prompting strategy, we provide a large generation window of 4096 tokens.

4.4 Results

We find that CoT and prompting with *explicit* rationales have similar overall performance on the answer labelling task and when solving new questions, while the latter outperforms CoT when labelling reasoning steps (+3.2%, averaged across all models and all datasets) and editing an incorrect answer (+2.1% avg.). This advantage is aligned with previous findings that LLMs benefit from observing incorrect answers and corrective feedback in their context. On the other hand, prompting

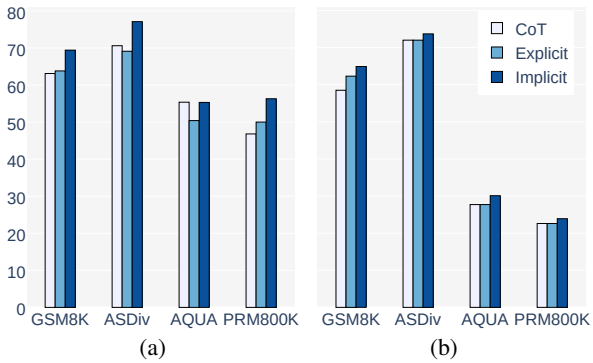


Figure 3: Scores per dataset of CoT, explicit and implicit prompting for (a) the weighted F1-score of the labelling task, and (b) the averaged accuracy across the editing and solving tasks. Scores are averaged across all LLMs.

for *implicit* learning achieves the highest overall performance, as evidenced in Table 1. When considering all combinations of model, dataset and task, implicit learning outperforms CoT in 85% of cases. It also outperforms explicit learning in 88% of cases. In nearly half of these, the advantage of implicit over explicit learning is substantial—well above 3%. This advantage is present even in tasks where, intuitively, we would expect in-context rationales to be particularly helpful, for example when editing an incorrect answer to make it correct. In fact, implicit learning gives the largest accuracy boost in the editing task: $+4.4\%$ over CoT and $+2.2\%$ over explicit learning, averaged across all models and datasets. On the solving task, its accuracy increases by 1.6 and 1.9 percentage points, respectively. Labelling answers also benefits from implicit learning prompts, with averaged F1-scores 5.6% above CoT and 6.2% above explicit learning. Finally, looking at the individual datasets, implicit learning gains the most on GSM8K, where it outperforms both explicit learning and CoT in over 90% of cases across all models and tasks. This proportion is 76% on ASDiv, 81% on AQuA and 64% on PRM800K. Note that the questions in GSM8K and ASDiv have a lower level of difficulty than those in AQuA and PRM800K, as evidenced by the performance differences across all LLMs. Generally, we observe that prompting for implicit learning improves performance across varying levels of difficulty, as shown in Figure 3. In the labelling task (Figure 3a), implicit learning gives the most substantial performance gains on ASDiv and PRM800K. When editing an incorrect answer and solving a new question (Figure 3b), on the other hand, it is GSM8K and AQuA that benefit the most from this strategy.

5 Analysis

To understand *why* implicit learning leads to the improved performance observed above, we carry out a thorough analysis. We investigate context length and answer diversity, and draw insights from new rationales generated under each prompting strategy.

5.1 Effect of Context Length and Diversity

Adding incorrect answers to a prompt introduces additional tokens into the context. As a result, there is a mismatch between the context length of CoT and that of implicit learning. Since an extended context length can in itself be responsible for improved performance, we investigate whether the additional tokens may be driving the improvement, rather than the presence of incorrect answers. We thus compare implicit learning with two distinct extended-context baselines: a few-shot CoT prompt containing additional valid question-answer pairs (we refer to this setup as CoT+), and one where two correct step-by-step answers are shown for each question (we refer to this as CoT-2).

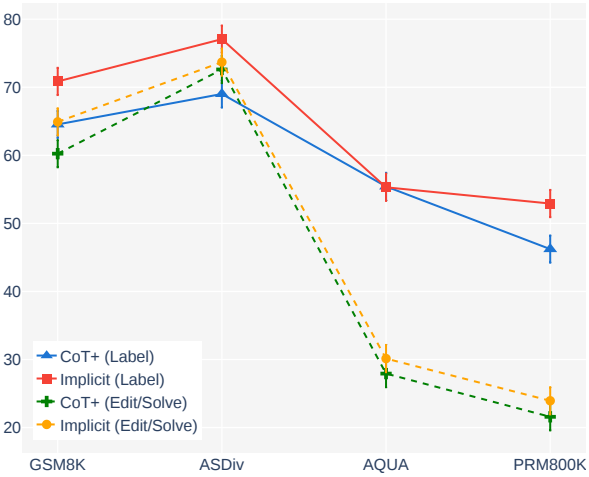


Figure 4: Scores per dataset of CoT+ and implicit prompting for (a) the weighted F1-score of the labelling task, and (b) the averaged accuracy across the editing and solving tasks. Scores are averaged across all LLMs.

CoT+ extends the context by increasing the number of examples in our CoT prompt from eight to fourteen. Additional examples are randomly selected from an identical distribution to the original eight examples. We compare this setup to implicit learning prompts containing eight few-shot examples as in our standard experimental setting. The addition of six new in-context examples to the CoT prompt results in an approximately equal context length between the two settings. It also constitutes a particularly strong baseline, since the new examples may provide the model with additional, novel scenarios to learn from. Figure 4 summarises the results of this comparison (results are shown in full in Table 6). With the exception of the AQuA labelling task, where the two methods perform equally, the average performance of implicit learning is above that of CoT+ on all datasets and tasks, often substantially (3+% improvement). This demonstrates that, at equal context length, the addition of incorrect answers is more beneficial for LLMs than the inclusion of additional diverse and valid question-answer pairs.

Strategy	Avg. human evaluation score
CoT	0.68
Explicit	1.01
Implicit	0.98

Table 2: Human evaluation of model-generated rationales. Rationales produced with explicit and implicit learning prompts obtain similar overall evaluations. CoT prompting results in substantially worse-quality rationales being generated.

Strategy	Avg. n-gram similarity
CoT	0.086
Explicit	0.152
Implicit	0.093

Table 3: Average n-gram similarity between exemplar and generated rationales for each prompting strategy.

CoT-2 provides two correct CoT answers for each in-context question. We show the differences in performance between CoT-2 and implicit learning, computed on Command R+, in Table 7. We note that implicit learning outperforms CoT-2 in almost all cases, with average performance gains of +1.5% in the labelling task (measured in F1-score) and +4.4% in the editing and solving tasks (measured in accuracy). These results indicate that LLMs prompted for implicit learning gain a better understanding of the patterns that inform correct answers—and how these differ from incorrect answers—which prompting with only correct reasoning traces may not sufficiently elicit.

5.2 Analysis of Generated Rationales

A follow-up research question aims to investigate how incorporating error information affects model outputs. We hypothesise that if the LLMs are incorporating error signal implicitly to improve reasoning, this should also be reflected in the downstream generated rationales. We thus ask the models to generate rationales for new incorrect answers under each prompting strategy. We assess and compare them through human evaluation, and examine whether and to what extent LLMs overfit to in-context rationales when these are provided. Finally, we inspect all rationales visually and give an overview of their representative characteristics.

Rationale quality. To ascertain whether, and to what extent, LLMs infer implicit information between incorrect and correct answers with different prompting strategies, we carry out a blind human evaluation study of rationales generated using distinct prompts. We randomly select 300 rationales generated by running the answer labelling task on GSM8K. We select 100 rationales for each prompting strategy (CoT, explicit learning, implicit learning), and have four annotators with domain expertise score them as *0–Poor*, *1–Fair* or *2–Good*. Table 2 illustrates the average human evaluation scores achieved under each prompting strategy. We observe that CoT's performance is considerably lower than either explicit or implicit learning, with an average score of 0.68. The performance of explicit and implicit learning is similar (1.01 and 0.98 respectively). It is noteworthy that rationales generated with implicit learning prompts achieve an average score that is within only 0.03 of that achieved by explicit learning. This is evidence that LLMs can infer high-quality corrective rationales implicitly, simply observing correct and incorrect answers side by side, and that the effect of adding example rationales to the context is negligible. To understand how much annotators agree with each other when assessing the different prompting strategies, we measure the proportion of 'poor', 'fair', and 'good' qualitative labels assigned by each annotator to rationales generated with CoT, explicit and implicit prompting respectively. For each prompting method, we compute the median absolute deviation (MAD) of each label across all annotators. We find the MAD to be within 0.05 in all cases, indicating that annotators largely agree with each other on the overall quality of the rationales generated with each prompting strategy, and the proportions of qualitative labels they assign to the rationales obtained with a given strategy are fairly similar. In Appendix G, we show the fine-grained proportions of labels assigned by the annotators to each strategy.

Rationale similarity. A plausible reason why learning with explicit corrective feedback underperforms implicit learning is that LLMs may be over-constrained by the rationales. To validate this hypothesis, we investigate how similar the new rationales generated by the models are to the in-context ones. As shown in Table 3, the average n-gram similarity score ($n = 2$) of rationales generated by LLMs prompted for explicit learning is substantially higher than that obtained with rationales output with the other methods (note that the

other methods do not include exemplar rationales in the context). Rouge-1 and Rouge-L scores, shown in Appendix F, follow a similar trend. It thus appears that LLMs tend to copy the patterns in the exemplar rationales when these are provided. This suggests that overfitting may be responsible for the lower performance of explicit learning.

Rationale length and appearance. We inspect the generated rationales and find that those generated via explicit learning prompting tend to be more verbose. As a representative example, consider the math reasoning problem *"There are 4 snails in one aquarium and 32 snails in another aquarium. The difference between the number of snails in the two aquariums is twice the amount of fish in both aquariums. If both aquariums have the same number of fish in them, how many fish are there in each aquarium?"*. We take an incorrect, model-generated answer to this problem which assigns the unknown number of fish to the variable x but does not proceed to solve for x. We show three representative rationales generated for this question-answer pair using the three prompting strategies: CoT (Rationale 1), explicit learning (Rationale 2) and implicit learning (Rationale 3). We observe that the rationale produced using CoT fails to identify the error. It also hallucinates that the number of fish in each aquarium is 14, which is neither the correct solution nor a value that appears in the answer. In contrast, prompting for both explicit and implicit learning produces accurate rationales. Note that the latter—generated without exemplar rationales in the context to use as guidelines—is more succinct yet equally exhaustive. Indeed, we observe that rationales generated via explicit learning prompting are substantially longer on average (373 average length, in characters), similar to those shown in-context (423 average length), which further confirms the overfitting hypothesis. In contrast, rationales produced with implicit learning prompting are over a third shorter (237 average length).

Rationale 1: CoT

```
The provided answer is correct as it solves the word problem correctly by letting the number of fish in each aquarium be x. The answer follows the narrative of the problem and uses the information that the difference in snails between the two aquariums is twice the amount of fish in both aquariums. The solution is coherent and arrives at the conclusion that each aquarium has 14 fish.
```

Rationale 2: Explicit

```
The provided answer is incorrect because it does not actually provide a numerical value for the number of fish in each aquarium, which is what the question is asking for. Instead, it repeats the expression "Let x be the number of fish in each aquarium" multiple times, which is not a valid answer. The answer also does not explain how the problem's conditions are reflected in the solution, which is twice the number of fish in each aquarium is equal to the difference in the number of snails between the two aquariums.
```

Rationale 3: Implicit

```
The provided answer is incorrect because the solution fails to provide the final calculation to determine the number of fish in each aquarium. The answer assumes the role of x as the number of fish in each aquarium but does not conclude the equation.
```

6 Conclusion

We have investigated in-context *implicit* learning from mistakes across a range of LLM families and sizes, and found that it outperforms both chain-of-thought prompting and *explicit* learning in challenging math reasoning tasks. Our analysis shows that although incorrect answers benefit LLMs more than additional correct ones, providing explicit corrective feedback limits those advantages, as models tend to overfit to it. Our findings are as noteworthy as they are surprising, since they call into question the benefits of widely used corrective rationales to aid LLMs in learning from mistakes. These rationales are prevalent in current frameworks despite being expensive to curate at scale, yet our investigation suggests that they are redundant and can even hurt performance by adding unnecessary constraints.

Limitations

We have carried out an exhaustive investigation of implicit learning from mistakes, focused on in-context learning. It is worth noting that implicit learning examples—which consist of triples of the form *(question, incorrect answer, correct answer)*—can be obtained at scale by simply running more and less capable LLMs on training set questions. This opens up the possibility of investigating performance differences between explicit and implicit learning also in other paradigms, such as in the fine-tuning setting. Future work can investigate whether the results established in this paper extend to models fine-tuned using similar strategies.

In our experiments, we use four datasets covering different math topics and difficulty levels, extract multiple subtasks from each dataset, use four prompting strategies (CoT, CoT+, implicit, explicit), and seven LLMs. This totals 364 distinct experimental setups, each run five times for robustness. Given the extensiveness of our experiments, it was infeasible to explore further domains other than math reasoning within the scope of this work. Math benchmarks were chosen as a reliable proxy for LLM reasoning in accordance with established prior literature (Ahn et al., 2024; Paul et al., 2024; Ruis et al., 2025; Liu et al., 2025). While there exists prior work investigating related topics with greater breadth (Lampinen et al., 2022), we leave similar investigations in other domains to future work.

Ethical Considerations

This study relies solely on established, publicly available math reasoning benchmarks and focuses on evaluating different prompting strategies. As such, it does not involve sensitive data or foreseeable ethical risks. Our use of the models and datasets described in this paper complies with all applicable licenses.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng,

Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. GPT-4 technical report. *Preprint*, arXiv:2303.08774.

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 225–237, St. Julian's, Malta. Association for Computational Linguistics.

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. 2023. Learning from mistakes makes LLM better reasoner. *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, 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. *Preprint*, arXiv:2305.10403.

Justin Chen, Swarnadeep Saha, and Mohit Bansal. 2024. ReConcile: Round-table conference improves reasoning via consensus among diverse LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7066–7085, Bangkok, Thailand. Association for Computational Linguistics.

Yew Ken Chia, Guizhen Chen, Luu Anh Tuan, Soujanya Poria, and Lidong Bing. 2023. Contrastive chain-of-thought prompting. *Preprint*, arXiv:2311.09277.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tai, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2024. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.*, 25(1).

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Yilun Du, Shuang Li, Antonio Torralba, Joshua B. Tenenbaum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multiagent debate. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen,

Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lind-

say, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. 2024. The Llama 3 herd of models. *Preprint*, arXiv:2407.21783.

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. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R. Bowman, Tim Rocktäschel, and Ethan Perez. 2024. Debating with more persuasive LLMs leads to more truthful answers. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Geunwoo Kim, Pierre Baldi, and Stephen McAleer. 2024. Language models can solve computer tasks. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Andrew Lampinen, Ishita Dasgupta, Stephanie Chan, Kory Mathewson, Mh Tessler, Antonia Creswell, James McClelland, Jane Wang, and Felix Hill. 2022. Can language models learn from explanations in context? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 537–563, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *International Conference on Representation Learning*, volume 2024, pages 39578–39601.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.

Junnan Liu, Hongwei Liu, Linchen Xiao, Ziyi Wang, Kuikun Liu, Songyang Gao, Wenwei Zhang, Songyang Zhang, and Kai Chen. 2025. Are your LLMs capable of stable reasoning? In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 17594–17632, Vienna, Austria. Association for Computational Linguistics.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2024. Self-refine: iterative refinement with self-feedback. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Junyu Mao, Stuart E. Middleton, and Mahesan Niranjan. 2024. Do prompt positions really matter? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4102–4130, Mexico City, Mexico. Association for Computational Linguistics.

Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. A diverse corpus for evaluating and developing English math word problem solvers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 975–984, Online. Association for Computational Linguistics.

Theo X. Olausson, Jeevana Priya Inala, Chenglong Wang, Jianfeng Gao, and Armando Solar-Lezama. 2024. Is self-repair a silver bullet for code generation? In *International Conference on Representation Learning*, volume 2024, pages 36545–36593.

Andreas Opedal, Haruki Shirakami, Bernhard Schölkopf, Abulhair Saparov, and Mrinmaya Sachan. 2025. MathGAP: Out-of-distribution evaluation on problems with arbitrarily complex proofs. In *International Conference on Representation Learning*, volume 2025, pages 100465–100487.

Simon Ott, Konstantin Hebenstreit, Valentin Liévin, Christoffer Egeberg Hother, Milad Moradi, Maximilian Mayrhauser, Robert Praas, Ole Winther, and Matthias Samwald. 2023. ThoughtSource: A central hub for large language model reasoning data. *Scientific Data*, 10:528.

Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. 2024. REFINER: Reasoning feedback on intermediate representations. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1100–1126, St. Julian's, Malta. Association for Computational Linguistics.

Laura Ruis, Maximilian Mozes, Juhan Bae, Siddhartha Rao Kamalakara, Dwaraknath Gnaneshwar, Acyr Locatelli, Robert Kirk, Tim Rocktaeschel, Edward Grefenstette, and Max Bartolo. 2025. Procedural knowledge in pretraining drives reasoning in large language models. In *International Conference on Representation Learning*, volume 2025, pages 29367–29429.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: language agents with verbal reinforcement learning. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Yongqi Tong, Dawei Li, Sizhe Wang, Yujia Wang, Fei Teng, and Jingbo Shang. 2024. Can LLMs learn from previous mistakes? investigating LLMs' errors to boost for reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3065–3080, Bangkok, Thailand. Association for Computational Linguistics.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. LLaMA: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

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. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Prapti Trivedi, Aditya Gulati, Oliver Molenschot, Meghana Arakkal Rajeev, Rajkumar Ramamurthy, Keith Stevens, Tanveesh Singh Chaudhery, Jahnavi Jambholkar, James Zou, and Nazneen Rajani. 2024. Self-rationalization improves LLM as a fine-grained judge. *Preprint*, arXiv:2410.05495.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2024. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, Qingwei Lin, and Daxin Jiang. 2024a. WizardLM: Empowering large pre-trained language models to follow complex instructions. In *International Conference on Representation Learning*, volume 2024, pages 30745–30766.

Wenda Xu, Daniel Deutsch, Mara Finkelstein, Juraj Juraska, Biao Zhang, Zhongtao Liu, William Yang Wang, Lei Li, and Markus Freitag. 2024b. LLMRefine: Pinpointing and refining large language models via fine-grained actionable feedback. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1429–1445, Mexico City, Mexico. Association for Computational Linguistics.

Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Wang. 2024c. Pride and prejudice: LLM amplifies self-bias in self-refinement. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15474–15492, Bangkok, Thailand. Association for Computational Linguistics.

Tianjun Zhang, Aman Madaan, Luyu Gao, Steven Zheng, Swaroop Mishra, Yiming Yang, Niket Tandon, and Uri Alon. 2024. In-context principle learning from mistakes. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Jiachen Zhao. 2023. In-context exemplars as clues to retrieving from large associative memory. In *Neural Conversational AI Workshop - What's left to TEACH (Trustworthy, Enhanced, Adaptable, Capable and Human-centric) chatbots?* ICML'23.

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. 2024. Judging LLM-as-a-judge with MT-bench and Chatbot Arena. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

A Prompts

All prompts are of the form

`[examples]\n\n\n\n[instruction]`,

where the *examples* are specific to the prompting strategy (i.e., CoT, explicit learning, implicit learning), while the *instruction* is task-dependent (i.e., labelling an answer or step, editing an incorrect answer, solving a new question). In the next sections, we illustrate the examples for each prompting strategy and the instructions for each task.

A.1 Examples

All few-shot examples shown below are for GSM8K. Examples for all datasets are included at https://github.com/lisaalaz/implicit-learning-with-llms.

CoT Examples. Below you will find some questions followed by the answer.

Question: A class has 32 students and they all need to present their projects. Every period is 40 minutes long. How many periods will it take for every student to present their project if they have 5 minutes to do so?

Answer: There are 32 students and each student needs 5 minutes to present their project.
So the total time needed is 32 x 5 = 160 minutes.
There are 40 minutes in each period.
So the number of periods needed is 160/40 = 4.
So, the final answer is 4.

Question: A bag contains 50 fewer baseball cards than football cards. There are 4 times as many football cards as hockey cards. If there are 200 hockey cards in the bag, how many cards are there altogether?

Answer: There are 4 times as many football cards as hockey cards, so there are 4 x 200 = 800 football cards.
There are 50 fewer baseball cards than football cards, so there are 800 - 50 = 750 baseball cards.
There are 200 hockey cards, 800 football cards and 750 baseball cards for a total of 200 + 800 + 750 = 1750 cards.
So, the final answer is 1750.

Question: A bag of caramel cookies has 20 cookies inside and a box of cookies has 4 bags in total. How many calories are inside the box if each cookie is 20 calories?

Answer: There are 20 cookies in each bag and 4 bags in total.
So there are 20 x 4 = 80 cookies in total.
Each cookie is 20 calories, so the total calories are 80 x 20 = 1600 calories.
So, the final answer is 1600.

Question: There are four members in one household. Each member consumes 3 slices of bread during breakfast and 2 slices of bread for snacks. A loaf of bread has 12 slices. How many days will five loaves of bread last in this family?

Answer: Each member consumes 3+2 = 5 slices of bread per day.
There are 4 members so they consume 5 x 4 = 20 slices of bread per day.
Each loaf has 12 slices so 5 loaves have 5 x 12 = 60 slices.
60 slices of bread will last 60/20 = 3 days.
So, the final answer is 3.

Question: Bill decides to bring donuts to work for a meeting that day. He buys a box of donuts that has 50 in total in them. He eats 2 on the ride in because he's hungry. When he gets to the office, the secretary takes another 4 out of the box when he's not looking. Lastly, right before the meeting Bill sets the box down on his desk and leaves the office to take a phone call. While Bill's away, his coworkers steal half the remaining donuts. Bill comes back and brings the box into his meeting. How many donuts are left in the box?

Answer: Bill eats 2 donuts on the ride in, leaving 50 - 2 = 48 donuts.
The secretary eats 4 donuts, leaving 48 - 4 = 44 donuts.
Bill's coworkers eat half of the remaining donuts, which is 44/2 = 22 donuts.
So Bill has 44 - 22 = 22 donuts left.
So, the final answer is 22.

Question: Boris has 100 pieces of Halloween candy. His daughter eats 8 pieces of candy. He separates the remaining pieces of candy into equal portions into 4 different bowls. Then he takes away 3 pieces of candy from each bowl to keep for himself. How many pieces of candy are in one bowl?

Answer: Boris has 100 - 8 = 92 pieces of candy

left.
He divides the 92 pieces of candy into 4 bowls. 92/4 = 23 pieces of candy in each bowl.
Then he takes away 3 pieces of candy from each bowl. 23 - 3 = 20 pieces of candy in each bowl.
So, the final answer is 20.

Question: Yvonne and Janna were writing their 1000-word pair research paper. Yvonne was able to write 400 words while Janna wrote 150 more words than Yvonne. When they edited their paper, they removed 20 words and added twice as many words as they removed. How many more words should they add to reach the research paper requirement?

Answer: Yvonne wrote 400 words and Janna wrote 150 more words than Yvonne, so Janna wrote 400 + 150 = 550 words.
Together they wrote 400 + 550 = 950 words.
After editing, they removed 20 words and added 2 x 20 = 40 words. So they have 950 - 20 + 40 = 970 words.
They need 1000 words, so they need 1000 - 970 = 30 more words.
So, the final answer is 30.

Question: Luke started working on a 1000-piece jigsaw puzzle. The first day he worked on it, he put together 10 percent of the pieces. On the second day, he put together another 20 percent of the remaining pieces from the first day. On the third day, he put together 30 percent of the remaining pieces from the previous day. How many pieces does he have left to complete after the third day?

Answer: 10 percent of 1000 is 100.
So he has 100 pieces done after the first day. He has 900 pieces left.
20 percent of 900 is 180.
So he has 100 + 180 = 280 pieces done after the second day.
He has 720 pieces left.
30 percent of 720 is 216.
So he has 280 + 216 = 496 pieces done after the third day.
He has 720 - 216 = 504 pieces left.
So, the final answer is 504.

Explicit Learning Examples. Below you will find some questions followed by a wrong answer, a rationale and the correct answer.

Question: A class has 32 students and they all need to present their projects. Every period is 40 minutes long. How many periods will it take for every student to present their project if they have 5 minutes to do so?

Wrong answer: 32 students x 5 minutes = 160 minutes.
5 minutes x 40 minutes = 200 minutes.
160 minutes + 200 minutes = 360 minutes.
So it will take 360 minutes or 6 hours for every student to present their project.
So, the final answer is 360.

Rationale: The error occurs in the second line of the wrong answer. The calculation of 5 minutes x 40 minutes = 200 minutes is incorrect and irrelevant to the problem. The correct approach is to calculate the total time needed for all presentations (32 students x 5 minutes each = 160 minutes) and then divide this by the length of each period (40 minutes) to find the number of periods needed. The incorrect answer also incorrectly converts the total minutes into hours, which is not asked for in the question.

Correct answer: There are 32 students and each student needs 5 minutes to present their project.
So the total time needed is 32 x 5 = 160 minutes.
There are 40 minutes in each period.
So the number of periods needed is 160/40 = 4.
So, the final answer is 4.

Question: A bag contains 50 fewer baseball cards than football cards. There are 4 times as many football cards as hockey cards. If there are 200 hockey cards in the bag, how many cards are there altogether?

Wrong answer: 50 fewer baseball cards than football cards.
There are 4 times as many football cards as hockey cards.
If there are 200 hockey cards in the bag, there are 200 x 4 = 800 football cards and 800 - 50 = 750 baseball cards.
So there are 750 + 800 = 1550 cards altogether.
So, the final answer is 1550.

Rationale: The error occurs in the last line of the wrong answer. The wrong answer incorrectly calculates the total number of cards. The total number of cards should be the sum of the hockey cards, football cards, and baseball cards. The

correct calculation is 200 (hockey cards) + 800 (football cards) + 750 (baseball cards) = 1750 cards.

Correct answer: There are 4 times as many football cards as hockey cards, so there are 4 x 200 = 800 football cards.
There are 50 fewer baseball cards than football cards, so there are 800 - 50 = 750 baseball cards.
There are 200 hockey cards, 800 football cards and 750 baseball cards for a total of 200 + 800 + 750 = 1750 cards.
So, the final answer is 1750.

Question: A bag of caramel cookies has 20 cookies inside and a box of cookies has 4 bags in total. How many calories are inside the box if each cookie is 20 calories?

Wrong answer: 20 cookies in a bag amount to 20 x 20 = 400 calories.
There are 4 bags in a box.
So 400/4 = 100 calories in a box.
So, the final answer is 100.

Rationale: The error occurs in line 3 of the wrong answer. The incorrect part of this line is the calculation of the total calories in a box. The total calories should be calculated by multiplying the total number of cookies in the box (which is 80, not 20) by the calories per cookie. The wrong answer incorrectly divided the total calories in a bag by the number of bags in a box.

Correct answer: There are 20 cookies in each bag and 4 bags in total.
So there are 20 x 4 = 80 cookies in total.
Each cookie is 20 calories, so the total calories are 80 x 20 = 1600 calories.
So, the final answer is 1600.

Question: There are four members in one household. Each member consumes 3 slices of bread during breakfast and 2 slices of bread for snacks. A loaf of bread has 12 slices. How many days will five loaves of bread last in this family?

Wrong answer: 12/4 = 3 slices of bread per person per day.
5 loaves of bread = 5 x 12 = 60 slices of bread.
60/4 = 15 days.
So, the final answer is 15.

Rationale: The error occurs in the first line of the wrong answer. The incorrect part of this line is the calculation of the slices of bread consumed per person per day. The problem states that each member consumes 3 slices of bread for breakfast and 2 slices for snacks, so each member consumes a total of 5 slices per day, not 3. Therefore, the total slices of bread consumed per day by the family should be 5 slices per person * 4 people = 20 slices, not 12. The correct calculation should then be 60 slices / 20 slices per day = 3 days.

Correct answer: Each member consumes 3+2 = 5 slices of bread per day.
There are 4 members so they consume 5 x 4 = 20 slices of bread per day.
Each loaf has 12 slices so 5 loaves have 5 x 12 = 60 slices.
60 slices of bread will last 60/20 = 3 days.
So, the final answer is 3.

Question: Bill decides to bring donuts to work for a meeting that day. He buys a box of donuts that has 50 in total in them. He eats 2 on the ride in because he's hungry. When he gets to the office, the secretary takes another 4 out of the box when he's not looking. Lastly, right before the meeting Bill sets the box down on his desk and leaves the office to take a phone call. While Bill's away, his coworkers steal half the remaining donuts. Bill comes back and brings the box into his meeting. How many donuts are left in the box?

Wrong answer: Bill eats 2 donuts on the way to work.
The secretary takes 4 donuts.
Bill's coworkers steal 25 donuts.
Bill brings the box into the meeting with 23 donuts.
So, the final answer is 23.

Rationale: The error occurs in line 2 of the wrong answer. The incorrect part of this line is the calculation of the number of donuts stolen by Bill's coworkers. The problem states that his coworkers steal half of the remaining donuts, not half of the original number of donuts. Therefore, the number of donuts stolen by his coworkers should be half of the remaining 44 donuts, not half of the original 50 donuts.

Correct answer: Bill eats 2 donuts on the ride in, leaving 50 - 2 = 48 donuts.
The secretary eats 4 donuts, leaving 48 - 4 = 44 donuts.
Bill's coworkers eat half of the remaining donuts,

which is 44/2 = 22 donuts.
So Bill has 44 - 22 = 22 donuts left.
So, the final answer is 22.

Question: Boris has 100 pieces of Halloween candy. His daughter eats 8 pieces of candy. He separates the remaining pieces of candy into equal portions into 4 different bowls. Then he takes away 3 pieces of candy from each bowl to keep for himself. How many pieces of candy are in one bowl?

Wrong answer: Boris has 100 pieces of candy.
His daughter eats 8 pieces of candy.
He separates the remaining pieces of candy into equal portions into 4 different bowls.
Then he takes away 3 pieces of candy from each bowl to keep for himself. So he has 100-8 = 92 pieces of candy in 4 bowls.
He takes away 3 pieces of candy from each bowl. So he has 92-3 = 89 pieces of candy in 4 bowls.
So, the final answer is 89.

Rationale: The error occurs at line 5 of the wrong answer. The incorrect part of this line is the calculation of the number of candies left after Boris takes away 3 pieces from each bowl. The subtraction should be done after dividing the total number of candies by 4 (the number of bowls), not before. Also, the subtraction should be 3*4=12, not just 3, because Boris takes away 3 pieces from each of the 4 bowls.

Correct answer: Boris has 100 - 8 = 92 pieces of candy left.
He divides the 92 pieces of candy into 4 bowls. 92/4 = 23 pieces of candy in each bowl.
Then he takes away 3 pieces of candy from each bowl. 23 - 3 = 20 pieces of candy in each bowl.
So, the final answer is 20.

Question: Yvonne and Janna were writing their 1000-word pair research paper. Yvonne was able to write 400 words while Janna wrote 150 more words than Yvonne. When they edited their paper, they removed 20 words and added twice as many words as they removed. How many more words should they add to reach the research paper requirement?

Wrong answer: Yvonne wrote 400 words and Janna wrote 150 more words than Yvonne. So Janna wrote 550 words.
They removed 20 words and added twice as many words as they removed. So they added 40 words.
They need to add 40 words to reach the research paper requirement.
So, the final answer is 40.

Rationale: The error occurs in line 3 of the wrong answer. The incorrect part of this line is the calculation of the total words they have after editing. They should subtract the words they removed and add the words they added to the total words they initially wrote. The wrong answer also incorrectly calculated the number of words they need to add to reach the research paper requirement. They should subtract the total words they have after editing from the research paper requirement.

Correct answer: Yvonne wrote 400 words and Janna wrote 150 more words than Yvonne, so Janna wrote 400 + 150 = 550 words.
Together they wrote 400 + 550 = 950 words.
After editing, they removed 20 words and added 2 x 20 = 40 words. So they have 950 - 20 + 40 = 970 words.
They need 1000 words, so they need 1000 - 970 = 30 more words.
So, the final answer is 30.

Question: Luke started working on a 1000-piece jigsaw puzzle. The first day he worked on it, he put together 10 percent of the pieces. On the second day, he put together another 20 percent of the remaining pieces from the first day. On the third day, he put together 30 percent of the remaining pieces from the previous day. How many pieces does he have left to complete after the third day?

Wrong answer: 1000 pieces = 1000/100 = 10 pieces
Luke put together 10 pieces on the first day.
He put together 20 pieces on the second day.
He put together 30 pieces on the third day.
So he has 10+20+30 = 60 pieces left to complete after the third day.
So, the final answer is 60.

Rationale: The error occurs in the first line of the wrong answer. The wrong answer incorrectly calculates 10 percent of 1000 as 10 pieces, when it should be 100 pieces. The same mistake is made for the calculations on the second and third day. The correct way to solve this problem is to calculate the percentage of the remaining pieces each day, not a percentage of the original 1000 pieces.

Correct answer: 10 percent of 1000 is 100.

So he has 100 pieces done after the first day. He has 900 pieces left.
20 percent of 900 is 180.
So he has 100 + 180 = 280 pieces done after the second day.
He has 720 pieces left.
30 percent of 720 is 216.
So he has 280 + 216 = 496 pieces done after the third day.
He has 720 - 216 = 504 pieces left.
So, the final answer is 504.

Implicit Learning Examples. Below you will find some questions followed by a wrong answer and the correct answer.

Question: A class has 32 students and they all need to present their projects. Every period is 40 minutes long. How many periods will it take for every student to present their project if they have 5 minutes to do so?

Wrong answer: 32 students x 5 minutes = 160 minutes.
5 minutes x 40 minutes = 200 minutes.
160 minutes + 200 minutes = 360 minutes.
So it will take 360 minutes or 6 hours for every student to present their project.
So, the final answer is 360.

Correct answer: There are 32 students and each student needs 5 minutes to present their project.
So the total time needed is 32 x 5 = 160 minutes.
There are 40 minutes in each period.
So the number of periods needed is 160/40 = 4.
So, the final answer is 4.

Question: A bag contains 50 fewer baseball cards than football cards. There are 4 times as many football cards as hockey cards. If there are 200 hockey cards in the bag, how many cards are there altogether?

Wrong answer: 50 fewer baseball cards than football cards.
There are 4 times as many football cards as hockey cards.
If there are 200 hockey cards in the bag, there are 200 x 4 = 800 football cards and 800 - 50 = 750 baseball cards.
So there are 750 + 800 = 1550 cards altogether.
So, the final answer is 1550.

Correct answer: There are 4 times as many football cards as hockey cards, so there are 4 x 200 = 800 football cards.
There are 50 fewer baseball cards than football cards, so there are 800 - 50 = 750 baseball cards.
There are 200 hockey cards, 800 football cards and 750 baseball cards for a total of 200 + 800 + 750 = 1750 cards.
So, the final answer is 1750.

Question: A bag of caramel cookies has 20 cookies inside and a box of cookies has 4 bags in total. How many calories are inside the box if each cookie is 20 calories?

Wrong answer: 20 cookies in a bag amount to 20 x 20 = 400 calories.
There are 4 bags in a box.
So 400/4 = 100 calories in a box.
So, the final answer is 100.

Correct answer: There are 20 cookies in each bag and 4 bags in total.
So there are 20 x 4 = 80 cookies in total.
Each cookie is 20 calories, so the total calories are 80 x 20 = 1600 calories.
So, the final answer is 1600.

Question: There are four members in one household. Each member consumes 3 slices of bread during breakfast and 2 slices of bread for snacks. A loaf of bread has 12 slices. How many days will five loaves of bread last in this family?

Wrong answer: 12/4 = 3 slices of bread per person per day.
5 loaves of bread = 5 x 12 = 60 slices of bread.
60/4 = 15 days.
So, the final answer is 15.

Correct answer: Each member consumes 3+2 = 5 slices of bread per day.
There are 4 members so they consume 5 x 4 = 20 slices of bread per day.
Each loaf has 12 slices so 5 loaves have 5 x 12 = 60 slices.
60 slices of bread will last 60/20 = 3 days.
So, the final answer is 3.

Question: Bill decides to bring donuts to work for a meeting that day. He buys a box of donuts that has 50 in total in them. He eats 2 on the ride in because he's hungry. When he gets to the office, the secretary takes another 4 out of the box when he's not looking. Lastly, right before the meeting

Bill sets the box down on his desk and leaves the office to take a phone call. While Bill's away, his coworkers steal half the remaining donuts. Bill comes back and brings the box into his meeting. How many donuts are left in the box?

Wrong answer: Bill eats 2 donuts on the way to work.
The secretary takes 4 donuts.
Bill's coworkers steal 25 donuts.
Bill brings the box into the meeting with 23 donuts.
So, the final answer is 23.

Correct answer: Bill eats 2 donuts on the ride in, leaving 50 - 2 = 48 donuts.
The secretary eats 4 donuts, leaving 48 - 4 = 44 donuts.
Bill's coworkers eat half of the remaining donuts, which is 44/2 = 22 donuts.
So Bill has 44 - 22 = 22 donuts left.
So, the final answer is 22.

Question: Boris has 100 pieces of Halloween candy. His daughter eats 8 pieces of candy. He separates the remaining pieces of candy into equal portions into 4 different bowls. Then he takes away 3 pieces of candy from each bowl to keep for himself. How many pieces of candy are in one bowl?

Wrong answer: Boris has 100 pieces of candy.
His daughter eats 8 pieces of candy.
He separates the remaining pieces of candy into equal portions into 4 different bowls.
Then he takes away 3 pieces of candy from each bowl to keep for himself. So he has 100-8 = 92 pieces of candy in 4 bowls.
He takes away 3 pieces of candy from each bowl. So he has 92-3 = 89 pieces of candy in 4 bowls.
So, the final answer is 89.

Correct answer: Boris has 100 - 8 = 92 pieces of candy left.
He divides the 92 pieces of candy into 4 bowls. 92/4 = 23 pieces of candy in each bowl.
Then he takes away 3 pieces of candy from each bowl. 23 - 3 = 20 pieces of candy in each bowl.
So, the final answer is 20.

Question: Yvonne and Janna were writing their 1000-word pair research paper. Yvonne was able to write 400 words while Janna wrote 150 more words than Yvonne. When they edited their paper, they removed 20 words and added twice as many words as they removed. How many more words should they add to reach the research paper requirement?

Wrong answer: Yvonne wrote 400 words and Janna wrote 150 more words than Yvonne. So Janna wrote 550 words.
They removed 20 words and added twice as many words as they removed. So they added 40 words.
They need to add 40 words to reach the research paper requirement.
So, the final answer is 40.

Correct answer: Yvonne wrote 400 words and Janna wrote 150 more words than Yvonne, so Janna wrote 400 + 150 = 550 words.
Together they wrote 400 + 550 = 950 words.
After editing, they removed 20 words and added 2 x 20 = 40 words. So they have 950 - 20 + 40 = 970 words.
They need 1000 words, so they need 1000 - 970 = 30 more words.
So, the final answer is 30.

Question: Luke started working on a 1000-piece jigsaw puzzle. The first day he worked on it, he put together 10 percent of the pieces. On the second day, he put together another 20 percent of the remaining pieces from the first day. On the third day, he put together 30 percent of the remaining pieces from the previous day. How many pieces does he have left to complete after the third day?

Wrong answer: 1000 pieces = 1000/100 = 10 pieces
Luke put together 10 pieces on the first day.
He put together 20 pieces on the second day.
He put together 30 pieces on the third day.
So he has 10+20+30 = 60 pieces left to complete after the third day.
So, the final answer is 60.

Correct answer: 10 percent of 1000 is 100.
So he has 100 pieces done after the first day. He has 900 pieces left.
20 percent of 900 is 180.
So he has 100 + 180 = 280 pieces done after the second day.
He has 720 pieces left.
30 percent of 720 is 216.
So he has 280 + 216 = 496 pieces done after the third day.
He has 720 - 216 = 504 pieces left.
So, the final answer is 504.

I1: Answer labelling instruction

```
Now apply what you have learned and determine if the answer below is wrong or correct. Begin your evaluation by providing a short explanation. Be as objective as possible. After providing your explanation, you must rate the answer as either wrong or correct by strictly following this format: "[[rating]]", for example: "Rating: [[wrong]]" or "Rating: [[correct]]". You can only use the words wrong or correct as the final rating.
```

I2: Step labelling instruction

```
Now apply what you have learned when reading the question and the step-by-step answer below. The answer may not yet be complete. Your task is to determine if the current step will lead to a wrong or correct final answer, based on the question and the previous steps. Begin your evaluation by providing a short explanation. Be as objective as possible. After providing your explanation, you must rate the current reasoning step as either wrong or correct by strictly following this format: "[[rating]]", for example: "Rating: [[wrong]]" or "Rating: [[correct]]". You can only use the words wrong or correct as the final rating.
```

I3: Editing instruction

```
Now apply what you have learned and given the question below and a wrong answer, write the correct answer.
```

I4: Solving instruction

```
Now apply what you have learned and answer the question below.
```

Dataset	Model for train samples (incorrect answers)	Model for test samples (incorrect + correct answers)
GSM8K	LLaMA 30B	Llama 2 7B
ASDiv	Llama 2 7B	Llama 2 7B
AQuA	Llama 3 8B	Llama 3 8B

Table 4: LLMs used for answer generation.

A.2 Instructions

The instructions for the answer labelling, step labelling, editing and solving tasks are shown in I1, I2, I3 and I4, respectively.

B Models

We list below each of the seven LLMs tested, with the corresponding API provider and model identifier. With all LLMs we use hyperparameters $t = 0$, $p = 1$, and $k = 1$.

- Llama 3.1 70B, *Amazon Bedrock*, `meta.llama3-70b-instruct-v1:0`
- Titan Text G1 Express, *Amazon Bedrock*, `amazon.titan-text-express-v1`
- Command R, *Cohere*, `command-r-03-2024`
- Command R Refresh, *Cohere*, `command-r-08-2024`
- Command R+, *Cohere*, `command-r-plus-04-2024`
- Command R+ Refresh, *Cohere*, `command-r-plus-08-2024`
- WizardLM, *TogetherAI*, `microsoft/WizardLM-2-8x22B`

C Data Preparation

C.1 Answer Generation

In Table 4 we show the LLMs used to generate answers to the questions in each dataset, for both the training few-shot examples and the test samples. For the training set, we only generate incorrect answers with the listed models, while all correct answers are generated with GPT-4 or extracted from the original dataset where possible. For test samples, we use these models to generate both correct and incorrect answers. We do not run this generation step for PRM800K, as this dataset already contains annotated correct and incorrect answers.

Model	Strategy	GSM8K			ASDiv		
		$\text{label}_{\text{ans}}$	edit	solve	$\text{label}_{\text{ans}}$	edit	solve
Llama 3 70B Instruct	CoT	(83.76, 83.84)	(81.30, 81.30)	(91.76, 91.84)	(90.08, 90.12)	(82.63, 82.77)	(90.78, 90.82)
	Explicit	(82.45, 82.55)	(84.18, 84.22)	(92.78, 92.83)	(89.98, 90.02)	(81.33, 81.47)	**(91.49, 91.51)**
	Implicit	**(83.94, 84.06)**	**(84.79, 84.81)**	**(93.27, 93.33)**	**(91.36, 91.44)**	**(84.84, 84.96)**	(91.08, 91.12)
Command R	CoT	(50.43, 50.57)	(17.13, 17.27)	**(63.06, 63.14)**	(53.31, 53.49)	(49.2, 49.4)	**(77.54, 77.66)**
	Explicit	(56.93, 57.07)	(24.99, 25.21)	(56.62, 56.78)	**(64.0, 64.2)**	(47.89, 48.11)	(69.52, 69.68)
	Implicit	**(64.13, 64.27)**	**(31.00, 31.20)**	(60.4, 60.6)	(60.24, 60.36)	**(51.34, 51.46)**	(70.05, 70.15)
Command R+	CoT	(65.74, 65.86)	(47.98, 48.02)	(69.66, 69.74)	(78.85, 78.95)	(61.81, 61.99)	(81.65, 81.75)
	Explicit	(64.28, 64.32)	(59.75, 59.85)	(75.93, 76.07)	(80.38, 80.42)	(69.20, 69.40)	(83.87, 83.93)
	Implicit	**(71.87, 71.93)**	**(61.93, 62.07)**	**(79.83, 79.97)**	**(82.58, 82.62)**	**(70.59, 70.81)**	**(85.28, 85.32)**
Command R Refresh	CoT	(55.43, 55.57)	(52.05, 52.15)	(78.87, 78.93)	(54.78, 54.82)	(64.78, 64.82)	(84.45, 84.55)
	Explicit	(48.61, 48.79)	(55.87, 55.93)	(75.83, 75.97)	(37.85, 37.95)	(69.11, 69.29)	(80.85, 80.95)
	Implicit	**(62.42, 62.58)**	**(57.34, 57.46)**	**(79.14, 79.26)**	**(70.31, 70.49)**	**(72.18, 72.22)**	**(84.76, 84.84)**
Command R+ Refresh	CoT	(46.83, 46.97)	(45.84, 45.96)	(75.53, 75.67)	(77.69, 77.71)	(78.76, 78.84)	(89.37, 89.43)
	Explicit	(40.23, 40.37)	(57.49, 57.71)	(81.94, 82.06)	(64.78, 64.82)	(76.03, 76.17)	(89.88, 89.92)
	Implicit	**(47.13, 47.27)**	**(62.73, 62.87)**	**(86.23, 86.37)**	**(79.58, 79.62)**	**(81.82, 81.98)**	**(90.37, 90.43)**
Titan Text G1 Express	CoT	(53.38, 53.42)	(2.08, 2.12)	(29.64, 29.76)	(45.35, 45.45)	**(13.08, 13.12)**	(61.17, 61.23)
	Explicit	(68.68, 68.72)	(2.77, 2.83)	(33.16, 33.24)	(52.93, 53.07)	(12.88, 12.92)	(61.05, 61.15)
	Implicit	**(69.26, 69.34)**	**(2.97, 3.03)**	**(34.63, 34.77)**	**(60.17, 60.23)**	(12.89, 12.91)	**(62.77, 62.83)**
WizardLM	CoT	(85.87, 85.93)	(72.31, 72.49)	(91.25, 91.35)	(93.78, 93.82)	(80.58, 80.62)	(90.87, 90.93)
	Explicit	(85.36, 85.44)	(78.39, 78.61)	**(91.43, 91.57)**	(93.48, 93.52)	(82.38, 82.42)	**(91.39, 91.41)**
	Implicit	**(86.86, 86.94)**	**(82.23, 82.37)**	**(91.44, 91.56)**	**(94.98, 95.02)**	**(82.58, 82.62)**	(90.58, 90.62)

Model	Strategy	AQuA			PRM800K			
		$\text{label}_{\text{ans}}$	edit	solve	$\text{label}_{\text{ans}}$	$\text{label}_{\text{step}}$	edit	solve
Llama 3 70B Instruct	CoT	**(66.51, 66.69)**	(37.27, 37.33)	(55.75, 55.85)	**(31.65, 31.75)**	(49.58, 49.62)	(20.49, 20.71)	(43.83, 43.97)
	Explicit	(55.61, 55.79)	(33.91, 34.09)	(55.02, 55.18)	(18.97, 19.03)	(48.17, 48.23)	(21.67, 21.93)	(48.07, 48.13)
	Implicit	(56.58, 56.62)	**(37.49, 37.71)**	**(56.37, 56.43)**	(19.18, 19.22)	**(49.95, 50.05)**	**(26.34, 26.66)**	**(48.35, 48.45)**
Command R	CoT	(37.03, 37.17)	(7.79, 8.01)	**(21.81, 21.99)**	(21.34, 21.46)	(36.27, 36.33)	(4.60, 4.80)	(13.22, 13.38)
	Explicit	(34.07, 34.33)	(6.61, 6.79)	(17.64, 17.96)	(32.65, 32.75)	(38.98, 39.02)	(7.43, 7.57)	(12.94, 13.06)
	Implicit	**(39.69, 39.91)**	**(11.10, 11.30)**	(19.08, 19.12)	**(55.93, 56.07)**	**(43.30, 43.50)**	**(8.77, 8.83)**	**(14.74, 14.86)**
Command R+	CoT	(43.73, 43.87)	(11.8, 12.0)	(31.87, 32.13)	(16.01, 16.19)	(35.77, 35.83)	(14.41, 14.59)	**(23.79, 24.01)**
	Explicit	(46.47, 46.53)	(12.38, 12.62)	(30.94, 31.26)	**(59.63, 59.77)**	(38.78, 38.82)	(12.82, 12.98)	(18.05, 18.15)
	Implicit	**(47.53, 47.67)**	**(16.71, 16.89)**	**(35.71, 35.89)**	(59.40, 59.60)	**(39.18, 39.22)**	**(16.53, 16.67)**	(21.05, 21.15)
Command R Refresh	CoT	(47.38, 47.62)	(8.42, 8.58)	(35.20, 35.40)	(68.04, 68.16)	(39.25, 39.35)	(11.64, 11.76)	(30.54, 30.66)
	Explicit	(42.37, 42.43)	(16.46, 16.54)	(38.95, 39.25)	(67.23, 67.37)	**(55.85, 55.95)**	**(13.03, 13.17)**	(30.76, 30.84)
	Implicit	**(50.63, 50.77)**	**(16.55, 16.65)**	**(40.43, 40.57)**	**(71.03, 71.17)**	(53.63, 53.77)	(11.71, 11.89)	**(32.04, 32.16)**
Command R+ Refresh	CoT	(60.97, 61.03)	**(23.44, 23.56)**	(44.43, 44.57)	(54.40, 54.60)	**(51.85, 51.95)**	(15.99, 16.21)	**(31.83, 31.97)**
	Explicit	(53.56, 53.64)	(53.56, 53.64)	(43.06, 43.34)	(73.27, 73.33)	(51.63, 51.77)	(15.79, 16.01)	(26.77, 26.83)
	Implicit	**(63.20, 63.40)**	(21.41, 21.59)	**(47.70, 47.90)**	**(74.69, 74.77)**	(51.63, 51.77)	**(17.41, 17.57)**	(29.35, 29.45)
Titan Text G1 Express	CoT	(28.14, 28.26)	(0.76, 0.86)	(6.83, 6.97)	(26.53, 26.67)	(4.88, 5.02)	(1.94, 2.06)	(4.06, 4.14)
	Explicit	(28.89, 29.01)	(1.43, 1.57)	(7.76, 7.84)	(27.85, 27.95)	(4.83, 5.01)	(1.96, 2.04)	(4.53, 4.67)
	Implicit	**(29.78,29.82)**	**(1.94,2.06)**	**(8.52,8.66)**	**(29.03,29.17)**	**(5.57,5.63)**	**(2.01,2.07)**	**(4.62,4.78)**
WizardLM	CoT	(29.97, 30.03)	(28.88, 29.12)	(40.87, 41.13)	(18.46, 18.54)	(41.26, 41.34)	(19.83, 20.17)	(30.54, 30.66)
	Explicit	(27.76, 28.04)	(34.44, 34.56)	**(46.18,46.22)**	**(23.04,23.16)**	**(45.11,45.29)**	**(25.97,26.03)**	**(33.07,33.13)**
	Implicit	**(31.74,31.86)**	**(36.76,36.84)**	(43.06, 43.14)	(21.66, 21.74)	(42.07, 42.13)	(22.71, 22.89)	(30.63, 30.77)

Table 5: Estimated 95% confidence intervals calculated using t-distribution for all main results.

C.2 Test Set Construction

For GSM8K and ASDiv we use all test samples, with or without the correct/incorrect answers generated as per Section C.1, depending on the task. For AQuA, we make minor changes to the test set before generating the answers. PRM800K already contains CoT-style answers, though these are annotated for correctness at the intermediary reasoning step level, and not as a whole. Thus, we adapt this

Model	Strategy	GSM8K			ASDiv			AQuA			PRM800K			
		$\text{label}_{\text{ans}}$	edit	solve	$\text{label}_{\text{ans}}$	edit	solve	$\text{label}_{\text{ans}}$	edit	solve	$\text{label}_{\text{ans}}$	$\text{label}_{\text{step}}$	edit	solve
Llama 3 70B Instruct	CoT+	**86.4**$_{0.4}$	82.8$_{0.4}$	92.7$_{0.1}$	90.6$_{0.5}$	82.2$_{1.1}$	90.9$_{0.3}$	**65.7**$_{1.5}$	33.2$_{1.5}$	56.1$_{0.9}$	**30.2**$_{1.1}$	**51.8**$_{1.0}$	19.0$_{1.9}$	43.2$_{0.4}$
	Implicit	84.0$_{0.7}$	**84.8**$_{0.1}$	**93.3**$_{0.4}$	**91.4**$_{0.5}$	**84.9**$_{0.7}$	**91.1**$_{0.2}$	56.6$_{0.2}$	**37.6**$_{1.3}$	**56.4**$_{0.4}$	19.2$_{0.2}$	50.0$_{0.6}$	**26.5**$_{1.9}$	**48.4**$_{0.6}$
Command R	CoT+	50.9$_{1.1}$	20.8$_{1.2}$	**64.3**$_{1.2}$	54.7$_{0.2}$	50.8$_{1.2}$	**77.3**$_{1.1}$	35.6$_{1.6}$	7.2$_{1.4}$	**21.8**$_{0.2}$	26.2$_{0.3}$	38.1$_{1.7}$	8.2$_{0.7}$	12.1$_{0.7}$
	Implicit	**64.2**$_{0.8}$	**31.1**$_{1.2}$	60.5$_{1.2}$	**60.3**$_{0.7}$	**51.4**$_{0.7}$	70.1$_{0.6}$	**39.8**$_{1.3}$	**11.2**$_{1.2}$	19.1$_{0.3}$	**56.0**$_{0.9}$	**43.4**$_{1.2}$	**8.8**$_{0.4}$	**14.8**$_{0.7}$
Command R+	CoT+	69.7$_{0.3}$	52.1$_{0.7}$	77.2$_{0.3}$	80.1$_{0.3}$	63.2$_{0.6}$	**86.3**$_{0.9}$	46.9$_{0.8}$	12.8$_{2.0}$	32.8$_{2.1}$	17.3$_{0.5}$	34.2$_{0.1}$	12.2$_{1.3}$	**22.9**$_{0.5}$
	Implicit	**71.9**$_{0.4}$	**62.0**$_{0.8}$	**79.9**$_{0.8}$	**82.6**$_{0.2}$	**70.7**$_{1.3}$	85.3$_{0.3}$	**47.6**$_{0.9}$	**16.8**$_{1.1}$	**35.8**$_{1.1}$	**59.5**$_{1.2}$	**39.2**$_{0.2}$	**16.6**$_{0.8}$	21.1$_{0.6}$
Command R Refresh	CoT+	**64.2**$_{0.5}$	49.8$_{0.3}$	**80.5**$_{1.2}$	50.4$_{0.4}$	70.6$_{1.2}$	83.6$_{0.4}$	41.9$_{0.9}$	14.5$_{1.1}$	36.3$_{1.3}$	65.0$_{0.7}$	36.7$_{0.4}$	11.2$_{0.9}$	27.9$_{1.0}$
	Implicit	62.5$_{1.0}$	**57.4**$_{0.7}$	79.2$_{0.7}$	**70.4**$_{1.1}$	**72.2**$_{0.3}$	**84.8**$_{0.5}$	**50.7**$_{0.9}$	**16.6**$_{0.6}$	**40.5**$_{0.8}$	**71.1**$_{0.9}$	**53.7**$_{0.8}$	**11.8**$_{1.1}$	**32.1**$_{0.7}$
Command R+ Refresh	CoT+	45.0$_{0.4}$	45.1$_{0.2}$	85.2$_{0.6}$	66.5$_{1.5}$	78.5$_{0.2}$	88.9$_{0.1}$	62.0$_{1.9}$	18.5$_{0.4}$	45.4$_{1.3}$	61.5$_{1.2}$	47.1$_{1.1}$	16.7$_{1.3}$	27.5$_{0.2}$
	Implicit	**47.2**$_{0.8}$	**62.8**$_{0.9}$	**86.3**$_{0.8}$	**79.6**$_{0.2}$	**81.9**$_{1.0}$	**90.4**$_{0.4}$	**63.3**$_{1.2}$	**21.5**$_{1.1}$	**47.8**$_{1.2}$	**73.9**$_{0.8}$	**47.8**$_{0.7}$	**18.7**$_{1.1}$	**29.7**$_{0.7}$
Titan Text G1 Express	CoT+	52.5$_{1.0}$	1.2$_{0.2}$	30.4$_{1.4}$	46.6$_{1.2}$	12.5$_{0.2}$	61.1$_{0.3}$	**57.0**$_{1.7}$	1.2$_{0.1}$	**11.2**$_{0.3}$	**66.3**$_{1.2}$	47.2$_{0.5}$	2.8$_{0.3}$	5.4$_{0.5}$
	Implicit	**69.3**$_{0.5}$	**3.0**$_{0.4}$	**34.7**$_{0.9}$	**60.2**$_{0.4}$	**12.9**$_{0.1}$	**62.8**$_{0.4}$	49.6$_{0.8}$	**1.4**$_{0.1}$	11.0$_{0.2}$	45.7$_{0.9}$	**49.7**$_{1.0}$	**4.0**$_{0.5}$	**5.8**$_{0.5}$
WizardLM	CoT+	83.1$_{0.7}$	69.3$_{1.2}$	**92.0**$_{0.5}$	94.2$_{0.2}$	80.3$_{0.2}$	90.4$_{0.7}$	78.9$_{0.2}$	42.9$_{0.4}$	57.1$_{0.3}$	63.2$_{0.6}$	62.3$_{0.1}$	33.3$_{1.3}$	59.7$_{0.2}$
	Implicit	**86.9**$_{0.5}$	**82.3**$_{0.9}$	91.5$_{0.7}$	**95.0**$_{0.2}$	**82.6**$_{0.2}$	**90.6**$_{0.3}$	**79.5**$_{0.1}$	**44.1**$_{0.5}$	**62.1**$_{0.6}$	**69.0**$_{0.4}$	**62.6**$_{0.5}$	**35.7**$_{1.3}$	**61.1**$_{0.6}$

Table 6: Results of CoT prompting with extended context (CoT+) to match the sequence length of implicit prompting. Note that CoT+ includes further, diverse exemplars that implicit prompting does not contain. We report the accuracy of the final numerical result for all tasks except $\text{label}_{\text{ans}}$ and $\text{label}_{\text{step}}$, where we report the weighted F1-score of the binary label. Results are averaged over five runs to account for small variations in model-generated outputs, likely due to dynamic batching in the APIs.

dataset to our tasks. We illustrate these adaptations below.

AQuA contains, in its original version, multiple-choice questions associated with five answer options, only one of which is correct. In our experiments, we discard the options and prompt the model to generate open-ended answers. For ease of verifying the correctness of the answers at test time, we drop from the test set all samples where the golden answer is non-numerical.

PRM800K comprises questions and the respective answers, split into intermediary reasoning steps. Each step is labelled by human annotators as correct (label 1), incorrect (label -1), or neutral (label 0). Some samples are associated with a series of steps that lead to the correct solution, while others contain errors that impact the final solution. For the step labelling task, we use the reasoning steps that are annotated as either correct or incorrect. We append each of them to the previous context where available, i.e., the (correct or neutral) steps that precede it in the answer. For the answer labelling and editing tasks, we join the individual steps and label the resulting answer as correct if all steps are labelled as either correct or neutral, and incorrect if at least one of the steps contains errors.

D Statistical Significance of Results

To confirm the statistical significance of our results, we compute the 95% t-based confidence intervals of the accuracies obtained with all seven LLMs on all datasets (GSM8K, ASDiv, AQuA, PRM800K) and tasks ($\text{label}_{\text{ans}}$, $\text{label}_{\text{step}}$, edit, solve) with each prompting strategy (CoT, implicit learning, explicit learning). The confidence intervals are shown in Table 5.

E Extended Context Length Experiments

E.1 CoT+

Table 6 illustrates the fine-grained results of CoT+, and compares them to implicit learning prompting. Firstly, we note that in the large majority of cases ($\sim$80%) adding further few-shot examples to the CoT prompt results in better or similar ($<$1% difference) performance than the same setup with fewer examples (shown in Table 1). In a minority of cases, however, we observe that performance declines. This is consistent with previous findings that more examples do not strictly guarantee performance improvements (Zhao, 2023), especially in complex tasks (Opedal et al., 2025). Indeed, instances where performance declines are predominantly concentrated in the PRM800K dataset, which contains particularly challenging problems. Notably, prompt-

Dataset	Task	Strategy	
		CoT-2	Implicit
GSM8K	**label$_{ans}$**	**76.3**$_{0.2}$	75.0$_{0.1}$
	edit	51.6$_{0.6}$	**63.0**$_{0.9}$
	solve	76.9$_{1.0}$	**81.1**$_{0.5}$
ASDiv	**label$_{ans}$**	79.5$_{0.0}$	**84.3**$_{0.1}$
	edit	66.6$_{0.7}$	**72.0**$_{0.4}$
	solve	85.0$_{0.3}$	**85.5**$_{0.6}$
PRM800K	**label$_{ans}$**	69.9$_{0.0}$	**70.7**$_{0.1}$
	label$_{step}$	34.4$_{0.0}$	**35.5**$_{0.1}$
	edit	15.0$_{1.0}$	**17.1**$_{1.0}$
	solve	25.5$_{0.7}$	**28.2**$_{1.0}$

Table 7: Results of CoT with two correct answers (CoT-2) and implicit learning. We report the accuracy of the final numerical result for all tasks except label$_{ans}$ and label$_{step}$, where we report the weighted F1-score of the binary label. Results are averaged over five runs.

Strategy	ROUGE-1	ROUGE-L
CoT	0.18	0.13
Implicit	0.26	0.19
Explcit	0.19	0.14

Table 8: ROUGE-1 and ROUGE-L recall scores between generated and in-context rationales.

ing for implicit learning outperforms CoT+ in over 80% of cases. This demonstrates that the advantage of implicit learning is indeed due to the presence of incorrect answers rather than increased context length or other effects.

E.2 CoT-2

Table 7 displays the results of running CoT-2, compared to prompting for implicit learning. We observe that implicit prompting is superior to CoT-2, with the largest overall advantage in the editing and solving tasks. Surprisingly, observing incorrect answers alongside correct ones does not help the LLM label new answers for correctness in the case of GSM8K. Overall, however, the advantage of implicit prompting over CoT-2 is consistent, further evidencing that LLMs benefit from incorrect answers in their context more than from additional correct answers.

It should be noted that this experiment—which was run early on in our work—tests one model (Command R+) on three datasets (AQuA was not yet part of our test suite). LLM instructions also have slightly different wording to those used in our main setup (hence the minor discrepancies in the final values). In particular, the labelling tasks here are set up so that the LLM outputs the label directly, as opposed to a rationale justifying its choice first, followed by the label. As detailed in Section 4.3, this was later updated to guarantee robustness.

F ROUGE Analysis of Rationales

In addition to the n-gram similarity analysis of Table 3, we assess overfitting of the generated rationales to the in-context ones via ROUGE-1 and ROUGE-L, under each prompting strategy. The former measures the overlap of individual words, while the latter measures the Longest Common Subsequence (LCS). Recall scores for both metrics are shown in Table 8. Consistent with the n-gram similarity analysis, we observe that explicit prompting leads to higher ROUGE scores, and thus generated rationales that are more similar to the exemplar ones, compared with CoT and implicit prompting.

G Human Evaluation

G.1 Guidelines

We report the guidelines given to annotators for the human evaluation task. Annotators were recruited among machine learning experts.

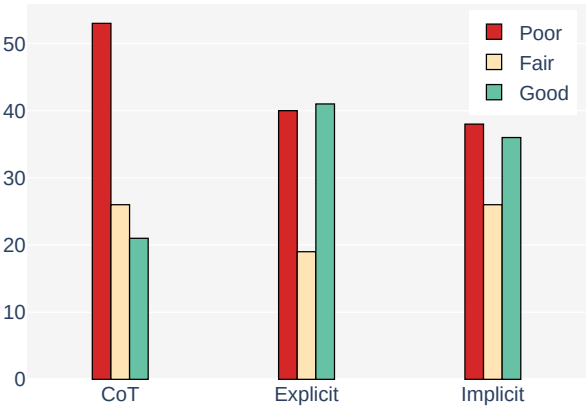


Figure 5: Fine-grained results of the human evaluation, showing the number of individual labels assigned to rationales for each prompting strategy. Explicit and implicit learning perform fairly similarly, with explicit learning obtaining a slightly higher number of labels at both extremes ('poor' and 'good') and implicit learning earning more mid-range labels ('fair'). In contrast, rationales output with the aid of CoT prompting are mainly scored as 'poor'.

Evaluating model feedback on math reasoning questions.

The attached sheet contains 100 mathematical questions and the corresponding answers given by language models. Each answer is highlighted in either green (meaning it reached the correct numerical solution) or red (meaning the numerical solution reached is wrong).
For each answer, three LLMs have generated a piece of feedback explaining why the answer is wrong or correct, your task is to score each feedback as "poor", "fair", or "good".
In your evaluation, you should only consider the correctness of the feedback. Did the model identify the strengths and/or weaknesses of the answer correctly? In your assessment, do not take feedback length into consideration. If two pieces of feedback both identify the same key points, they should be awarded the same score, even if one is much more succinct than the other. If a piece of feedback is completely missing however (meaning the model did not generate one), you should assign the label "poor". Also please ignore formatting and the presence of any special tokens or characters in your evaluation, only focus on the meaning. In each row, the three models are displayed in different order to avoid annotation bias. So, for example, "Model 1" in the first row may not be the same model as "Model 1" in the second row, and so on.

G.2 Fine-grained Results

In Figure 5, we show a breakdown of the labels assigned by human evaluators to model-generated rationales produced with each prompting strategy. While most rationales generated via CoT are assigned the minimum score, explicit and implicit learning prompting exhibit similar trends, with explicit learning obtaining slightly more labels at both ends of the spectrum ('poor' and 'good') and implicit learning receiving more mid-range labels.