multi-step reasoning, (2) a large language model is used, and (3) the scaling curve is relatively flat. Conversely, the benefits are smaller when one or more of these conditions are not met.

These intuitions are perhaps supported by the arithmetic reasoning results. The performance gain from chain-of-thought prompting is largest for PaLM 540B on GSM8K (challenging multi-step problems, flat scaling curve), which meets these conditions. The performance gain is small for the subsets of MAWPS that only require one or two steps (SingleOP, SingleEq, and AddSub), for which PaLM 540B already achieves performance of 90% or higher (and it is also generally true that there is less headroom for improvement when performance is already strong).

Although in this paper we focused on multi-step reasoning tasks (arithmetic, commonsense, and symbolic), chain-of-thought prompting can potentially be applied to any task for which humans use a "chain of thought" to solve (at least in principle). We leave the empirical evaluation of chain-of-thought prompting on such diverse tasks (e.g., machine translation, etc.) to future work.

A.4 Why is prompting with the equation only not enough for some arithmetic reasoning datasets?

Prompting with the equation only as an intermediate step does help on many datasets, especially when the datasets only require a few reasoning steps (SVAMP, ASDiv, MAWPS). For GSM8K, however, using the equation only did not improve performance substantially. Based on qualitative analysis, we believe that these questions are too semantically challenging for the model to directly translate them into a math equation. Consider this example from LaMDA 137B:

> **QUESTION:** Mike plays ping pong for 40 minutes. In the first 20 minutes, he scores 4 points. In the second 20 minutes, he scores 25% more points. How many total points did he score?
>
> **EQUATION ONLY (WRONG ANSWER):** (4 + 20 * 0.25) = 6. The answer is 6.
>
> **CHAIN OF THOUGHT (CORRECT):** Mike played ping pong for 40 minutes. In the first 20 minutes, he scored 4 points. In the second 20 minutes, he scored 25% more points. So he scored 25% more in the second 20 minutes. 4 x 1.25 = 5. So he scored 5 points in the second 20 minutes. So he scored 9 points in total. The answer is 9.

It is hard for the model to directly translate all of the semantics into a single equation, but chain of thought allows it to better reason about each part of the question via intermediate steps in natural language.

B All Experimental Results

This section contains tables for experimental results for varying models and model sizes, on all benchmarks, for standard prompting vs. chain-of-thought prompting.

For the arithmetic reasoning benchmarks, some chains of thought (along with the equations produced) were correct, except the model performed an arithmetic operation incorrectly. A similar observation was made in Cobbe et al. (2021). Hence, we can further add a Python program as an external calculator (using the Python `eval` function) to all the equations in the generated chain of thought. When there are multiple equations in a chain of thought, we propagate the external calculator results from one equation to the following equations via string matching. As shown in Table 1, we see that adding a calculator significantly boosts performance of chain-of-thought prompting on most tasks.

Table 1: Chain of thought prompting outperforms standard prompting for various large language models on five arithmetic reasoning benchmarks. All metrics are accuracy (%). Ext. calc.: post-hoc external calculator for arithmetic computations only. Prior best numbers are from the following. *a*: Cobbe et al. (2021). *b* & *e*: Pi et al. (2022), *c*: Lan et al. (2021), *d*: Piękos et al. (2021).

	Prompting	GSM8K	SVAMP	ASDiv	AQuA	MAWPS
Prior best	N/A (finetuning)	55[a]	57.4[b]	75.3[c]	37.9[d]	88.4[e]
UL2 20B	Standard	4.1	10.1	16.0	20.5	16.6
	Chain of thought	4.4 (+0.3)	12.5 (+2.4)	16.9 (+0.9)	23.6 (+3.1)	19.1 (+2.5)
	+ ext. calc	6.9	28.3	34.3	23.6	42.7
LaMDA 137B	Standard	6.5	29.5	40.1	25.5	43.2
	Chain of thought	14.3 (+7.8)	37.5 (+8.0)	46.6 (+6.5)	20.6 (-4.9)	57.9 (+14.7)
	+ ext. calc	17.8	42.1	53.4	20.6	69.3
GPT-3 175B	Standard	15.6	65.7	70.3	24.8	72.7
(`text-davinci-002`)	Chain of thought	46.9 (+31.3)	68.9 (+3.2)	71.3 (+1.0)	35.8 (+11.0)	87.1 (+14.4)
	+ ext. calc	49.6	70.3	71.1	35.8	87.5
Codex	Standard	19.7	69.9	74.0	29.5	78.7
(`code-davinci-002`)	Chain of thought	63.1 (+43.4)	76.4 (+6.5)	80.4 (+6.4)	45.3 (+15.8)	92.6 (+13.9)
	+ ext. calc	65.4	77.0	80.0	45.3	93.3
PaLM 540B	Standard	17.9	69.4	72.1	25.2	79.2
	Chain of thought	56.9 (+39.0)	79.0 (+9.6)	73.9 (+1.8)	35.8 (+10.6)	93.3 (+14.2)
	+ ext. calc	58.6	79.8	72.6	35.8	93.5

Table 2: Standard prompting versus chain of thought prompting on five arithmetic reasoning benchmarks. Note that chain of thought prompting is an emergent ability of model scale—it does not positively impact performance until used with a model of sufficient scale.

		GSM8K		SVAMP		ASDiv		AQuA		MAWPS	
Model		standard	CoT	standard	CoT	standard	CoT	standard	CoT	standard	CoT
UL2	20B	4.1	**4.4**	10.1	**12.5**	16.0	**16.9**	20.5	**23.6**	16.6	**19.1**
LaMDA	420M	2.6	0.4	2.5	1.6	3.2	0.8	23.5	8.3	3.2	0.9
	2B	3.6	1.9	3.3	2.4	4.1	3.8	22.9	17.7	3.9	3.1
	8B	3.2	1.6	4.3	3.4	5.9	5.0	22.8	18.6	5.3	4.8
	68B	5.7	**8.2**	13.6	**18.8**	21.8	**23.1**	22.3	20.2	21.6	**30.6**
	137B	6.5	**14.3**	29.5	**37.5**	40.1	**46.6**	25.5	20.6	43.2	**57.9**
GPT	350M	2.2	0.5	1.4	0.8	2.1	0.8	18.1	8.7	2.4	1.1
	1.3B	2.4	0.5	1.5	1.7	2.6	1.4	12.6	4.3	3.1	1.7
	6.7B	4.0	2.4	6.1	3.1	8.6	3.6	15.4	13.4	8.8	3.5
	175B	15.6	**46.9**	65.7	**68.9**	70.3	**71.3**	24.8	**35.8**	72.7	**87.1**
Codex	-	19.7	**63.1**	69.9	**76.4**	74.0	**80.4**	29.5	**45.3**	78.7	**92.6**
PaLM	8B	4.9	4.1	15.1	**16.8**	23.7	**25.2**	19.3	**21.7**	26.2	**30.5**
	62B	9.6	**29.9**	48.2	46.7	58.7	**61.9**	25.6	22.4	61.8	**80.3**
	540B	17.9	**56.9**	69.4	**79.0**	72.1	**73.9**	25.2	**35.8**	79.2	**93.3**

Table 3: Standard prompting versus chain of thought prompting on the four subsets of the MAWPS benchmark. The point of stratifying the MAWPS benchmark is to show that performance gains are minimal on easy one-step or two-step problems where large language models already achieve high performance (e.g., SingleOp, SingleEq, and AddSub).

		SingleOp		SingleEq		AddSub		MultiArith	
Model		standard	CoT	standard	CoT	standard	CoT	standard	CoT
UL2	20B	24.9	**27.2**	18.0	**20.2**	18.5	18.2	5.0	**10.7**
LaMDA	420M	2.8	1.0	2.4	0.4	1.9	0.7	5.8	1.5
	2B	4.6	4.1	2.4	3.3	2.7	3.2	5.8	1.8
	8B	8.0	7.0	4.5	4.4	3.4	5.2	5.2	2.4
	68B	36.5	**40.8**	23.9	**26.0**	17.3	**23.2**	8.7	**32.4**
	137B	73.2	**76.2**	48.8	**58.7**	43.0	**51.9**	7.6	**44.9**
GPT	350M	3.2	1.8	2.0	0.2	2.0	1.5	2.3	0.8
	1.3B	5.3	3.0	2.4	1.6	2.3	1.5	2.2	0.5
	6.7B	13.5	3.9	8.7	4.9	8.6	2.5	4.5	2.8
	175B	90.9	88.8	82.7	**86.6**	83.3	81.3	33.8	**91.7**
Codex	-	93.1	91.8	86.8	**93.1**	90.9	89.1	44.0	**96.2**
PaLM	8B	41.8	**46.6**	29.5	28.2	29.4	**31.4**	4.2	**15.8**
	62B	87.9	85.6	77.2	**83.5**	74.7	**78.2**	7.3	**73.7**
	540B	94.1	94.1	86.5	**92.3**	93.9	91.9	42.2	**94.7**