

Redes Neurais e Aprendizado Profundo

Artur Jordão

Escola Politécnica – Engenharia de Computação e Sistemas Digitais
Universidade de São Paulo

Chain-of-Thought

Introduction

Chain-of-Thought

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.

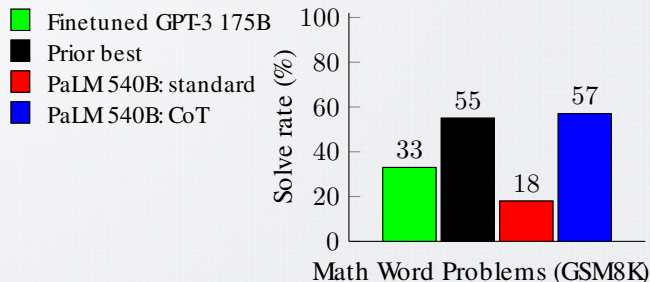
- A chain of thought is a series of intermediate natural language reasoning steps that lead to the final output
- Wei et al.¹ showed that such reasoning abilities emerge naturally in sufficiently large models via a simple method called *chain-of-thought*

¹Wei et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. NeurIPS, 2022

Introduction

Chain-of-Thought

- Chain-of-Thought (CoT) significantly improves the ability of large language models to perform complex reasoning¹



¹Wei et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. NeurIPS, 2022

Preliminaries

Chain-of-Thought

- Prompt
 - An input or instruction given to a language model to generate a response
- Few-shot
 - A learning approach where we provide only a small number of examples to teach or guide a model
- Few-shot prompting (in-context learning)
 - A few examples of a task **within the prompt** itself
 - Examples include both positive cases showing desired outputs and negative ones demonstrating incorrect responses

The Role of Prompting

Chain-of-Thought

- Given a prompt that consists of triples: input, CoT, output
- Key insights behind CoT¹
 - Techniques for arithmetic reasoning can benefit from generating natural language rationales that lead to the final answer
 - Large language models offer the exciting prospect of in-context few-shot learning via prompting (*prompting* the model with a few input–output exemplars)
- A prompting-only approach is important because it does not require a large training dataset and because a single model checkpoint can perform many tasks without loss of generality

¹Wei et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. NeurIPS, 2022

Prompt Design

Chain-of-Thought

- Standard prompting: In-context exemplars of input–output pairs before outputting a prediction for a test-time example
 - Questions and answers compose exemplars and the model gives the answer directly
- Chain-of-thought prompting: Augment each exemplar in few-shot prompting with a chain of thought for an associated answer

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. ✅

CoT Prompt for Math Word Problems

Chain-of-Thought

Q: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today?

A: There are 21 trees now and there are 15 trees in the beginning, so the workers plant $21 - 15 = 6$ trees. The answer is 6.

Q: If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?

A: There are 3 cars in the beginning, 2 more arrive, so now there should be $3 + 2 = 5$ cars. The answer is 5.

Q: Leah had 32 chocolates and her sister had 42. If they ate 35, how many pieces do they have left in total?

A: Leah had 32 chocolates and her sister had 42, in total they have $32 + 42 = 74$ chocolates. After they ate 35, now there are $74 - 35 = 39$ chocolates. The answer is 39.

CoT Prompt for Algebraic Word Problems

Chain-of-Thought

Q: John found that the average of 15 numbers is 40. If 10 is added to each number then the mean of the numbers is?

Answer Choices: (a) 50 (b) 45 (c) 65 (d) 78 (e) 64

A: If 10 is added to each number, then the mean of the numbers also increases by 10. So the new mean would be 50. The answer is (a).

Q: If $a / b = 3/4$ and $8a + 5b = 22$, then find the value of a .

Answer Choices: (a) $1/2$ (b) $3/2$ (c) $5/2$ (d) $4/2$ (e) $7/2$

A: If $a / b = 3/4$, then $b = 4a / 3$. So $8a + 5(4a / 3) = 22$. This simplifies to $8a + 20a / 3 = 22$, which means $44a / 3 = 22$. So a is equal to $3/2$. The answer is (b).

Q: A person is traveling at 20 km/hr and reached his destiny in 2.5 hr then find the distance?

Answer Choices: (a) 53 km (b) 55 km (c) 52 km (d) 60 km (e) 50 km

A: The distance that the person traveled would have been $20 \text{ km/hr} * 2.5 \text{ hrs} = 50 \text{ km}$. The answer is (e).

Effectiveness on Small and Large Models

Chain-of-Thought

- CoT is an emergent ability of model scale
 - It does not positively impact performance for small models
 - Performance gains appear in models with $\approx 100\text{B}$ parameters
 - Wei et al.¹ found that models of smaller scale produced fluent but illogical chains of thought, leading to lower performance than standard prompting

Model		GSM8K		AQuA	
		Standard	CoT	Standard	CoT
LaMDA	420M	2.6	0.4	23.5	8.3
	8B	3.2	1.6	22.8	18.6
	68B	5.7	8.2	22.3	20.2
	137B	6.5	14.3	25.5	20.6
GPT	350M	2.2	0.5	18.1	8.7
	1.3B	2.4	0.5	12.6	4.3
	6.7B	4.0	2.4	15.4	13.4
	175B	15.6	46.9	24.8	35.8

¹Wei et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. NeurIPS, 2022

Incoherent CoT

Chain-of-Thought

- Correct by chance

Question: There are 36 penguins sunbathing in the snow. One-third of them jump in and swim in the ocean. Another one-third go inside the cave to eat their dinner. How many penguins are still left sunbathing?

Model Answer (correct by chance): There are 36 penguins. One-third of them jump in and swim in the ocean. So that is 12 penguins. Another one-third go inside the cave to eat their dinner. So that is 12 penguins. The answer is 12.

Question: A local gas station is selling gas for \$3.00 a gallon. An app company is offering \$.20 cashback per gallon if you fill up at this station. If someone buys 10 gallons of gas, how much with their gas be, after the cashback rewards?

Model Answer (correct but imperfect equation): The gas is originally 3 dollars per gallon. If you buy 10 gallons, that is $3 \times 10 = 30$ dollars. If you get 20 cents off per gallon, that is $20 \times 10 = 2$ dollars. So the total is $30 - 2 = 28$ dollars. The answer is 28 dollars.

Self-Consistency and Test-Time Compute

Preliminaries

Self-Consistency & Test-Time Compute

- Greedy decoding
 - It selects the highest probability token at each step
- Basic Sampling with Temperature
 - It selects a token from the entire vocabulary based on the probabilities
 - Temperature controls the balance between creativity and determinism
- Top-k sampling
 - It selects a token from the k most likely tokens based on the probabilities
 - (i) Sort tokens by probability; (ii) keep only the top k tokens; (iii) renormalize their probabilities to sum to 1 and (iv) sample from this reduced distribution

Preliminaries

Self-Consistency & Test-Time Compute

- Reasoning (LLM context)
 - Draw inferences from existing knowledge as opposed to acquiring new knowledge¹
- Test-time compute
 - Amount of computational power a model uses when generating a response **after** training
- LLM inference via search²
 - Formulate downstream tasks as sequential decision-making problems solved via search

¹Snell et al. *Scaling LLM Test-Time Compute Optimally Can be More Effective than Scaling Parameters for Reasoning*. ICLR (Oral), 2025

²Li. *A Survey on LLM Test-Time Compute via Search: Tasks, LLM Profiling, Search Algorithms, and Relevant Frameworks*. Transactions on Machine Learning Research, 2025

Self-Consistency

Self-Consistency & Test-Time Compute

Published as a conference paper at ICLR 2023

SELF-CONSISTENCY IMPROVES CHAIN OF THOUGHT REASONING IN LANGUAGE MODELS

Xuezhi Wang^{†‡}, Jason Wei[†], Dale Schuurmans[†], Quoc Le[†], Ed H. Chi[†],
Sharan Narang[†], Aakanksha Chowdhery[†], Denny Zhou^{†§}

[†]Google Research, Brain Team

[‡]xuezhiw@google.com, [§]dennyzhou@google.com

ABSTRACT

Chain-of-thought prompting combined with pre-trained large language models has achieved encouraging results on complex reasoning tasks. In this paper, we propose a new decoding strategy, *self-consistency*, to replace the naive greedy decoding used in chain-of-thought prompting. It first samples a diverse set of reasoning paths instead of only taking the greedy one, and then selects the most consistent answer by marginalizing out the sampled reasoning paths. Self-consistency leverages the intuition that a complex reasoning problem typically admits multiple different ways of thinking leading to its unique correct answer. Our extensive empirical evaluation shows that self-consistency boosts the performance of chain-of-thought prompting with a striking margin on a range of popular arithmetic and commonsense reasoning benchmarks, including GSM8K (+17.9%), SVAMP (+11.0%), AQuA (+12.2%), StrategyQA (+6.4%) and ARC-challenge (+3.9%).

- Wang et al.¹ propose a new decoding strategy, named *self-consistency*, to replace the naive greedy decoding used in chain-of-thought prompting
- The authors hypothesize that correct reasoning processes, even if they are diverse, tend to have greater agreement in their final answer than incorrect processes

¹Wang et al. Snell et al. *Self-Consistency Improves Chain of Thought Reasoning in Language Models*. ICLR, 2023

Self-Consistency

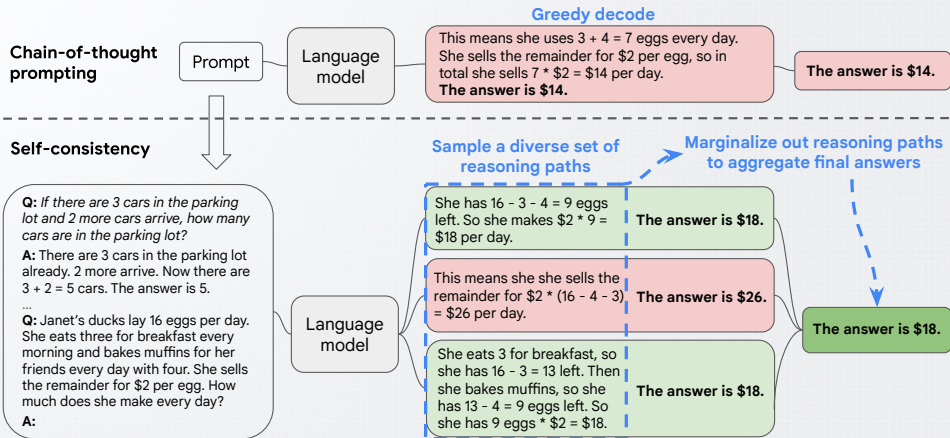
Self-Consistency & Test-Time Compute

- Self-consistency leverages the intuition that a complex reasoning problem typically admits multiple different ways of thinking leading to its unique correct answer¹
- The method involves the following steps:
 - Prompt the language model with chain-of-thought prompting to generate a diverse set of reasoning paths
 - Determine the optimal answer by sampling reasoning paths to find the most consistent answer in the final answer set
- Each reasoning path might lead to a different final answer (diversity)

¹Wang et al. Snell et al. *Self-Consistency Improves Chain of Thought Reasoning in Language Models*. ICLR, 2023

Self-Consistency

Self-Consistency & Test-Time Compute



Self-Consistency

Self-Consistency & Test-Time Compute

Question	Greedy Decode	Sampled Path 1	Sampled Path 2
[MultiArith] <i>Sam had 10 socks. If he threw away 3 old ones that didn't fit and bought 36 new ones, how many socks would he have?</i>	Sam had 10 socks. He threw away 3 old ones that didn't fit. He bought 36 new ones. That means he has $10 + 36 - 3 = 41$ socks. The answer is 41.	Sam had 10 socks. He threw away 3 old ones. So now he has 7 left. Then he bought 36 new ones. So now he has $7 + 36 = 43$ socks. The answer is 43.	He had 10 socks. He threw away 3 old ones that didn't fit. He bought 36 new ones. So in total he has $10 - 3 + 36 = 43$ socks. The answer is 43.
[SVAMP] <i>In a school there are 697 girls and the rest are boys. If there are 228 more girls than boys. How many boys are there in that school?</i>	There are 697 girls and the rest are boys. That means there are $697 + 228 = 925$ girls. The rest are boys. So the number of boys is $925 - 697 = 228$. The answer is 228.	The number of boys is the difference between 697 and 228. This is $697 - 228 = 469$. The answer is 469.	In the beginning there are 697 girls and the rest are boys. So, the number of boys must be $697 - 228 = 469$ boys. The answer is 469.
[AQuA] <i>There are 10 oranges in a basket. Find the no. of ways in which 2 oranges are chosen from the basket? (a) 45 (b) 90 (c) 120 (d) 150 (e) 180</i>	There are 10 oranges in a basket. So there are 10 ways in which 2 oranges are chosen from the basket.	There are 10 oranges in a basket. We need to choose 2 oranges from the basket. So the number of ways are 10C2. So the answer is (a).	There are 10 oranges in the basket. If you choose 2 oranges, you have $10C2 = 45$ ways of choosing 2 oranges. So the answer is (a).

Self-Consistency

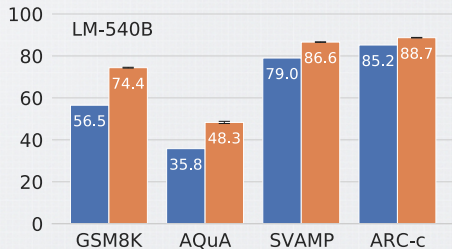
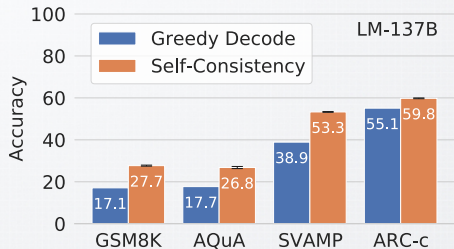
Self-Consistency & Test-Time Compute

Question	Greedy Decode	Sampled Path 1	Sampled Path 2
[CommonsenseQA] <i>The man laid on the soft moss and looked up at the trees, where was the man?</i> (a) niagra falls (b) forest (c) waterfall (d) ground (e) tree	The answer must be something that is soft and mossy. Of the above choices, only ground is soft and mossy . So the answer is (d) .	The answer must be somewhere where moss is found. Of the above choices, only forests are known for having moss . So the answer is (b) .	The answer must be a place where the man is laying down and looking up at trees. Of the above choices, only forest is a place where people lay down and look at trees . So the answer is (b) .
[ARC] <i>Which of these will most likely harm a habitat?</i> (a) planting trees. (b) water pollution. (c) rainfall. (d) sunlight.	Planting trees will not harm a habitat. So the answer is (a) .	The answer is (b) , water pollution, because it will most likely harm the habitat . The other answers are not likely to harm the habitat.	The answer is (b), water pollution. Trees are good for a habitat, rainfall is necessary for a habitat, and sunlight is necessary for a habitat . So the answer is (b) .
[ARC-challenge] <i>Which process in the carbon cycle takes the longest to complete?</i> (a) emission of waste. (b) respiration in animals. (c) photosynthesis in plants. (d) formation of fossil fuels.	The process of photosynthesis in plants takes the longest to complete. So the answer is (c) .	The formation of fossil fuels takes the longest time to complete. So the answer is (d) .	The process that takes the longest time to complete is the formation of fossil fuels, which happens over millions of years . So the answer is (d) .

Self-Consistency

Self-Consistency & Test-Time Compute

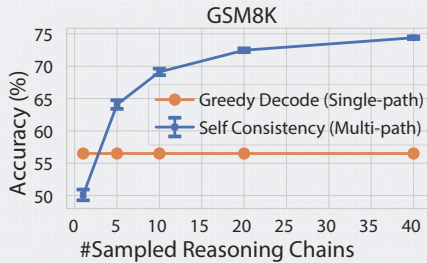
- The gain is relatively lower for smaller models
 - Certain abilities (e.g., arithmetic) only emerge when the model reaches a sufficient scale



Self-Consistency

Self-Consistency & Test-Time Compute

- Advantages: It works off-the-shelf with pre-trained language models, requires no additional human annotation, and avoids any additional training, auxiliary models or fine-tuning
- Practical issues
 - Suitable for problems where the final answer is from a fixed answer set
 - Due to the number of paths, self-consistency incurs more computation cost



Test-Time Compute

Self-Consistency & Test-Time Compute

Published as a conference paper at ICLR 2025

SCALING LLM TEST-TIME COMPUTE OPTIMALLY CAN BE MORE EFFECTIVE THAN SCALING PARAMETERS FOR REASONING

Charlie Snell^{*}, Jaehoon Lee[§], Kelvin Xu^{§†}, Aviral Kumar^{§†}

ABSTRACT

Enabling LLMs to improve their outputs by using more test-time compute is a critical step towards building self-improving agents that can operate on open-ended natural language. In this paper, we scale up inference-time computation in LLMs, with a focus on answering: *if an LLM is allowed to use a fixed but non-trivial amount of inference-time compute, how much can it improve its performance on a challenging prompt?* Answering this question has implications not only on performance, but also on the future of LLM pretraining and how to tradeoff inference-time and pre-training compute. Little research has attempted to understand the scaling behaviors of test-time inference methods, with current work largely providing negative results for a number of these strategies. In this work, we analyze two primary mechanisms to scale test-time computation: (1) searching against dense, process-based verifier reward models (PRMs); and (2) updating the model's distribution over a response adaptively, given the prompt at test time. We find that in both cases, the effectiveness of different approaches to scaling test-time compute critically varies depending on the difficulty of the prompt. This observation motivates applying a “**compute-optimal**” scaling strategy, which acts to, as effectively as possible, allocate test-time compute per prompt in an adaptive manner. Using this compute-optimal strategy, we can improve the efficiency of test-time compute scaling for math reasoning problems by more than 4 \times compared to a best-of-N baseline. Additionally, in a FLOPs-matched evaluation, we find that on problems where a smaller base model attains somewhat non-trivial success rates, test-time compute can be used to outperform a 14 \times larger model.

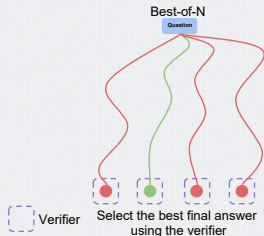
- Snell et al.¹ find that on easy and intermediate questions, additional test-time compute is often preferable to scaling pretraining
- The authors suggest that rather than focusing purely on scaling pretraining, in some settings it is more efficient to pretrain smaller models with less compute, and then apply test-time compute to improve outputs

¹Snell et al. *Scaling LLM Test-Time Compute Optimally Can be More Effective than Scaling Parameters for Reasoning*. ICLR (Oral), 2025

Preliminaries

Self-Consistency & Test-Time Compute

- Process-based reward model (PRM)¹ – **Verifier**
 - A model that produces a score of the correctness of each intermediate step in a solution
- Best-of- N search algorithm
 - Sample N answers independently from the LLM and, then, select the best answer according to Verifier score (answer judgment)



¹Lightman et al. *Let's Verify Step by Step*. ICLR, 2024

Overview of the Results

Self-Consistency & Test-Time Compute

- Longer term, the results hint at a future where fewer FLOPs are spent during pretraining and more FLOPs are spent at inference

