Teaching Small Language Models to Reason

Lucie Charlotte Magister[*]
University of Cambridge
lcm67@cam.ac.uk

Jonathan Mallinson
Google Research
jonmall@google.com

Jakub Adamek
Google Research
enkait@google.com

Eric Malmi
Google Research
emalmi@google.com

Aliaksei Severyn
Google Research
severyn@google.com

Abstract

Chain of thought prompting successfully improves the reasoning capabilities of large language models, achieving state of the art results on a range of datasets. However, these reasoning capabilities only appear to emerge in models with at least tens of billions of parameters. In this paper, we explore the transfer of such reasoning capabilities to smaller models via knowledge distillation, also investigating model and dataset size trade-off. Specifically, we finetune a student model on the chain of thought outputs generated by a larger teacher model. Our experiments show that the proposed method improves task performance across arithmetic, commonsense and symbolic reasoning datasets. For example, the accuracy of T5 XXL on GSM8K improves from 8.11% to 21.99% and 18.42% when finetuned on PaLM 540B and GPT-3 175B generated chains of thought, respectively.

1 Introduction

Chain of thought (CoT) prompting encourages language models (LMs) to break down a reasoning task into a series of intermediate steps (Wei et al., 2022). They demonstrate that this prompting significantly increases the task accuracy of large language models (LLMs) across commonsense, symbolic and mathematical reasoning datasets. Here, LLMs are models with at least tens of billions of parameters, such as PaLM 540B (Chowdhery et al., 2022), GPT-3 175B (Brown et al., 2020), or UL2 20B (Tay et al., 2022). However, the reasoning capabilities of smaller LMs do not improve with CoT prompting, mostly producing illogical CoT. Notably, CoT prompting even reduces the accuracy of models with less than 10 billion parameters. Wei et al. (2022) attribute this to abilities, such as semantic understanding and symbolic mapping, only emerging at larger scales. This leads us to our re-

search question: *can the reasoning capabilities of LLMs be transferred to smaller LMs via finetuning?*

This work explores CoT knowledge distillation (Hinton et al., 2015) from PaLM 540B (Chowdhery et al., 2022) and GPT-3 175B (Brown et al., 2020) to different sizes of the smaller language model T5 (Raffel et al., 2020), such as T5 XXL, XL and base, which have 11 billion, 3 billion and 220 million parameters, respectively. As a result of our work, we make two recommendations: (1) perform knowledge distillation by finetuning the student model on the CoT generated by a large teacher model; and (2) generate the CoT from an LLM, as proposed by Wei et al. (2022), but crucially provide the solution to the task in the few-shot prompt. We demonstrate that the proposed method improves task performance across arithmetic, commonsense and symbolic reasoning datasets irrespective of the teacher model used. For example, we show an accuracy increase from 8.11% to 21.99% and 18.42% on the GSM8K (Cobbe et al., 2021) dataset when finetuning T5 XXL on PaLM 540B and GPT-3 175B generated CoT data, respectively.

2 Related Work

This work is inspired by the seminal work of Wei et al. (2022) on CoT prompting. They demonstrate that prefixing an input with 2-8 exemplars of CoT reasoning encourages LMs to do the same, reaching state-of-the-art performance on datasets such as GSM8K (Cobbe et al., 2021). Wang et al. (2022) show that task accuracy can be further improved by using self-consistency in CoT prompting. Self-consistency samples CoT reasoning paths from a model's decoder and returns the most consistent path by taking the majority vote. Subsequently, Chung et al. (2022) explore finetuning a FLAN-based (Wei et al., 2021) version of PaLM on manually generated CoT data.

Concurrent to our work, a small number of other works propose methods focused on CoT student–

---

[*]Research conducted during an internship at Google.

1773

teacher knowledge distillation. Ho et al. (2022) and Li et al. (2022) also explore knowledge distillation with the difference of proposing diverse sampling and rationalization prompting, respectively. In contrast to their work, our work explores more teacher models and demonstrates both the effects of dataset and model size on accuracy. We also achieve a higher accuracy on common datasets, such as GSM8K, than Ho et al. (2022). In contrast to our work, Shridhar et al. (2022) focus on training two models, one for problem decomposition and one for solving. Yet differently, the focus of Eisenstein et al. (2022) relies on producing markup-and-mask explanations for open-book question answering. Lastly, Huang et al. (2022) present one related experiment, however, we present a more in-depth exploration on more datasets. To the best of our knowledge, our work is the first to extensively explore the improvement of the reasoning ability of small LMs via knowledge distillation across multiple model architectures, and observing the effects of student model size and dataset size on accuracy.

3 Method

We propose a two-step pipeline for CoT knowledge distillation. The first step comprises annotating an existing supervised dataset with CoT reasoning generated by a teacher model. To generate high quality data, we propose using LLMs, such as PaLM 540B or GPT-3 175B, as teachers, based on the finding that CoT reasoning improves with model scale (Wei et al., 2022). Specifically, we perform few-shot prompting with 8 exemplars on these models to generate CoTs. However, we make a key modification to the prompts proposed by Wei et al. (2022). We adapt the few-shot prompts to provide the model with the target after posing the question and before providing example CoT. This is based on the observation that providing this guidance allows LLMs to correct small mistakes in the CoT. Lastly, we remove all incorrect CoT based on the target answer to prevent the student to learn from bad examples. The second step comprises finetuning a student model via teacher forcing (Williams and Zipser, 1989). The student is provided with the question as input, and the CoT and answer as the target. As the model is trained on producing a CoT during finetuning, prompting is not required. Figure 1 provides an overview of the proposed method.
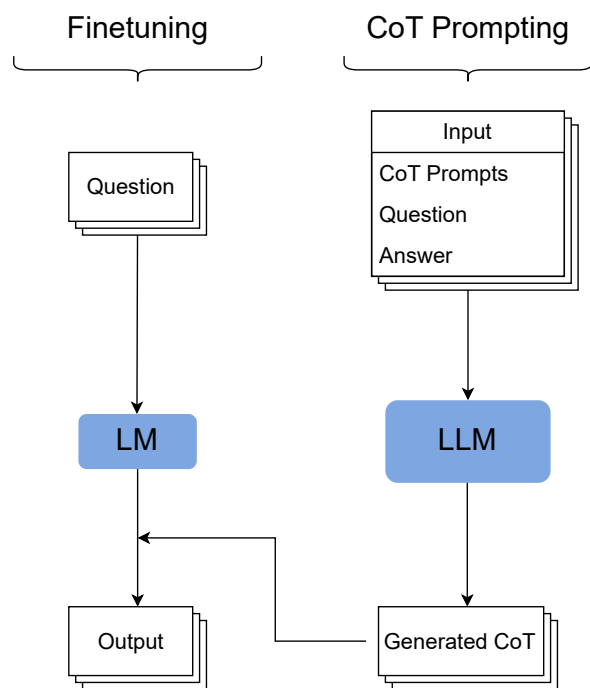

Figure 1: Overview of the proposed method.

4 Experimental Setup

We follow a similar experimental setup to Wei et al. (2022), focusing on tasks covering arithmetic, commonsense and symbolic reasoning.

4.1 Benchmarks and Metrics

4.1.1 Arithmetic Reasoning

We benchmark the proposed method on the following math word problem datasets: (1) GSM8K (Cobbe et al., 2021), (2) MAWPS (Koncel-Kedziorski et al., 2016) and (3) ASDiv (Miao et al., 2021). We use the official training and testing split for GSM8K, taking the last 10% of the training split for validation, and the 5-fold cross validation splits available for MAWPS and ASDiv. We evaluate task accuracy by checking for the target answer as the final answer in the CoT. In addition, we compute the task accuracy given an external calculator, to account for arithmetic mistakes made by the model, despite the CoT being correct. The external calculator moves through the generated output, recalculating the left hand-side of equations. It then replaces the right-hand side with the calculated output, to avoid arithmetic mistakes being carried forward. For example, if a model outputted '5 + 5 = 11. 11 * 2 = 22', then the external calculator would first calculate '5+5' and replace the '11' with a '10'. In the subsequent equation, it would

also replace the '11' with a '10' and arrive at the final result of '20'.

4.1.2 Commonsense Reasoning

We benchmark the model's ability to perform commonsense reasoning on the StrategyQA dataset (Geva et al., 2021a). As a testing split is not available, we do not shuffle the dataset to allow reproducing our split of taking the first 80% as training data, the following 10% as validation data, and the final 10% as testing data. We compute task accuracy in the same manner as previously mentioned.

4.1.3 Symbolic Reasoning

Lastly, we benchmark the model on two synthetic tasks for symbolic reasoning: (1) last letter concatenation and (2) coinflip (Wei et al., 2022). Last letter concatenation prompts the model to concatenate the last letter of each word in a string. Coinflip prompts the model to perform state tracking of the coin being flipped. We evaluate task accuracy in the same manner as before. Due to the rigid structure of the datasets, we focus on evaluating the model's generalizability to out-of-distribution (OOD) examples. We finetune the models on examples of length two and evaluate on sequences of length three and four. We initially infer the CoT using PaLM 540B, however, find that the LLM is able to perfectly replicate the desired CoT bar one example due to the rigidness of the template. We therefore decide to use the template generated CoT in our experiments.

4.2 Baselines and setup

We select PaLM 540B (Chowdhery et al., 2022) and GPT-3 175B (Brown et al., 2020) as teacher models. We select PaLM 540B based on the state-of-the-art results on the benchmarking datasets reported by Wei et al. (2022), and confirm the observed trends with GPT-3 175B. The publicly accessible teacher models are prompted as described in Section 3.

We select different sizes of T5 (Raffel et al., 2020) as student models, as T5 is publicly available in many sizes. The student models are trained on the PaLM 540B or GPT-3 175B generated CoT data as described in Section 3. We establish T5 XXL model finetuned on the original target as the baseline. We refrain from shuffling the datasets to allow for reproducibility. For the MAWPS and ASDiv dataset, we perform 5-fold cross validation. For all remaining datasets, we take 10% of the
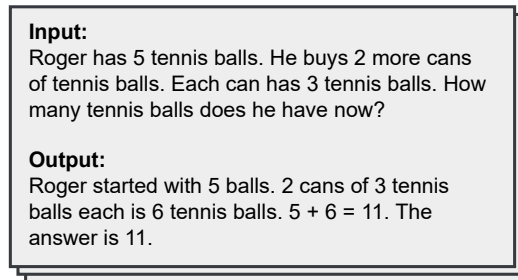

Figure 2: A training example from Wei et al. (2022) demonstrating the input and output provided to T5.

training set as a validation set to select the best model checkpoint. Figure 2 showcases an input examples for T5. We refer the reader to Wei et al. (2022) for more training examples, as well as the prompts used for generating the CoT using PaLM 540B and GPT-3 175B.

We refer the reader to Appendix A for an overview of the dataset licenses. We also refer the reader to Appendix B for an overview of the computatinal resources.

5 Results

5.1 Arithmetic reasoning

Table 1 details the task accuracy with and without an external calculator for the arithmetic reasoning benchmarks. Our results show that the proposed method improves task accuracy across all datasets. Most notably, the task accuracy of MAWPS is significantly improved. The accuracy achieved given a calculator comes close to the accuracy of 8-shot PaLM 540B, demonstrating that knowledge distillation is effective, but potentially limited by the mathematical abilities of small models.

	Baseline T5 XXL	CoT Finetuned T5 XXL		CoT 8-shot PaLM 540B	
	Acc.	Acc.	Acc. with Calc.	Acc.	Acc. with Calc.
GSM8K	8.11	**21.99**	38.21	56.90	58.60
Dataset Size	6725	5337	5337	-	-
MAWPS	54.15	**70.41**	88.22	93.00	93.66
Dataset Size	1590	1590	1590	-	-
ASDiv	39.64	**42.12**	60.73	73.9	72.6
Dataset Size	1844	1544	1544	-	-

Table 1: Task accuracy across arithmetic reasoning datasets for T5 XXL without finetuning (baseline) and finetuned on PaLM 540B generated chain-of-thought (CoT). We report the accuracy of PaLM 540B on the used datasets for reference. We do not finetune PaLM for this, but employ 8 chain of thought prompts.

5.1.1 Ablation study on generating chain-of-thought data

We perform an ablation study to confirm that providing a LLM with the target during CoT generation is beneficial. We found that for the GSM8K dataset, PaLM 540B only achieves a 59.98% accuracy if prompted without the target. In comparison, when including the target in the prompt the accuracy is 79.37%. A superficial explanation would be that when the model is conditioned on the expected answer, it produces the same CoT but copies the answer. However, an analysis of a subset of the differences between CoT produced with and without this conditioning shows that most of the benefits actually come from the model correcting CoT that had a single step missing or was wrong.

5.2 Commonsense reasoning

For the StrategyQA dataset (Table 3), we found that using CoT finetuning improves accuracy from 68.12% to 71.98%, using only 1319 of the original 1648 examples. Compared to the arithmetic reasoning datasets, the improvement is not as significant. This can be explained by the model lacking factual knowledge that the dataset requires. The task is heavily focused on the model reasoning on such knowledge, however, a smaller LM is most likely not in possession of this knowledge compared to a larger model with higher memorisation capacity.

5.3 Symbolic reasoning

Table 2 shows the results obtained for the synthetic symbolic reasoning datasets, focusing on OOD generalization. Focusing on Last Letter Concatenation, it can be stated that both traditional finetuning and the suggested method fail at generalizing to a longer sequence length. In comparison, the proposed method significantly increases accuracy for the Coinflip dataset with regard to generalizing to three coinflips. In contrast, generalisation to four coinflips is slightly weaker than the baseline, which performs very strongly. This may be related to the task length being twice that of the training task.

5.4 Replicating Results using different Teacher Models

We demonstrate the robustness of our method using a different teacher model, namely GPT-3 175B. Table 3 shows the results for GSM8K and StrategyQA when T5 XXL is finetuned on CoT data generated by GPT-3. The results show that the proposed method elicits improvements also with other

		Baseline T5 XXL	CoT Finetuned T5 XXL	CoT 8-shot PaLM 540B
Last Letter Concat.	OOD: 3	0.00	0.00	94.8
	OOD: 4	0.00	0.00	63.0
Coinflip	OOD: 3	13.10	**86.70**	98.6
	OOD: 4	**73.80**	70.50	90.2

Table 2: Task accuracy across the symbolic reasoning datasets for T5 XXL finetuned on chain-of-thought (CoT) data. For each dataset, there are 1000 training and testing examples. We report the accuracy of PaLM 540B from (Wei et al., 2022) for reference.

LLMs as teachers. We also report the accuracy of T5 XXL finetuned on golden CoT provided with the datasets. For the StrategyQA dataset, the model finetuned on the golden CoT performs best, which may be attributed to the dataset being the largest, as both PaLM and GPT-3 get some examples wrong. In contrast, the model finetuned on PaLM generated CoT performs the best for GSM8K.

	Base Task	Original Cot	CoT finetuned T5 XXL using		CoT 8-Shot	
			PaLM 540B	GPT-3 175B	PaLM 540B	GPT-3 175B
GSM8K	8.11	19.94	**21.99**	18.42	56.9	46.9
acc. with Calc.	-	26.99	38.21	33.06	58.6	49.6
Dataset Size	6725	6725	5337	5298	-	-
StrategyQA	68.12	**71.98**	67.15	63.77	77.8	65.4
Dataset Size	1648	1648	1319	1319	-	-

Table 3: Task accuracy for T5 XXL finetuned on chain-of-thought (CoT) data generated by PaLM 540B and GPT-3 175B. We also finetune on the reasoning steps provided by the datasets. We report the accuracy of PaLM 540B on the used datasets for reference. We do not finetune PaLM for this, but employ 8 chain of thought prompts.

5.5 Ablation study on model size

We investigate the performance gain achieved via finetuning student models of different sizes. Figure 3 shows the performance gain achieved when finetuning T5 of different sizes on the GSM8K dataset. Our results show that T5 base, with 44 times fewer parameters than T5 XXL, matches the performance of the baseline T5 XXL when trained on CoT data. Moreover, given an external calculator, even T5 small outperforms the baseline T5 XXL.

5.6 Ablation study on dataset size

We also investigate the trade-off between the performance gain from CoT finetuning and dataset size. Table 4 details the test accuracy achieved when finetuning T5 XXL on only 4% and 20% of the data, randomly selected. In comparison to the
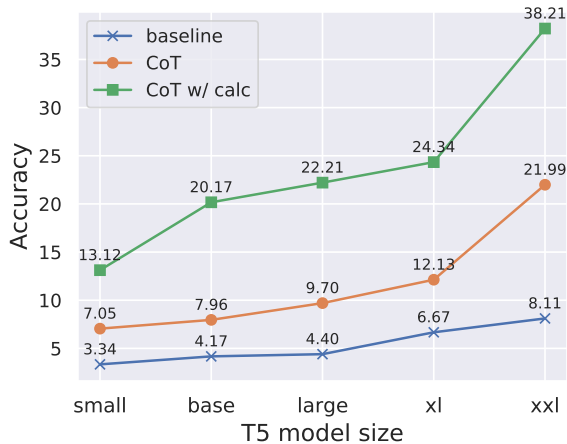
Figure 3: Effect of student model (T5) size on accuracy on GSM8K.

baseline accuracy of 8.11% (Table 3), we see that our method is 6x more data efficient, achieving accuracy of 11.22% with only 20% of the examples. However, training on just 20% of the data still creates a quality gap, and it's possible that with e.g. 200% larger dataset we could outperform the results in Table 3.

| Percentage of GSM8K | CoT finetuned T5 XXL | |
data used to train	Acc.	Acc. with Calc.
4% (213 examples)	6.29	12.28
20% (1067 examples)	11.22	20.47
100% (5337 examples)	21.99	38.21

Table 4: Task accuracy of T5 XXL finetuned on different amounts of chain-of-thought (CoT) data generated by PaLM 540B.

6 Discussion

We demonstrate that finetuning larger LMs on the CoT data generated by LLMs of over 100 billion parameters can significantly improve task accuracy. Even a small number of CoT examples appear to suffice for this. However, such improvements appear to be task dependent. For example, the effects are limited for the StrategyQA dataset, which can be attributed to the task requiring specific factual knowledge, which smaller LMs may not have memorised due to their limited capacity. Nevertheless, there is some performance improvement, which may be attributed to the model learning how to approach such tasks. Moreover, the CoT knowledge distillation pipeline presented allows to trade-off model and dataset size with accuracy. Future work could explore improving the reasoning of small

models in multi-task settings, as well as the generation of new training data using LLMs, rather than annotating existing datasets.

7 Conclusion

This work explores CoT knowledge distillation from LLMs of over 100 billion parameters to smaller LMs. We propose a knowledge distillation pipeline consisting of two keys steps: (1) generate CoT for existing datasets using LLMs and (2) finetune smaller LMs on the CoT. Our results demonstrate that finetuning on CoT improves task accuracy across a range of benchmarking datasets.

8 Limitations

The results we present must be viewed in the context of a few limitations. A limitation is that we only perform experiments in English and on one task at a time. To be more comparable to a LLM few-shot settings, other languages and a multi-task setup could be explored. Furthermore, in order to replicate the results access to none public models is required and inference must be performed on large amounts of data. Another limitation of our work is that it only explores the original CoT prompting approach, but we do not explore subsequent improvements, such a self-consistency (Wang et al., 2022).

9 Ethical Considerations

The main ethical considerations of our research arise from the text generation performed. The concerns here are that both the teacher and student model may potentially generate non-factual (Ji et al., 2022; Pagnoni et al., 2021; Kreps et al., 2022) or offensive output (Gehman et al., 2020). This is largely influenced by the input data, which is our case are standard, peer-reviewed benchmarking tasks in the NLP domain.

References

BIG-bench collaboration. 2021. Beyond the imitation game: Measuring and extrapolating the capabilities of language models. *In preparation*.

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

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

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

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Jacob Eisenstein, Daniel Andor, Bernd Bohnet, Michael Collins, and David Mimno. 2022. Honest students from untrusted teachers: Learning an interpretable question-answering pipeline from a pretrained language model. *arXiv preprint arXiv:2210.02498*.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A Smith. 2020. Realtoxicityprompts: Evaluating neural toxic degeneration in language models. *arXiv preprint arXiv:2009.11462*.

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

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

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

Namgyu Ho, Laura Schmid, and Se-Young Yun. 2022. Large language models are reasoning teachers. *arXiv preprint arXiv:2212.10071*.

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

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2022. Survey of hallucination in natural language generation. *ACM Computing Surveys*.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. Mawps: A math word problem repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1152–1157.

Sarah Kreps, R Miles McCain, and Miles Brundage. 2022. All the news that's fit to fabricate: Ai-generated text as a tool of media misinformation. *Journal of Experimental Political Science*, 9(1):104–117.

Shiyang Li, Jianshu Chen, Yelong Shen, Zhiyu Chen, Xinlu Zhang, Zekun Li, Hong Wang, Jing Qian, Baolin Peng, Yi Mao, et al. 2022. Explanations from large language models make small reasoners better. *arXiv preprint arXiv:2210.06726*.

Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2021. A diverse corpus for evaluating and developing english math word problem solvers. *arXiv preprint arXiv:2106.15772*.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with frank: A benchmark for factuality metrics. *arXiv preprint arXiv:2104.13346*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2022. Distilling multi-step reasoning capabilities of large language models into smaller models via semantic decompositions. *arXiv preprint arXiv:2212.00193*.

Yi Tay, Mostafa Dehghani, Vinh Q Tran, Xavier Garcia, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Neil Houlsby, and Donald Metzler. 2022. Unifying language learning paradigms. *arXiv preprint arXiv:2205.05131*.

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

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

A Dataset Usage and Licenses

In this section, we list the licenses for the datasets used and any ethical concerns regarding their usage. We describe the dataset splits used for all datasets in Section 4 of the paper.

A.1 Arithmetic Reasoning

The GSM8K dataset (Cobbe et al., 2021) is available under the MIT license. The MAWPS dataset (Koncel-Kedziorski et al., 2016) is available under the CC BY 4.0 and the ASDiv dataset (Miao et al., 2021) is available under the CC BY-NC 4.0 license. We follow the intended usage of the datasets.

A.2 Commonsense Reasoning

The StrategyQA dataset (Geva et al., 2021b) is available under the MIT license. Similar to Wei et al. (2022), we use the open-domain setting version available as part of the Big-bench collaboration (BIG-bench collaboration, 2021), available under the Apache License 2.0. We follow the intended usage of the datasets.

A.3 Symbolic Reasoning

We generate the symbolic reasoning datasets as described in Wei et al. (2022).

B Computational Resources

We perform inference and finetuning on different sizes of T5 on TPUs. We perform inference on PaLM 540B also on TPUs. Our results can be replicated via the public API (https://developers.generativeai.google/products/palm). To make requests to GPT-3 175B, we use the public API (https://beta.openai.com/docs/introduction).

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 8

☑ A2. Did you discuss any potential risks of your work?
Section 9

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 4

☑ B1. Did you cite the creators of artifacts you used?
Section 4

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Appendix A

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 4

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
We did not discuss this as the datasets are commonly used NLP benchmarks that do not contain personal data.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
We discuss this in Section 8, the limitations section. We discuss the coverage of domains in Section 4.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
We discuss this in Section 4.

C ☑ Did you run computational experiments?

Sections 4 and 5

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
We report the model specifics in section 4. We describe the computing infrastructure in Appendix 2, but do not estimate the computational budget.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Sections 4 and 5

☐ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Not applicable. Left blank.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Not applicable. Left blank.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Not applicable. Left blank.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.