

source (examples in this dataset already included reasoning steps like a chain of thought).<sup>2</sup> Figure 6 shows that these prompts performed comparably with our manually written exemplars, also substantially outperforming standard prompting.

In addition to robustness to annotators, independently-written chains of thought, different exemplars, and various language models, we also find that chain-of-thought prompting for arithmetic reasoning is robust to different exemplar orders and varying numbers of exemplars (see Appendix A.2).

## 4 Commonsense Reasoning

Although chain of thought is particularly suitable for math word problems, the language-based nature of chain of thought actually makes it applicable to a broad class of commonsense reasoning problems, which involve reasoning about physical and human interactions under the presumption of general background knowledge. Commonsense reasoning is key for interacting with the world and is still beyond the reach of current natural language understanding systems (Talmor et al., 2021).

**Benchmarks.** We consider five datasets covering a diverse range of commonsense reasoning types. The popular **CSQA** (Talmor et al., 2019) asks commonsense questions about the world involving complex semantics that often require prior knowledge. **StrategyQA** (Geva et al., 2021) requires models to infer a multi-hop strategy to answer questions. We choose two specialized evaluation sets from the BIG-bench effort (BIG-bench collaboration, 2021): **Date** Understanding, which involves inferring a date from a given context, and **Sports** Understanding, which involves determining whether a sentence relating to sports is plausible or implausible. Finally, the **SayCan** dataset (Ahn et al., 2022) involves mapping a natural language instruction to a sequence of robot actions from a discrete set. Figure 3 shows examples with chain of thought annotations for all datasets.

**Prompts.** We follow the same experimental setup as the prior section. For CSQA and StrategyQA, we randomly selected examples from the training set and manually composed chains of thought for them to use as few-shot exemplars. The two BIG-bench tasks do not have training sets, so we selected the first ten examples as exemplars in the evaluation set as few-shot exemplars and report numbers on the rest of the evaluation set. For SayCan, we use six examples from the training set used in Ahn et al. (2022) and also manually composed chains of thought.

**Results.** Figure 7 highlights these results for PaLM (full results for LaMDA, GPT-3, and different model scales are shown in Table 4). For all tasks, scaling up model size improved the performance of standard prompting; chain-of-thought prompting led to further gains, with improvements appearing to be largest for PaLM 540B. With chain-of-thought prompting, PaLM 540B achieved strong performance relative to baselines, outperforming the prior state of the art on StrategyQA (75.6% vs 69.4%) and outperforming an unaided sports enthusiast on sports understanding (95.4% vs 84%). These results demonstrate that chain-of-thought prompting can also improve performance on tasks requiring a range of commonsense reasoning abilities (though note that gain was minimal on CSQA).

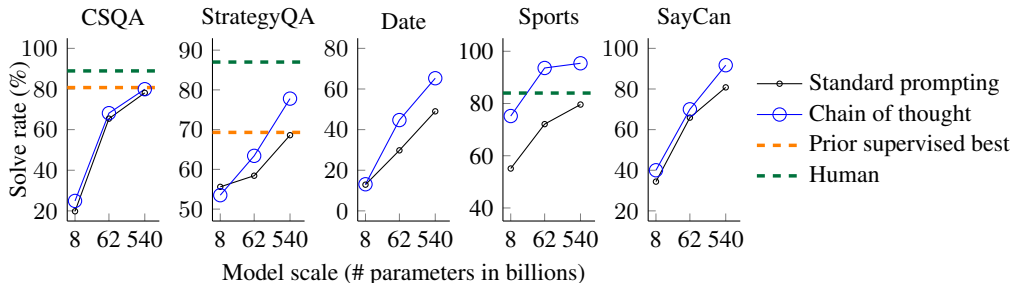


Figure 7: Chain-of-thought prompting also improves the commonsense reasoning abilities of language models. The language model shown here is PaLM. Prior best numbers are from the leaderboards of CSQA (Talmor et al., 2019) and StrategyQA (Geva et al., 2021) (single-model only, as of May 5, 2022). Additional results using various sizes of LaMDA, GPT-3, and PaLM are shown in Table 4.

<sup>2</sup>We sample examples  $\leq 60$  tokens to fit into our input context window, and also limit the examples to  $\leq 2$  steps to solve for a fair comparison with the eight exemplars that we composed.

## 5 Symbolic Reasoning

Our final experimental evaluation considers symbolic reasoning, which is simple for humans but potentially challenging for language models. We show that chain-of-thought prompting not only enables language models to perform symbolic reasoning tasks that are challenging in the standard prompting setting, but also facilitates length generalization to inference-time inputs longer than those seen in the few-shot exemplars.

**Tasks.** We use the following two toy tasks.

- **Last letter concatenation.** This task asks the model to concatenate the last letters of words in a name (e.g., “Amy Brown” → “yn”). It is a more challenging version of first letter concatenation, which language models can already perform without chain of thought.<sup>3</sup> We generate full names by randomly concatenating names from the top one-thousand first and last names from name census data (<https://namecensus.com/>).
- **Coin flip.** This task asks the model to answer whether a coin is still heads up after people either flip or don’t flip the coin (e.g., “A coin is heads up. Phoebe flips the coin. Osvaldo does not flip the coin. Is the coin still heads up?” → “no”).

As the construction of these symbolic reasoning tasks is well-defined, for each task we consider an *in-domain* test set for which examples had the same number of steps as the training/few-shot exemplars, as well as an *out-of-domain* (OOD) test set, for which evaluation examples had more steps than those in the exemplars. For last letter concatenation, the model only sees exemplars of names with two words, and then performs last letter concatenation on names with 3 and 4 words.<sup>4</sup> We do the same for the number of potential flips in the coin flip task. Our experimental setup uses the same methods and models as in the prior two sections. We again manually compose chains of thought for the few-shot exemplars for each task, which are given in Figure 3.

**Results.** The results of these in-domain and OOD evaluations are shown in Figure 8 for PaLM, with results for LaMDA shown in Appendix Table 5. With PaLM 540B, chain-of-thought prompting leads to almost 100% solve rates (note that standard prompting already solves coin flip with PaLM 540, though not for LaMDA 137B). Note that these in-domain evaluations are “toy tasks” in the sense that perfect solution structures are already provided by the chains of thought in the few-shot exemplars; all the model has to do is repeat the same steps with the new symbols in the test-time example. And yet, small models still fail—the ability to perform abstract manipulations on unseen symbols for these three tasks only arises at the scale of 100B model parameters.

As for the OOD evaluations, standard prompting fails for both tasks. With chain-of-thought prompting, language models achieve upward scaling curves (though performance is lower than in the in-domain setting). Hence, chain-of-thought prompting facilitates length generalization beyond seen chains of thought for language models of sufficient scale.

## 6 Discussion

We have explored chain-of-thought prompting as a simple mechanism for eliciting multi-step reasoning behavior in large language models. We first saw that chain-of-thought prompting improves performance by a large margin on arithmetic reasoning, yielding improvements that are much stronger than ablations and robust to different annotators, exemplars, and language models (Section 3). Next,

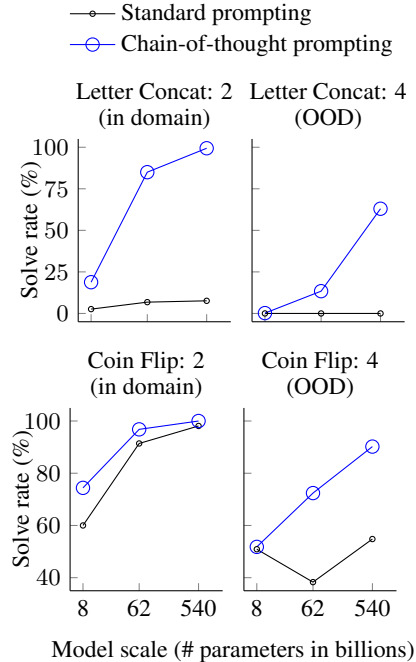


Figure 8: Using chain-of-thought prompting facilitates generalization to longer sequences in two symbolic reasoning tasks.

<sup>3</sup>We tested 10 common names using GPT-3 davinci and it got all but one correct.

<sup>4</sup>For names of length longer than 2 words, we concatenate multiple first and last names together.

experiments on commonsense reasoning underscored how the linguistic nature of chain-of-thought reasoning makes it generally applicable (Section 4). Finally, we showed that for symbolic reasoning, chain-of-thought prompting facilitates OOD generalization to longer sequence lengths (Section 5). In all experiments, chain-of-thought reasoning is elicited simply by prompting an off-the-shelf language model. No language models were finetuned in the process of writing this paper.

The emergence of chain-of-thought reasoning as a result of model scale has been a prevailing theme (Wei et al., 2022b). For many reasoning tasks where standard prompting has a flat scaling curve, chain-of-thought prompting leads to dramatically increasing scaling curves. Chain-of-thought prompting appears to expand the set of tasks that large language models can perform successfully—in other words, our work underscores that standard prompting only provides a lower bound on the capabilities of large language models. This observation likely raises more questions than it answers—for instance, how much more can we expect reasoning ability to improve with a further increase in model scale? What other prompting methods might expand the range of tasks that language models can solve?

As for limitations, we first qualify that although chain of thought emulates the thought processes of human reasoners, this does not answer whether the neural network is actually “reasoning,” which we leave as an open question. Second, although the cost of manually augmenting exemplars with chains of thought is minimal in the few-shot setting, such annotation costs could be prohibitive for finetuning (though this could potentially be surmounted with synthetic data generation, or zero-shot generalization). Third, there is no guarantee of correct reasoning paths, which can lead to both correct and incorrect answers; improving factual generations of language models is an open direction for future work (Rashkin et al., 2021; Ye and Durrett, 2022; Wiegrefe et al., 2022, *inter alia*). Finally, the emergence of chain-of-thought reasoning only at large model scales makes it costly to serve in real-world applications; further research could explore how to induce reasoning in smaller models.

## 7 Related Work

This work is inspired by many research areas, which we detail in an extended related work section (Appendix C). Here we describe two directions and associated papers that are perhaps most relevant.

The first relevant direction is using intermediate steps to solve reasoning problems. Ling et al. (2017) pioneer the idea of using natural language rationales to solve math word problems through a series of intermediate steps. Their work is a remarkable contrast to the literature using formal languages to reason (Roy et al., 2015; Chiang and Chen, 2019; Amini et al., 2019; Chen et al., 2019). Cobbe et al. (2021) extend Ling et al. (2017) by creating a larger dataset and using it to finetune a pretrained language model rather than training a model from scratch. In the domain of program synthesis, Nye et al. (2021) leverage language models to predict the final outputs of Python programs via first line-to-line predicting the intermediate computational results, and show that their step-by-step prediction method performs better than directly predicting the final outputs.

Naturally, this paper also relates closely to the large body of recent work on prompting. Since the popularization of few-shot prompting as given by Brown et al. (2020), several general approaches have improved the prompting ability of models, such as automatically learning prompts (Lester et al., 2021) or giving models instructions describing a task (Wei et al., 2022a; Sanh et al., 2022; Ouyang et al., 2022). Whereas these approaches improve or augment the input part of the prompt (e.g., instructions that are prepended to inputs), our work takes the orthogonal direction of augmenting the outputs of language models with a chain of thought.

## 8 Conclusions

We have explored chain-of-thought prompting as a simple and broadly applicable method for enhancing reasoning in language models. Through experiments on arithmetic, symbolic, and commonsense reasoning, we find that chain-of-thought reasoning is an emergent property of model scale that allows sufficiently large language models to perform reasoning tasks that otherwise have flat scaling curves. Broadening the range of reasoning tasks that language models can perform will hopefully inspire further work on language-based approaches to reasoning.