

# Chain-of-Thought & Verification

- Raw LLMs during inference do “pattern matching” and are not good at reasoning, e.g., stitching together sub-results or ideas.
- Chain-of-Thought prompting helps solve this problem.
- A verifier is a separate model from the generator. It is used predict (score) whether an answer is correct or incorrect.

2022

12,000+ citations

# Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Jason Wei   Xuezhi Wang   Dale Schuurmans   Maarten Bosma

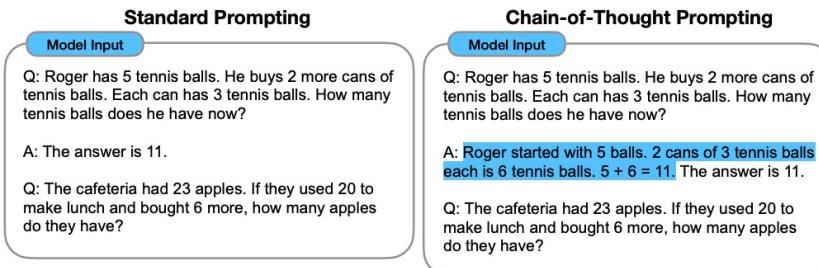
Brian Ichter   Fei Xia   Ed H. Chi   Quoc V. Le   Denny Zhou

Google Research, Brain Team  
{jasonwei, dennyzhou}@google.com

## Abstract

We explore how generating a *chain of thought*—a series of intermediate reasoning steps—significantly improves the ability of large language models to perform complex reasoning. In particular, we show how such reasoning abilities emerge naturally in sufficiently large language models via a simple method called *chain-of-thought prompting*, where a few chain of thought demonstrations are provided as exemplars in prompting.

Experiments on three large language models show that chain-of-thought prompting improves performance on a range of arithmetic, commonsense, and symbolic reasoning tasks. The empirical gains can be striking. For instance, prompting a PaLM 540B with just eight chain-of-thought exemplars achieves state-of-the-art accuracy on the GSM8K benchmark of math word problems, surpassing even finetuned GPT-3 with a verifier.



## Standard Prompting

### Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

### Model Output

A: The answer is 27. 

## Chain-of-Thought Prompting

### Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls.  $5 + 6 = 11$ . The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

### Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had  $23 - 20 = 3$ . They bought 6 more apples, so they have  $3 + 6 = 9$ . The answer is 9. 

# GSM8K

A dataset of 8.5K  
high quality  
problems at the  
grade school math  
level.

**Problem:** Beth bakes 4, 2 dozen batches of cookies in a week. If these cookies are shared amongst 16 people equally, how many cookies does each person consume?

**Solution:** Beth bakes 4 2 dozen batches of cookies for a total of  $4*2 = \textcolor{red}{<<4*2=8>>} 8$  dozen cookies

There are 12 cookies in a dozen and she makes 8 dozen cookies for a total of  $12*8 = \textcolor{red}{<<12*8=96>>} 96$  cookies

She splits the 96 cookies equally amongst 16 people so they each eat  $96/16 = \textcolor{red}{<<96/16=6>>} 6$  cookies

**Final Answer:** 6

**Problem:** Mrs. Lim milks her cows twice a day. Yesterday morning, she got 68 gallons of milk and in the evening, she got 82 gallons. This morning, she got 18 gallons fewer than she had yesterday morning. After selling some gallons of milk in the afternoon, Mrs. Lim has only 24 gallons left. How much was her revenue for the milk if each gallon costs \$3.50?

Mrs. Lim got 68 gallons - 18 gallons =  $\textcolor{red}{<<68-18=50>>} 50$  gallons this morning.

So she was able to get a total of 68 gallons + 82 gallons + 50 gallons =  $\textcolor{red}{<<68+82+50=200>>} 200$  gallons.

She was able to sell 200 gallons - 24 gallons =  $\textcolor{red}{<<200-24=176>>} 176$  gallons.

Thus, her total revenue for the milk is \$3.50/gallon x 176 gallons = \$ $\textcolor{red}{<<3.50*176=616>>} 616$ .

**Final Answer:** 616

**Problem:** Tina buys 3 12-packs of soda for a party. Including Tina, 6 people are at the party. Half of the people at the party have 3 sodas each, 2 of the people have 4, and 1 person has 5. How many sodas are left over when the party is over?

**Solution:** Tina buys 3 12-packs of soda, for  $3*12 = \textcolor{red}{<<3*12=36>>} 36$  sodas

6 people attend the party, so half of them is  $6/2 = \textcolor{red}{<<6/2=3>>} 3$  people

Each of those people drinks 3 sodas, so they drink  $3*3 = \textcolor{red}{<<3*3=9>>} 9$  sodas

Two people drink 4 sodas, which means they drink  $2*4 = \textcolor{red}{<<4*2=8>>} 8$  sodas

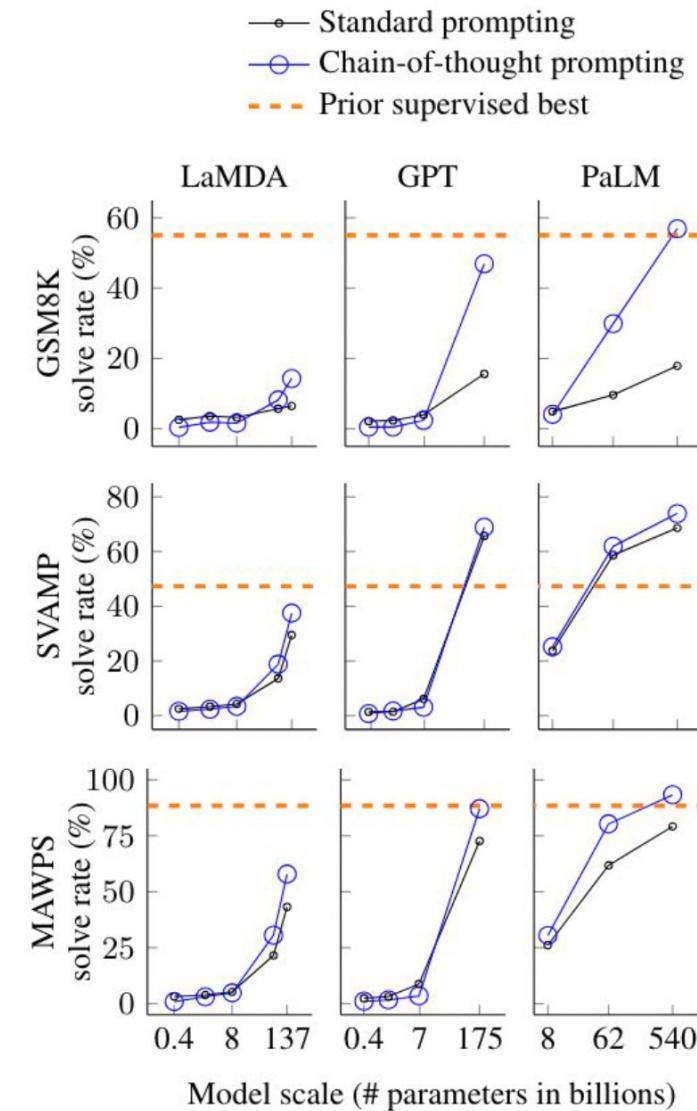
With one person drinking 5, that brings the total drank to  $5+9+8+3 = \textcolor{red}{<<5+9+8+3=25>>} 25$  sodas

As Tina started off with 36 sodas, that means there are  $36-25 = \textcolor{red}{<<36-25=11>>} 11$  sodas left

**Final Answer:** 11

# Results-

- CoT prompting is an emergent ability of model scale
- CoT prompting has larger performance gains for more-complicated problems
- CoT prompting via GPT-3 175B and PaLM 540B compares favorably to prior state of the art, which typically finetunes a task-specific model on a labeled training dataset.



# Large Language Models are Zero-Shot Reasoners

**Takeshi Kojima**  
The University of Tokyo  
[t.kojima@weblab.t.u-tokyo.ac.jp](mailto:t.kojima@weblab.t.u-tokyo.ac.jp)

**Shixiang Shane Gu**  
Google Research, Brain Team

**Machel Reid**  
Google Research\*

**Yutaka Matsuo**  
The University of Tokyo

**Yusuke Iwasawa**  
The University of Tokyo

## Abstract

Pretrained large language models (LLMs) are widely used in many sub-fields of natural language processing (NLP) and generally known as excellent *few-shot* learners with task-specific exemplars. Notably, chain of thought (CoT) prompting, a recent technique for eliciting complex multi-step reasoning through step-by-step answer examples, achieved the state-of-the-art performances in arithmetics and symbolic reasoning, difficult *system-2* tasks that do not follow the standard scaling laws for LLMs. While these successes are often attributed to LLMs' ability for few-shot learning, we show that LLMs are decent *zero-shot* reasoners by simply adding "Let's think step by step" before each answer. Experimental results demonstrate that our Zero-shot-CoT, using the same single prompt template, significantly outperforms zero-shot LLM performances on diverse benchmark reasoning tasks including arithmetics (MultiArith, GSM8K, AQUA-RAT, SVAMP), symbolic reasoning (Last Letter, Coin Flip), and other logical reasoning tasks (Date Understanding, Tracking Shuffled Objects), without any hand-crafted few-shot examples, e.g. increasing the accuracy on MultiArith from 17.7% to 78.7% and GSM8K from 10.4% to 40.7% with large-scale InstructGPT model (text-davinci-002), as well as similar magnitudes of improvements with another off-the-shelf large model, 540B parameter PaLM. The versatility of this single prompt across very diverse reasoning tasks hints at untapped and understudied fundamental *zero-shot* capabilities of LLMs, suggesting high-level, multi-task broad cognitive capabilities may be extracted by simple prompting. We hope our work not only serves as the minimal strongest zero-shot baseline for the challenging reasoning benchmarks, but also highlights the importance of carefully exploring and analyzing the enormous zero-shot knowledge hidden inside LLMs before crafting finetuning datasets or few-shot exemplars.

## 1 Introduction

Jan 2023 5000 citations

### (a) Few-shot

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

*(Output) The answer is 8. X*

### (c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

*(Output) 8 X*

### (b) Few-shot-CoT

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls.  $5 + 6 = 11$ . The answer is 11.

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A:

*(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are  $16 / 2 = 8$  golf balls. Half of the golf balls are blue. So there are  $8 / 2 = 4$  blue golf balls. The answer is 4. ✓*

### (d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: **Let's think step by step.**

*(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓*

Table 1: Accuracy comparison of Zero-shot-CoT with Zero-shot on each tasks. The values on the left side of each task are the results of using answer extraction prompts depending on answer format as described at § 3. The values on the right side are the result of additional experiment where standard answer prompt "The answer is" is used for answer extraction. See Appendix A.5 for detail setups.

	Arithmetic					
	SingleEq	AddSub	MultiArith	GSM8K	AQUA	SVAMP
zero-shot	<b>74.6/78.7</b>	<b>72.2/77.0</b>	17.7/22.7	10.4/12.5	22.4/22.4	58.8/58.7
zero-shot-cot	<b>78.0/78.7</b>	69.6/74.7	<b>78.7/79.3</b>	<b>40.7/40.5</b>	<b>33.5/31.9</b>	<b>62.1/63.7</b>
	Common Sense		Other Reasoning Tasks		Symbolic Reasoning	
	Common SenseQA	Strategy QA	Date Understand	Shuffled Objects	Last Letter (4 words)	Coin Flip (4 times)
zero-shot	<b>68.8/72.6</b>	12.7/ <b>54.3</b>	49.3/33.6	31.3/29.7	0.2/-	12.8/53.8
zero-shot-cot	64.6/64.0	<b>54.8/52.3</b>	<b>67.5/61.8</b>	<b>52.4/52.9</b>	<b>57.6/-</b>	<b>91.4/87.8</b>

# Let's Verify Step by Step

May 2023

Hunter Lightman\*    Vineet Kosaraju\*    Yura Burda\*    Harri Edwards

Bowen Baker    Teddy Lee    Jan Leike    John Schulman    Ilya Sutskever

Karl Cobbe\*

OpenAI

500+ citations

## Abstract

In recent years, large language models have greatly improved in their ability to perform complex multi-step reasoning. However, even state-of-the-art models still regularly produce logical mistakes. To train more reliable models, we can turn either to outcome supervision, which provides feedback for a final result, or process supervision, which provides feedback for each intermediate reasoning step. Given the importance of training reliable models, and given the high cost of human feedback, it is important to carefully compare the both methods. Recent work has already begun this comparison, but many questions still remain. We conduct our own investigation, finding that process supervision significantly outperforms outcome supervision for training models to solve problems from the challenging MATH dataset. Our process-supervised model solves 78% of problems from a representative subset of the MATH test set. Additionally, we show that active learning significantly improves the efficacy of process supervision. To support related research, we also release PRM800K, the complete dataset of 800,000 step-level human feedback labels used to train our best reward model.

## 1 Introduction

Large language models are capable of solving tasks that require complex multi-step reasoning by generating solutions in a step-by-step chain-of-thought format (Narang et al., 2021; Wei et al., 2022; Hall et al., 2022). However, they often make

# Let's Verify Step by Step

- Trains to two types of verifiers for math problems: outcome reward model (ORM); and process reward model (PRM).
- During inference, there is a generator LLM that provides an answer to a question.
- Reward model takes question and generator's answer as input and predicts whether answer is correct.
- Can ask LLM to generate multiple answers, and then have generator evaluate each one.
- First consider the simpler ORM...

# Generator and Outcome Reward Model (ORM)

- Both begin with GPT-4.
- Both are lightly finetuned on some math problems to provide answers in a specific format for automated correctness checking.
- **Generator** is then finetuned to generate newline-delimited, step-by-step math solutions.
  - Finetuning is done on correct solutions only, to encourage valid reasoning structure
  - Trained with next-token prediction

# Generator and Outcome Reward Model (ORM)

- **ORM** is fine-tuned to predict whether an answer is correct or not.
  - Adds a linear head and softmax to transformer, giving output  $p_\theta(\text{correct} | x_t)$ , where  $x_t$  is the question and first t generated tokens.
  - Labeled Dataset D: question and answer ( $x$ ); and binary label  $y$  in {0,1} indicating whether answer is correct.
  - Cross entropy loss function:
  - $L(\theta) = \sum_{(x,y) \in D} \sum_t p_\theta(\text{correct} | x_t)$
- At inference, output  $p_\theta(\text{correct} | x_T)$ , where T indexes last token in  $x$ .
- Thus, after training, reward model provides the probability the answer (included in  $x$ ) is correct or not.

# Process Reward Model

- Same base architecture as ORM
- Trained on the PRM800K dataset that authors created with the help of human labelers:
  - 12,000 MATH problems
  - Each problem is fed into generator to get multiple step-by-step solutions.
  - Total of 75,000 solutions and 800,000 labeled steps.
  - Humans label each of the steps: correct, incorrect, neutral (ambiguous, misleading, or technically valid but poor)
  - Labeling stops after first incorrect solution
- PRM model is trained to predict the class (correct, incorrect, neutral) for each step in dataset.
- Loss function is again cross-entropy loss, where now sum is over steps rather than tokens.

# PRM Scoring at Inference

1. Input question and answer into PRM model
  2. Get the step-level probabilities of being correct:
  3. The score of a solution is the product of step correctness probabilities.
- 
- The score for ORM is simply the probability that the entire solution is correct.

# Comparing ORM and PRM at test time

For each held-out math problem:

1. Generate N candidate solutions using the same generator (GPT-4).
2. Use the ORM or PRM to score and rank these solutions.
3. Select the top-scoring solution.
4. Check whether the final answer is correct using an automated checker.

Voting strategy: No verifier. Instead choose the answer that occurs the most among the 1860 solutions.

	ORM	PRM	Majority Voting
% Solved (Best-of-1860)	72.4	<b>78.2</b>	69.6

