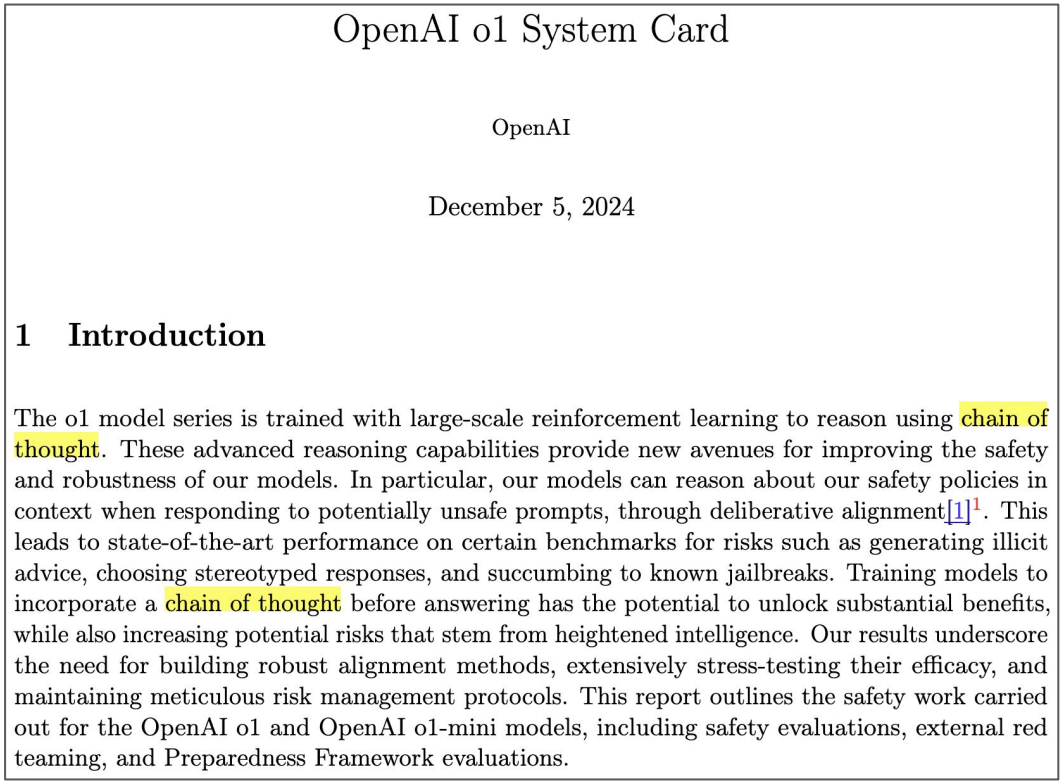
Chain-of-Thought Prompting

CS 6501 NLP @ UVA 2025 Spring

Wei-Lin Chen & Xinyu Zhu

Chain-of-Thought: A Core Concept for Improving Reasoning in LLMs

Chain-of-Thought: A Core Concept for Improving Reasoning in LLMs



OpenAI o1 System Card

OpenAI

December 5, 2024

1 Introduction

The o1 model series is trained with large-scale reinforcement learning to reason using chain of thought. These advanced reasoning capabilities provide new avenues for improving the safety and robustness of our models. In particular, our models can reason about our safety policies in context when responding to potentially unsafe prompts, through deliberative alignment[1][1]. This leads to state-of-the-art performance on certain benchmarks for risks such as generating illicit advice, choosing stereotyped responses, and succumbing to known jailbreaks. Training models to incorporate a chain of thought before answering has the potential to unlock substantial benefits, while also increasing potential risks that stem from heightened intelligence. Our results underscore the need for building robust alignment methods, extensively stress-testing their efficacy, and maintaining meticulous risk management protocols. This report outlines the safety work carried out for the OpenAI o1 and OpenAI o1-mini models, including safety evaluations, external red teaming, and Preparedness Framework evaluations.

Fig source: https://cdn.openai.com/o1-system-card-20241205.pdf

Chain-of-Thought: A Core Concept for Improving Reasoning in LLMs

OpenAI o1 System Card

OpenAI

December 5, 2024

1 Introduction

The o1 model series is trained with large-scale reinforcement learning
thought. These advanced reasoning capabilities provide new avenues
and robustness of our models. In particular, our models can reason abo
context when responding to potentially unsafe prompts, through deliber
leads to state-of-the-art performance on certain benchmarks for risks
advice, choosing stereotyped responses, and succumbing to known jailbr
incorporate a chain of thought before answering has the potential to un
while also increasing potential risks that stem from heightened intelligenc
the need for building robust alignment methods, extensively stress-te
maintaining meticulous risk management protocols. This report outline
out for the OpenAI o1 and OpenAI o1-mini models, including safety evaluations, external red teaming, and Preparedness Framework evaluations.

deepseek

DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

DeepSeek-AI

1.1. Contributions

Post-Training: Large-Scale Reinforcement Learning on the Base Model

- We directly apply RL to the base model without relying on supervised fine-tuning (SFT) as a preliminary step. This approach allows the model to explore chain-of-thought (CoT) for solving complex problems, resulting in the development of DeepSeek-R1-Zero. DeepSeek-R1-Zero demonstrates capabilities such as self-verification, reflection, and generating long CoTs, marking a significant milestone for the research community. Notably, it is the first open research to validate that reasoning capabilities of LLMs can be incentivized purely through RL, without the need for SFT. This breakthrough paves the way for future advancements in this area.

Fig source:
https://cdn.openai.com/o1-system-card-20241205.pdf
https://arxiv.org/abs/2501.12948

Outline

- Chain-of-Thought Prompting Elicits Reasoning in Large Language Models
- Least-to-Most Prompting Enables Complex Reasoning in Large Language Models
- Tree of Thoughts: Deliberate Problem Solving with Large Language Models

Outline

- Chain-of-Thought Prompting Elicits Reasoning in Large Language Models
- Least-to-Most Prompting Enables Complex Reasoning in Large Language Models
- Tree of Thoughts: Deliberate Problem Solving with Large Language Models

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

Fig source: https://arxiv.org/abs/2201.11903

Scaling Up Model Size Alone Might not be Enough (Sometimes)

- Chain-of-Thought (CoT) prompting is emergent in that it does not have a positive effect until a certain model scale.

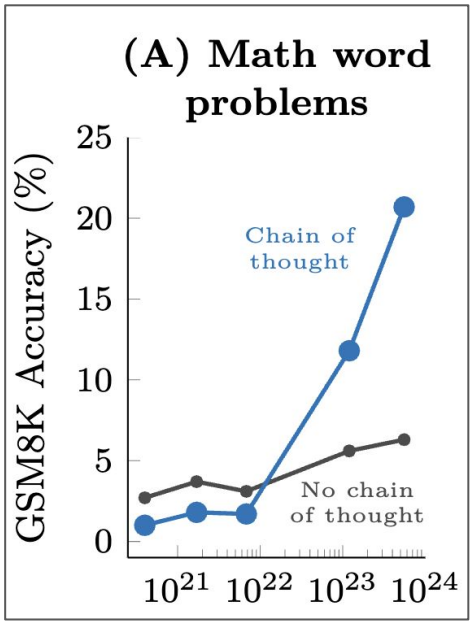


Fig source: https://arxiv.org/abs/2206.07682

Technical Motivation (1/2)

- Generating **natural language rationales** (tokens of intermediate reasoning steps) that lead to the final answer improve LMs' performance on reasoning tasks.

Problem 1:
Question: Two trains running in opposite directions cross a man standing on the platform in 27 seconds and 17 seconds respectively and they cross each other in 23 seconds. The ratio of their speeds is:
Options: A) 3/7 B) 3/2 C) 3/88 D) 3/8 E) 2/2
Rationale: Let the speeds of the two trains be x m/sec and y m/sec respectively. Then, length of the first train = 27x meters, and length of the second train = 17 y meters. $(27x + 17y) / (x + y) = 23 \rightarrow 27x + 17y = 23x + 23y \rightarrow 4x = 6y \rightarrow x/y = 3/2.$
Correct Option: B

Fig source: https://arxiv.org/abs/1705.04146

Technical Motivation (1/2)

- **Limitation**: Costly to create a large set of high quality rationales, which is much more complicated than simple input–output pairs used in typical machine learning systems.

Input

Problem 1:
Question: Two trains running in opposite directions cross a man standing on the platform in 27 seconds and 17 seconds respectively and they cross each other in 23 seconds. The ratio of their speeds is:
Options: A) 3/7 B) 3/2 C) 3/88 D) 3/8 E) 2/2

Output

Correct Option: B

Fig source: https://arxiv.org/abs/1705.04146

Technical Motivation (1/2)

- **Limitation**: Costly to create a large set of high quality rationales, which is much more complicated than simple input–output pairs used in typical machine learning systems.

Input

Problem 1:
Question: Two trains running in opposite directions cross a man standing on the platform in 27 seconds and 17 seconds respectively and they cross each other in 23 seconds. The ratio of their speeds is:
Options: A) 3/7 B) 3/2 C) 3/88 D) 3/8 E) 2/2

Output

Rationale: Let the speeds of the two trains be x m/sec and y m/sec respectively. Then, length of the first train = 27x meters, and length of the second train = 17 y meters. $(27x + 17y) / (x + y) = 23 \rightarrow 27x + 17y = 23x + 23y \rightarrow 4x = 6y \rightarrow x/y = 3/2$.
Correct Option: B

Fig source: https://arxiv.org/abs/1705.04146

Technical Motivation (1/2)

- **Limitation**: Costly to create a large set of high quality rationales, which is much more complicated than simple input–output pairs used in typical machine learning systems.

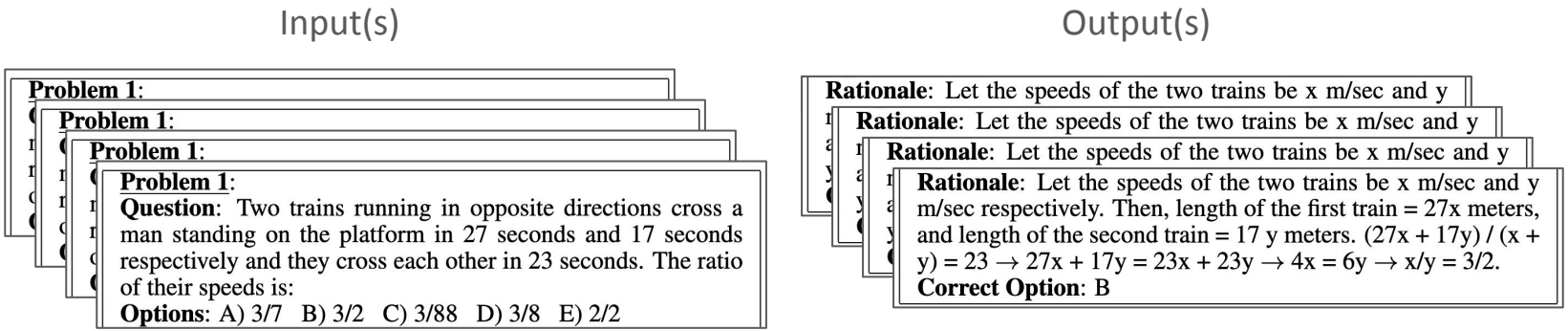


Fig source: https://arxiv.org/abs/1705.04146

Technical Motivation (2/2)

- Large language models (LLMs) offer the exciting prospect of **in-context few-shot "learning"** via prompting.

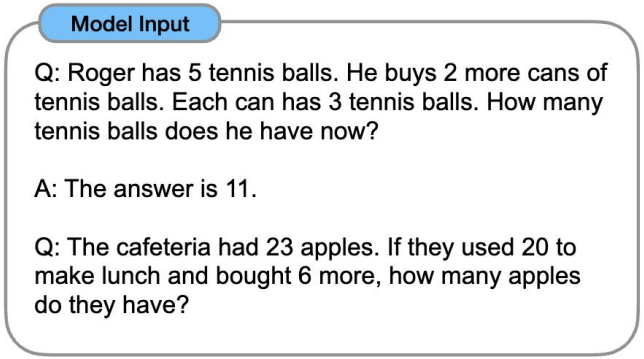


Fig source: https://arxiv.org/abs/2201.11903

Technical Motivation (2/2)

- **Limitation**: For the traditional few-shot prompting method used in Brown et al. (2020), it works poorly on tasks that require reasoning abilities.

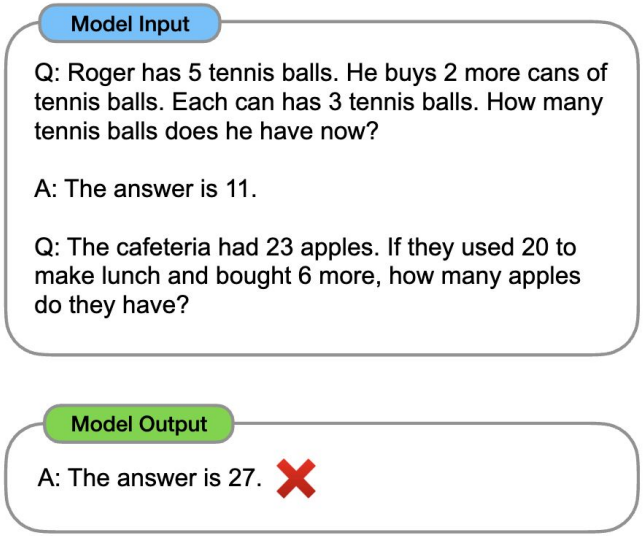


Fig source: https://arxiv.org/abs/2201.11903

Chain-of-Thought (CoT) Prompting

- Combine the strengths of these two ideas in a way that avoids their limitations.

Chain-of-Thought (CoT) Prompting

- Combine the strengths of these two ideas in a way that avoids their limitations.
- Method
  - Curate demonstrations consisting of ***input-CoT-output triples*** to enable LMs to perform few-shot prompting for reasoning tasks.

Chain-of-Thought (CoT) Prompting

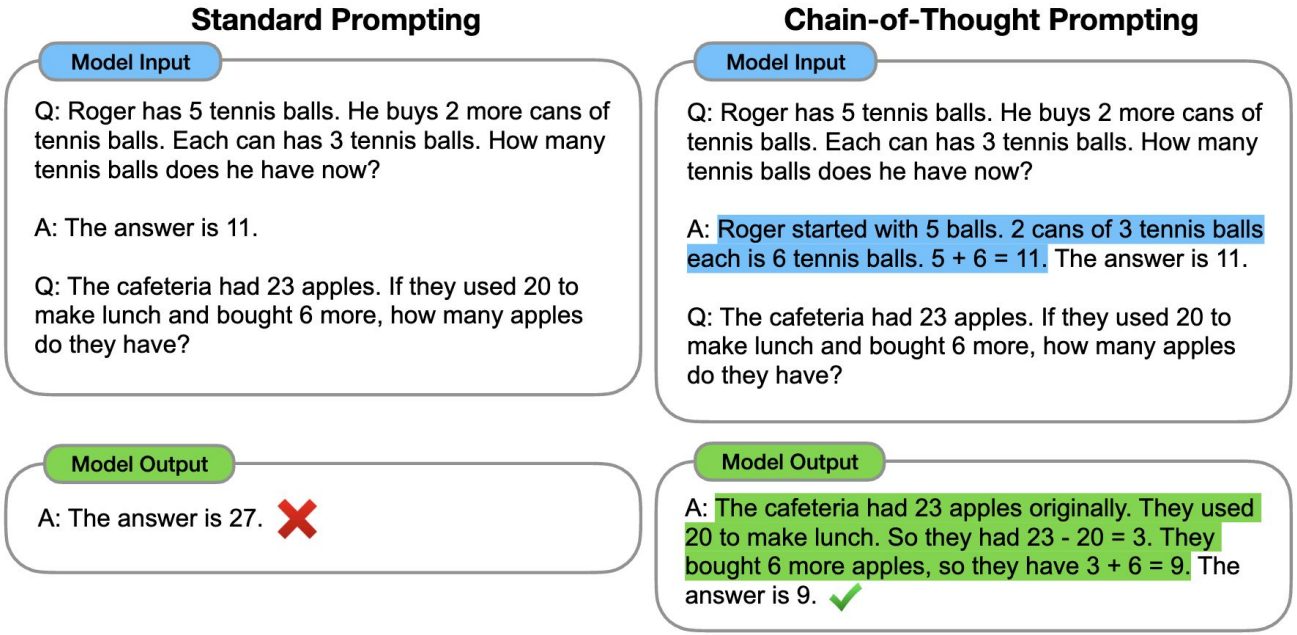


Fig source: https://arxiv.org/abs/2201.11903

Chain-of-Thought (CoT) Prompting

- A **chain of thought** is
  - a coherent series of intermediate reasoning steps that lead to the final answer for a problem.
- The approach of few-shot prompting with input-CoT-output demonstration: CoT prompting.

Why CoT? (1/4)

- CoT allows models to decompose multi-step problems into intermediate steps
  - **Additional computation can be allocated** to problems that require more reasoning steps.

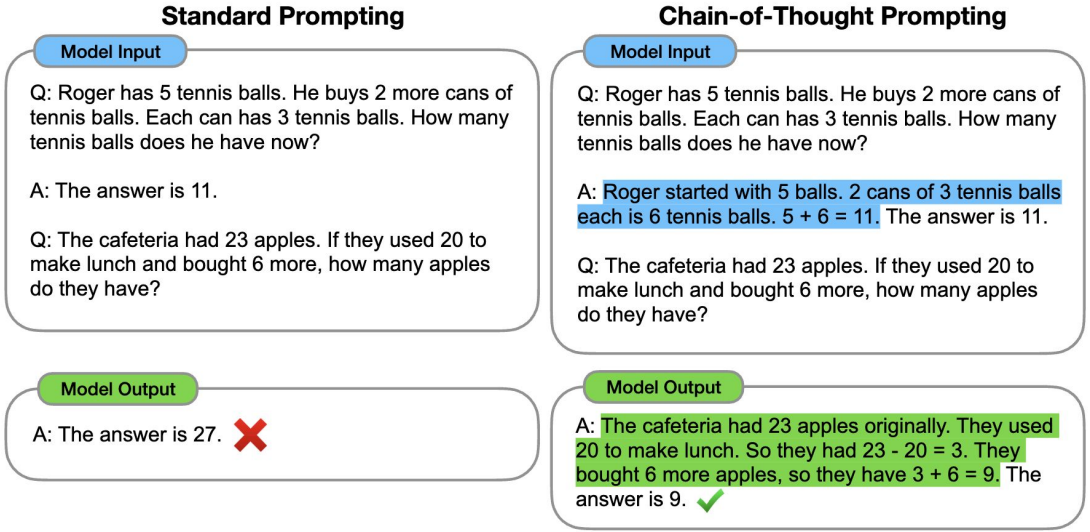


Fig source: https://arxiv.org/abs/2201.11903

Why CoT? (1/4)

- CoT allows models to decompose multi-step problems into intermediate steps
  - **Additional computation can be allocated** to problems that require more reasoning steps.

o1's test-time scaling – models increasing the length of the CoT and "think" longer for potentially harder questions.

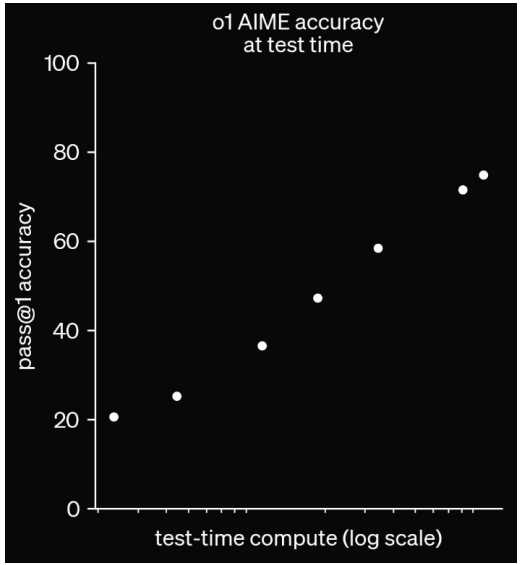


Fig source: https://openai.com/index/learning-to-reason-with-llms/

Why CoT? (2/4)

- CoT provides an **interpretable window** into the behavior of the model.
  - Suggest how it might have arrived at a particular answer.
  - Provide opportunities to debug where the reasoning path went wrong.

Why CoT? (3/4)

- Generalization potential
  - In principle, CoT is applicable to any kinds of reasoning tasks that humans can solve via language.

Why CoT? (4/4)

- No need for training
  - Readily elicited in sufficiently large **off-the-shelf** language models simply by inserting chain of thought sequences into few-shot demonstrations.

Experiment & Main Results

- Math reasoning
- Commonsense reasoning
- Symbolic reasoning

Math Reasoning

- Math word problem datasets (*GSM8K*, *SVAMP*, *ASDiv*, *AQuA*, *MAWPS*)

Math Word Problems (free response)

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.

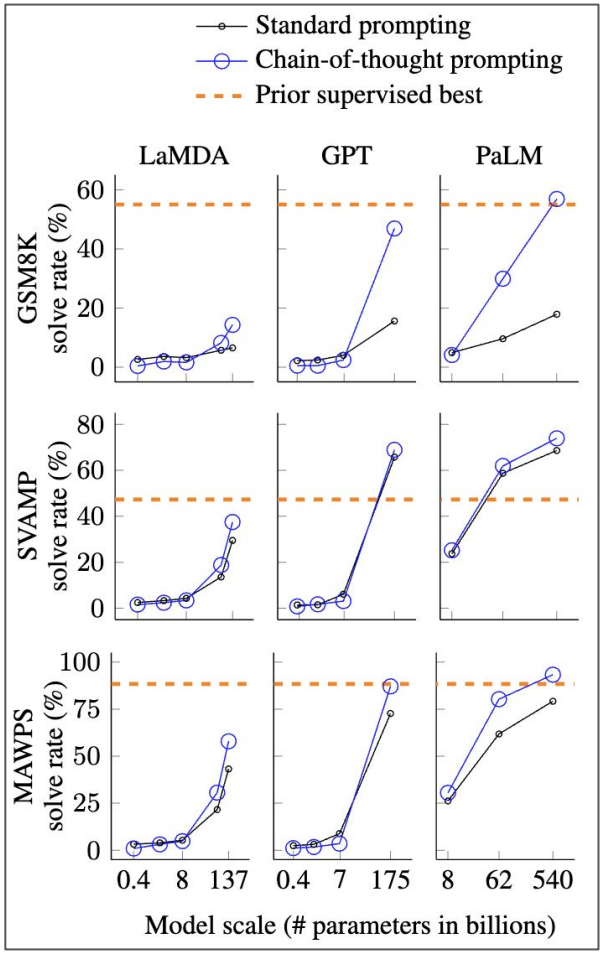


Fig source: https://arxiv.org/abs/2201.11903

Commonsense Reasoning

- Commonsense QA datasets (CSQA, StrategyQA, Date, Sports, SayCan)

Date Understanding

Q: The concert was scheduled to be on 06/01/1943, but was delayed by one day to today. What is the date 10 days ago in MM/DD/YYYY?

A: One day after 06/01/1943 is 06/02/1943, so today is 06/02/1943. 10 days before today is 05/23/1943. So the answer is 05/23/1943.

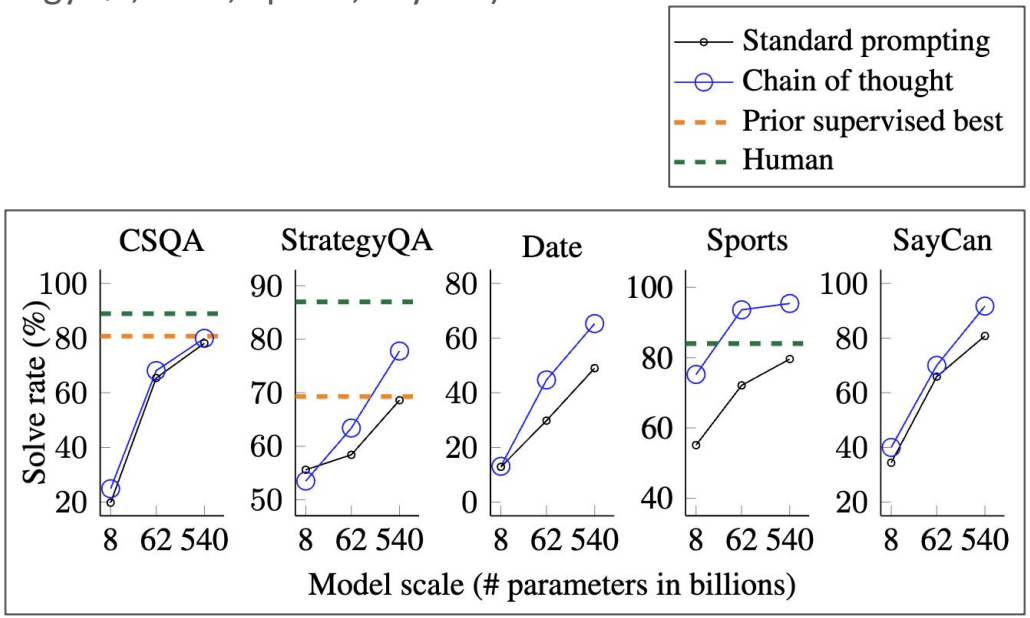


Fig source: https://arxiv.org/abs/2201.11903

Symbolic Reasoning

- Two toy datasets (*Last letter concatenation*, *Coin flip*)

Last Letter Concatenation

Q: Take the last letters of the words in “Lady Gaga” and concatenate them.

A: The last letter of “Lady” is “y”. The last letter of “Gaga” is “a”. Concatenating them is “ya”. So the answer is ya.

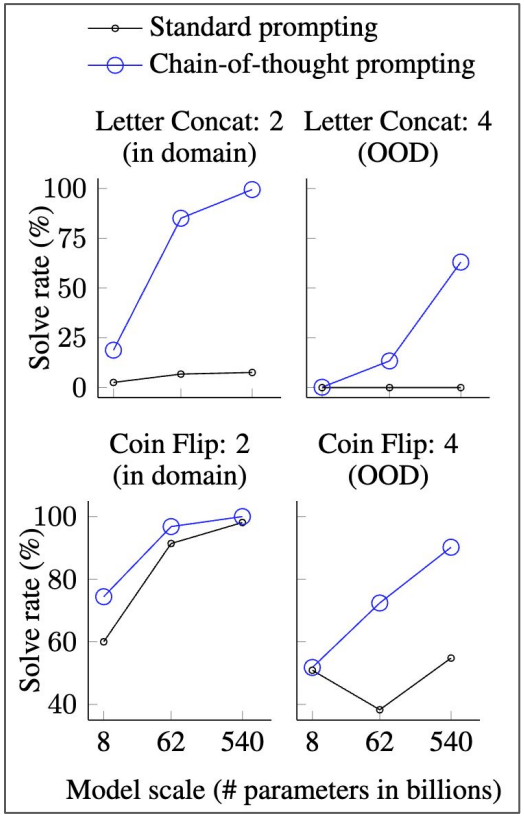


Fig source: https://arxiv.org/abs/2201.11903

Analysis: What Matters in CoT? (1/3)

- Equation-only prompting
  - The model is prompted to output only mathematical equations before giving the answer.

Analysis: What Matters in CoT? (1/3)

- Equation-only prompting
  - The model is prompted to output only mathematical equations before giving the answer.
- **Findings**
  - Equation-only prompting does not help much.

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.

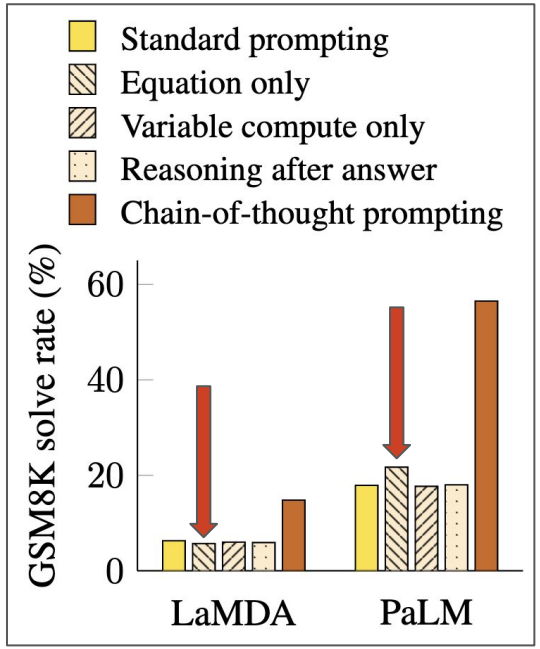


Fig source: https://arxiv.org/abs/2201.11903

Analysis: What Matters in CoT? (1/3)

- **Implications**
  - Questions are too semantically challenging for the model to directly translate them into a math equation.
  - CoT allows LLMs to better reason about each part of the question via intermediate steps in natural language.

Analysis: What Matters in CoT? (2/3)

- Variable-compute-only prompting
  - The model is prompted to **output a only sequence of dots (. . .)** equal to the number of characters in the reasoning chain.

Analysis: What Matters in CoT? (2/3)

- Variable-compute-only prompting
  - The model is prompted to **output a only sequence of dots (. . .)** equal to the number of characters in the reasoning chain.
- **Findings**
  - Performs ~ the same as the baseline.

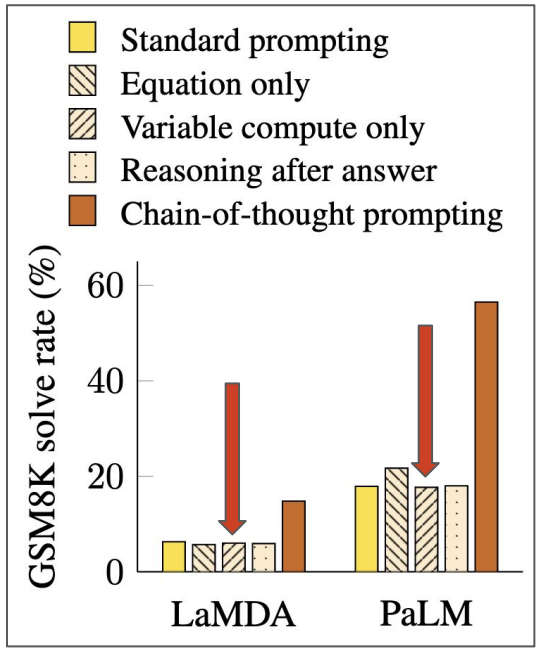


Fig source: https://arxiv.org/abs/2201.11903

Analysis: What Matters in CoT? (2/3)

- **Implications**
  - Naively increasing token computation by itself is not the reason for the success of CoT prompting.
  - There appears to be utility from expressing intermediate steps via natural language.

Analysis: What Matters in CoT? (3/3)

- Answer-after-CoT prompting
  - Assumption: CoT allow LMs to arrive at a better state that can better access relevant knowledge acquired during pretraining.

Analysis: What Matters in CoT? (3/3)

- Answer-after-CoT prompting
  - Assumption: CoT allow LMs to arrive at a better state that can better access relevant knowledge acquired during pretraining.
- **Findings**
  - Performs ~ the same as the baseline.

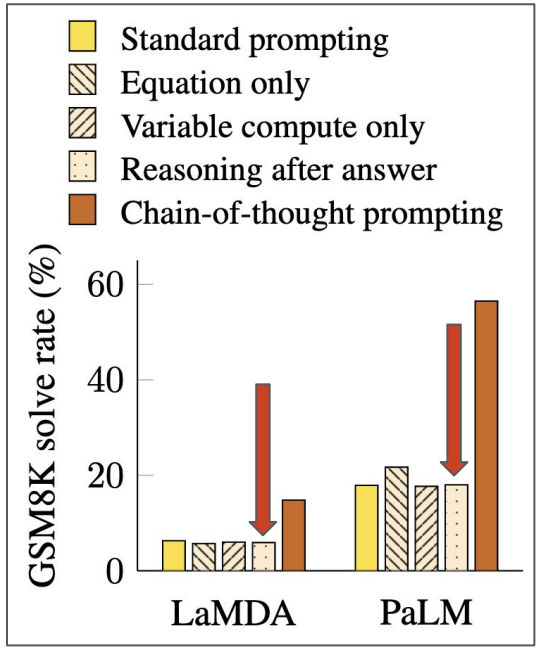


Fig source: https://arxiv.org/abs/2201.11903

Analysis: What Matters in CoT? (3/4)

- **Implications**
  - The **sequential reasoning** embodied in CoT is useful for reasons beyond just activating knowledge.

Analysis: How Robust is CoT? (How Sensitive CoT is to Prompt Engineering?)

- Sensitivity to exemplars is a key consideration of prompting approaches.
  - Different annotators
  - Annotators without machine learning background
  - Different demonstration split from the training data

Analysis: How Robust is CoT? (How Sensitive CoT is to Prompt Engineering?)

- **Findings**
  - There is variance among different chain of thought annotations
  - All sets of CoT prompts outperform the standard baseline by a large margin.

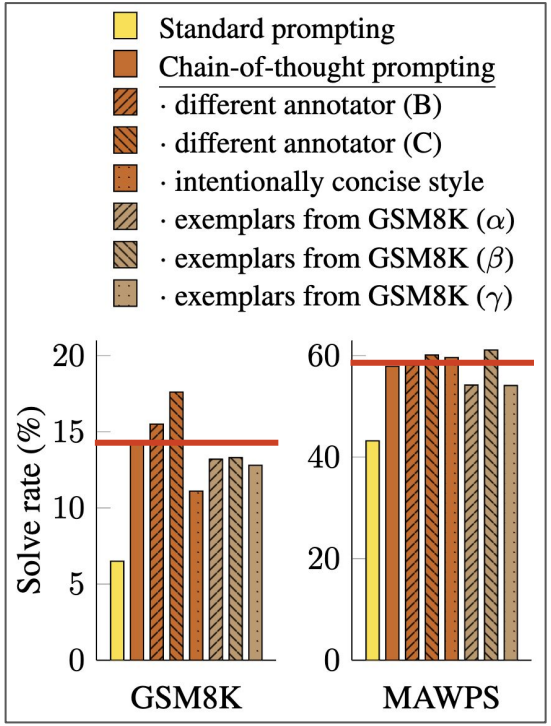


Fig source: https://arxiv.org/abs/2201.11903

Analysis: How Robust is CoT? (How Sensitive CoT is to Prompt Engineering?)

- **Findings**
  - There is variance among different chain of thought annotations
  - All sets of CoT prompts outperform the standard baseline by a large margin.
- **Implications**
  - Successful use of chain of thought **does not depend on a particular linguistic style**.

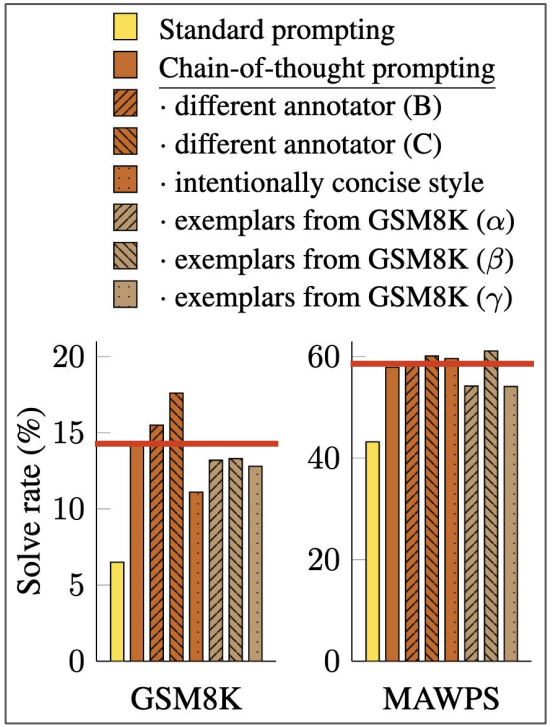


Fig source: https://arxiv.org/abs/2201.11903

Limitations

- CoT reasoning is an emergent property that requires sufficiently large models to be effective.
  - CoT actually hurts performance for most models < 10B parameters.
- Curating CoT demonstrations demands more effort compared to conventional “answer-only” few-shot prompting.
- CoT prompting induces longer output sequences → leads to a higher inference cost.

LEAST-TO-MOST PROMPTING ENABLES COMPLEX REASONING IN LARGE LANGUAGE MODELS

Denny Zhou[†*] **Nathanael Schärli**[†] **Le Hou**[†] **Jason Wei**[†] **Nathan Scales**[†] **Xuezhi Wang**[†]
Dale Schuurmans[†] **Claire Cui**[†] **Olivier Bousquet**[†] **Quoc Le**[†] **Ed Chi**[†]
[†]Google Research, Brain Team

Fig source: https://arxiv.org/abs/2205.10625

Technical Motivation

- The **easy-to-hard generalization** challenge for CoT
  - CoT often performs poorly on tasks that require generalization of **solving problems much harder than the demonstration** examples.

Least-to-Most Prompting

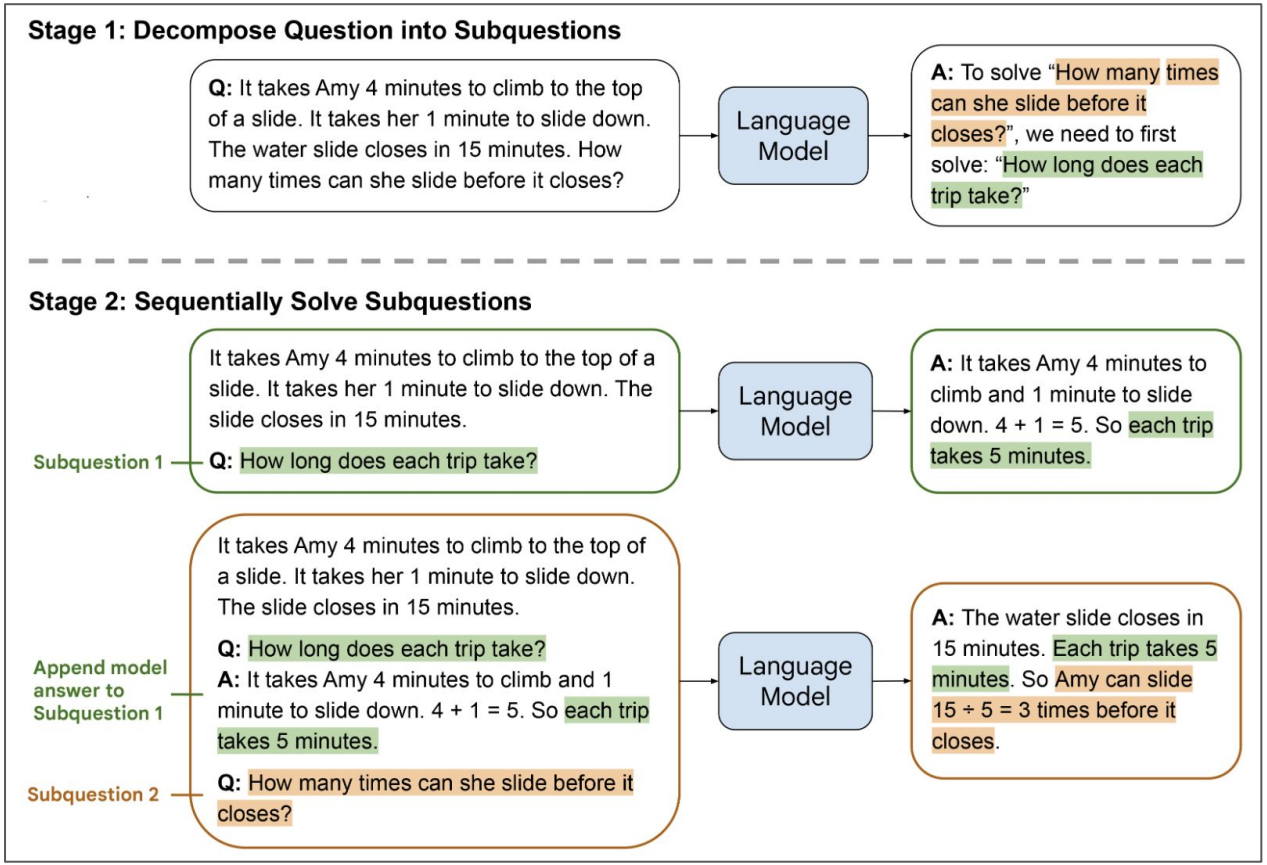
- Two stages
  - **Decomposition**
    - The prompt in this stage contains constant examples that demonstrate the decomposition, followed by the specific question to be decomposed.
  - **Subproblem solving**
    - The prompt in this stage consists of three parts: *(1)* constant examples demonstrating how subproblems are solved; *(2)* previously answered subquestions and generated solutions, and *(3)* the question to be answered next.

Least-to-Most Prompting

- Core idea - Teach LMs how to solve a complex problem by **decomposing it to a series of simpler subproblems**.
- Could be considered as a special instantiation of CoT, where we layout a specific kind of thinking plan for the model.

Fig source: https://arxiv.org/abs/2201.11903

Least-to-Most Prompting



Experiments & Main Results

- Symbolic reasoning
- Compositional generalization
- Math reasoning

Symbolic Reasoning

- *Last-letter-concatenation* task.

Q: “think, machine, learning”
A: “think”, “think, machine”, “think, machine, learning”

Table 1: Least-to-most prompt context (decomposition) for the last-letter-concatenation task. It can decompose arbitrary long lists into sequential subsists with an accuracy of 100%.

Q: “think, machine”
A: The last letter of “think” is “k”. The last letter of “machine” is “e”. Concatenating “k”, “e” leads to “ke”. So, “think, machine” outputs “ke”.

Q: “think, machine, learning”
A: “think, machine” outputs “ke”. The last letter of “learning” is “g”. Concatenating “ke”, “g” leads to “keg”. So, “think, machine, learning” outputs “keg”.

Fig source: https://arxiv.org/abs/2201.11903

Symbolic Reasoning

- CoT performs poorly when the testing lists are much longer than the lists in the prompt demonstrations.
- Least-to-most prompting overcomes this limitation and significantly outperforms CoT prompting on length generalization.

	$L = 4$	$L = 6$	$L = 8$	$L = 10$	$L = 12$
Standard prompting	0.0	0.0	0.0	0.0	0.0
Chain-of-Thought	84.2	69.2	50.2	39.8	31.8
Least-to-Most	**94.0**	**88.4**	**83.0**	**76.4**	**74.0**

Fig source: https://arxiv.org/abs/2201.11903

Compositional Generalization

- The *SCAN* dataset.
  - Task - mapping natural language commands to action sequences

Command	Action Sequence
“look thrice after jump”	JUMP LOOK LOOK LOOK
“run left and walk”	TURN_LEFT RUN WALK
“look opposite right”	TURN_RIGHT TURN_RIGHT LOOK

Fig source: https://arxiv.org/abs/2201.11903

Compositional Generalization

- The *SCAN* dataset.
  - Task - mapping natural language commands to action sequences

Q: “look opposite right thrice after walk”
A: “look opposite right thrice” can be solved by: “look opposite right”, “look opposite right thrice”. “walk” can be solved by “walk”. So, “look opposite right thrice after walk” can be solved by: “look opposite right”, “look opposite right thrice”, “walk”.

Q: “look around right thrice and walk”
A: “look around right thrice” can be solved by: “look right”, “look around right”, “look around right thrice”. “walk” can be solved by “walk”. So, “look around right thrice and walk” can be solved by: “look right”, “look around right”, “look around right thrice”, “walk”.

Fig source: https://arxiv.org/abs/2201.11903

Compositional Generalization

- Least-to-most prompting achieves an accuracy of 99.7% under length split.

Method	Standard prompting	Chain-of-Thought	Least-to-Most
code-davinci-002	16.7	16.2	**99.7**
text-davinci-002	6.0	0.0	**76.0**
code-davinci-001	0.4	0.0	**60.7**

Fig source: https://arxiv.org/abs/2201.11903

Math Reasoning

- Datasets: *GSM8K* & *DROP*
- The difficulty of the problems are measured by the number of solving steps.

Math Reasoning

- GSM8K
  - Least-to-most prompting slightly improves CoT prompting: from 60.87% → 62.39%

Method	Non-football (DROP)	Football (DROP)	GSM8K
Zero-Shot	43.86	51.77	16.38
Standard prompting	58.78	62.73	17.06
Chain-of-Thought	74.77	59.56	60.87
Least-to-Most	**82.45**	**73.42**	**62.39**

Fig source: https://arxiv.org/abs/2201.11903

Math Reasoning

- GSM8K
  - Least-to-most prompting slightly improves CoT prompting: from 60.87% → 62.39%
  - For problems with >= 5 steps to be solved: from 39.07% → 45.23%

Accuracy by Steps (GSM8K)	All	2 Steps	3 Steps	4 steps	≥ 5 steps
Least-to-Most	**62.39**	74.53	**68.91**	**59.73**	**45.23**
Chain-of-Thought	60.87	**76.68**	67.29	59.39	39.07

Fig source: https://arxiv.org/abs/2201.11903

Math Reasoning

- DROP
  - Least-to-most prompting outperforms CoT prompting by a large margin.

Method	Non-football (DROP)	Football (DROP)	GSM8K
Zero-Shot	43.86	51.77	16.38
Standard prompting	58.78	62.73	17.06
Chain-of-Thought	74.77	59.56	60.87
Least-to-Most	**82.45**	**73.42**	**62.39**

Fig source: https://arxiv.org/abs/2201.11903

Limitations

- Least-to-most prompting has limited generalizabilities **across domains**.
  - New prompts must be designed to demonstrate decomposition for math reasoning vs symbolic reasoning.
- Least-to-most prompting has limited generalizabilities **within domain**.
  - For GSM8k tasks, different challenging problems might require different problem-specific decomposition demonstrations.

Tree of Thoughts: Deliberate Problem Solving with Large Language Models

Shunyu Yao
Princeton University

Dian Yu
Google DeepMind

Jeffrey Zhao
Google DeepMind

Izhak Shafran
Google DeepMind

Thomas L. Griffiths
Princeton University

Yuan Cao
Google DeepMind

Karthik Narasimhan
Princeton University

Fig source: https://arxiv.org/abs/2305.10601

Problem of Chain of Thought

- Single path reasoning
  - Complex tasks are hard to solve with one single forward
- Next token prediction
  - There’s no backtrack
  - There’s no status check

Problem of Chain of Thought

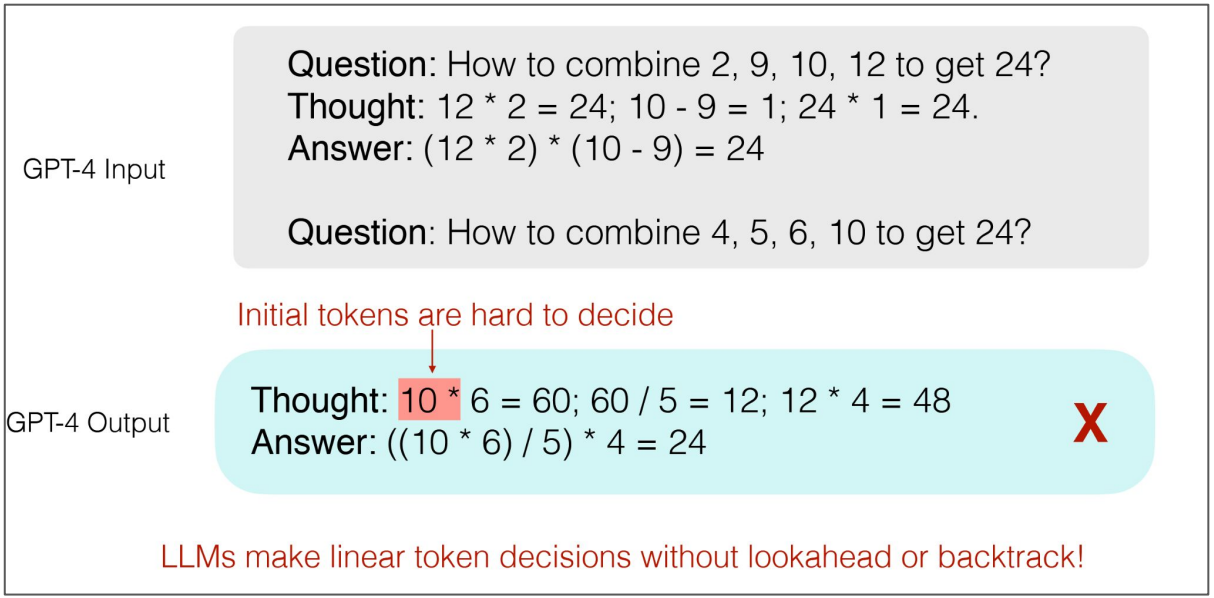


Fig source: https://ysymyth.github.io/papers/fpo.pdf

How to address these issues?

- Tree of Thought took inspirations from human cognition theory
  - System 1: fast and automatic (~next-token prediction)
  - System 2: slow and deliberate (~control algorithm)
- One of the oldest ideas in AI: Tree search

Comparison of Different Prompting Methods

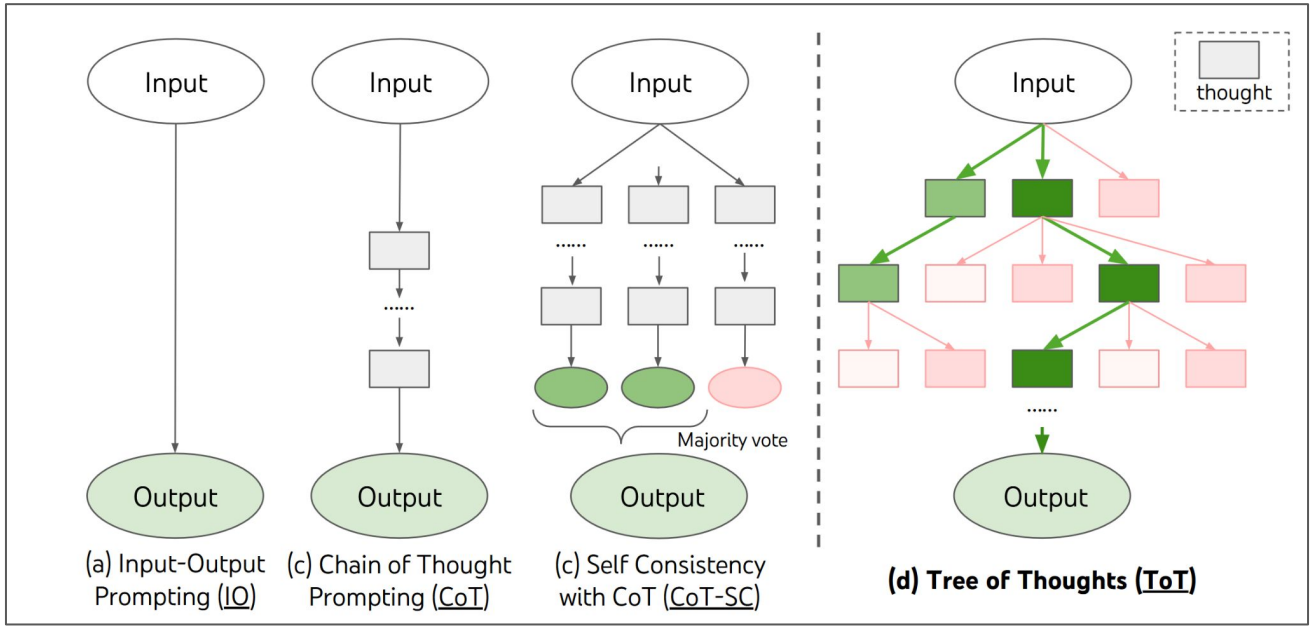


Fig source: http://arxiv.org/abs/2305.10601

Tree of Thought - Game of 24

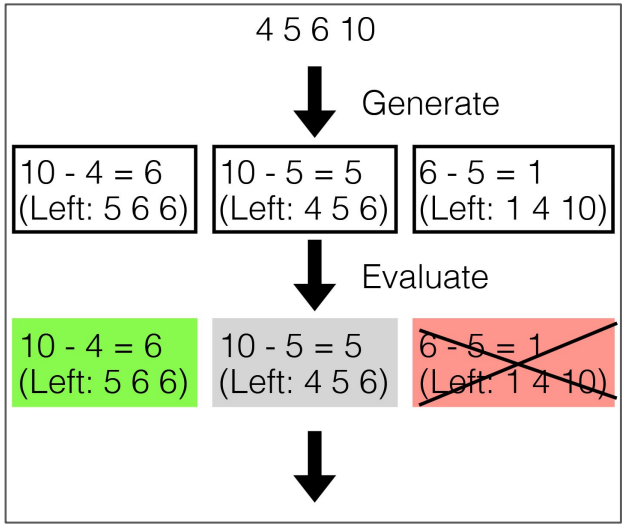
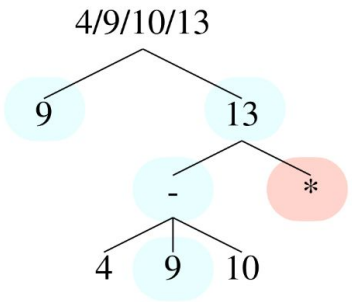


Fig source: https://ysymyth.github.io/papers/fpo.pdf

Thought granularity

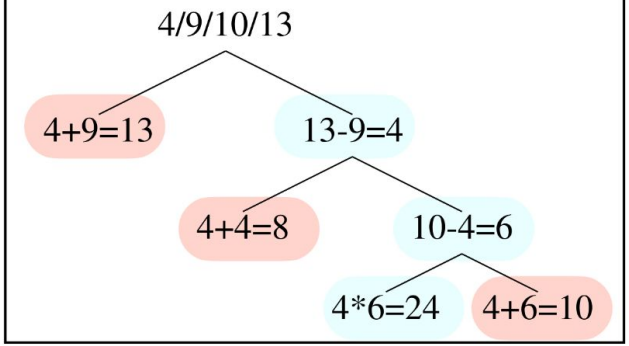
Each token as thought

- Easy to generate
- Hard to evaluate



Each equation as thought

- Relatively easy to generate/evaluate
- A problem-specific tradeoff design



Whole reasoning as thought

- Easy to evaluate
- Hard to generate

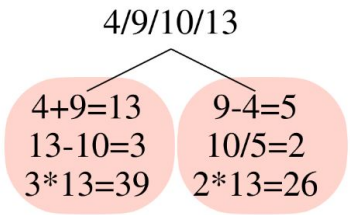


Fig source: https://ysymyth.github.io/papers/fpo.pdf

Experiment - Tasks

	Game of 24	Creative Writing	5x5 Crosswords
Input	4 numbers (4 9 10 13)	4 random sentences	10 clues (h1. presented;..)
Output	An equation to reach 24 (13-9)*(10-4)=24	A passage of 4 paragraphs ending in the 4 sentences	5x5 letters: SHOWN; WIRRA; AVAIL; ...
Thoughts	3 intermediate equations (13-9=4 (left 4,4,10); 10-4=6 (left 4,6); 4*6=24)	A short writing plan (1. Introduce a book that connects...)	Words to fill in for clues: (h1. shown; v5. naled; ...)
#ToT steps	3	1	5-10 (variable)

Table 1: Task overview. Input, output, thought examples are in blue.

Fig source: http://arxiv.org/abs/2305.10601

Experiment - Game of 24 Results

Method	Success
IO prompt	7.3%
CoT prompt	4.0%
CoT-SC (k=100)	9.0%
ToT (ours) (b=1)	45%
ToT (ours) (b=5)	**74%**
IO + Refine (k=10)	27%
IO (best of 100)	33%
CoT (best of 100)	49%

Table 2: Game of 24 Results.

Fig source: http://arxiv.org/abs/2305.10601

Analysis

- CoT scales better than IO, while ToT scales much better than CoT
- 60% CoT samples failed in first step, highlighting the issues with direct left-to-right decoding.

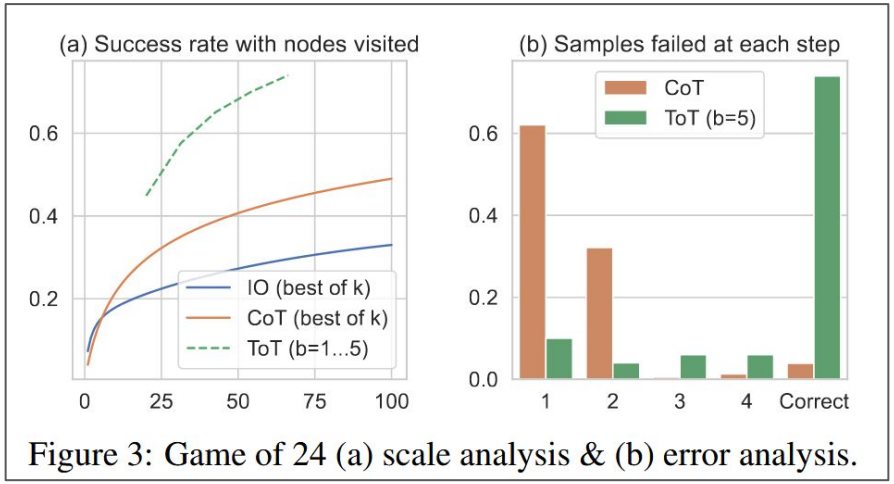


Figure 3: Game of 24 (a) scale analysis & (b) error analysis.

Fig source: http://arxiv.org/abs/2305.10601

Tree of Thought - Creative writing

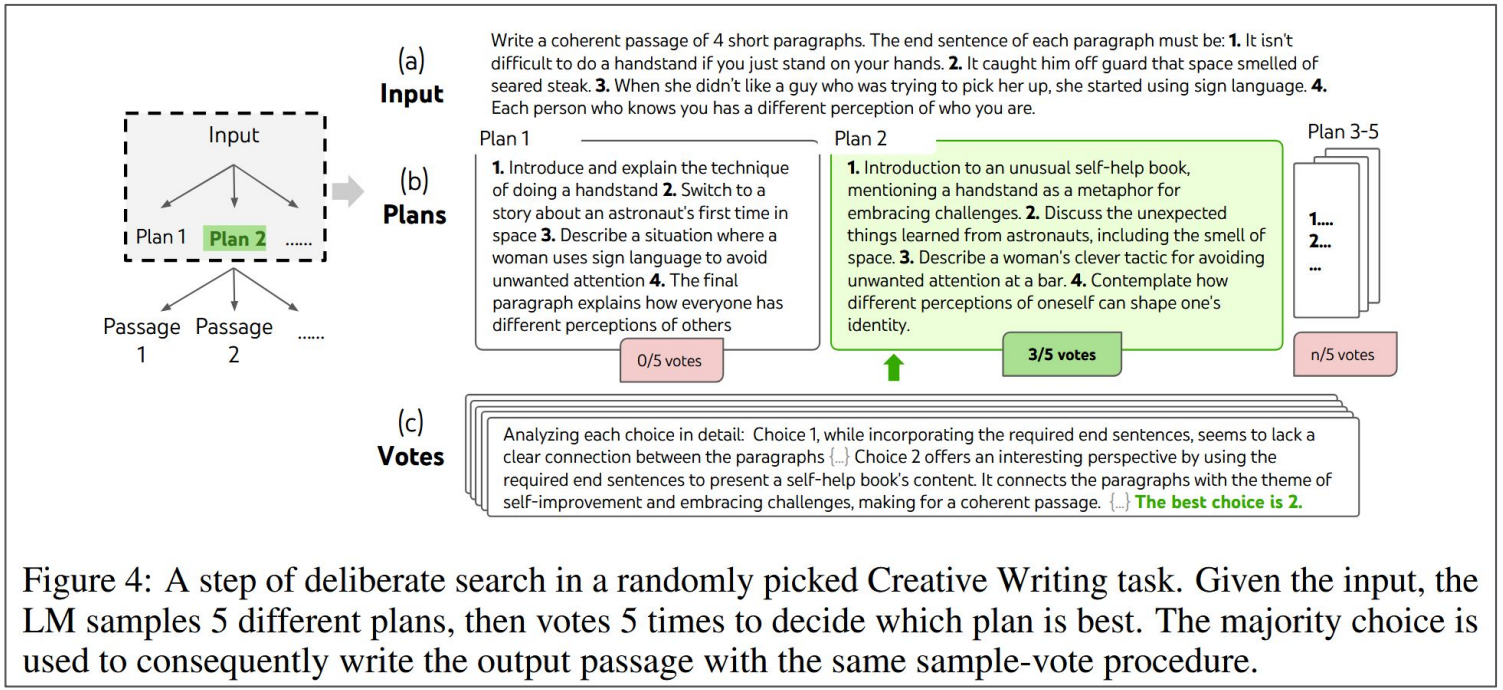


Figure 4: A step of deliberate search in a randomly picked Creative Writing task. Given the input, the LM samples 5 different plans, then votes 5 times to decide which plan is best. The majority choice is used to consequently write the output passage with the same sample-vote procedure.

Fig source: http://arxiv.org/abs/2305.10601

Experiment - Creative Writing Results

- ToT got 7.56 score, indicating that it can generate more coherent passages than IO (6.19) and CoT (6.93) on average.
- humans prefer ToT over CoT in 41 out of 100 passage pairs, while only prefer CoT over ToT in 21

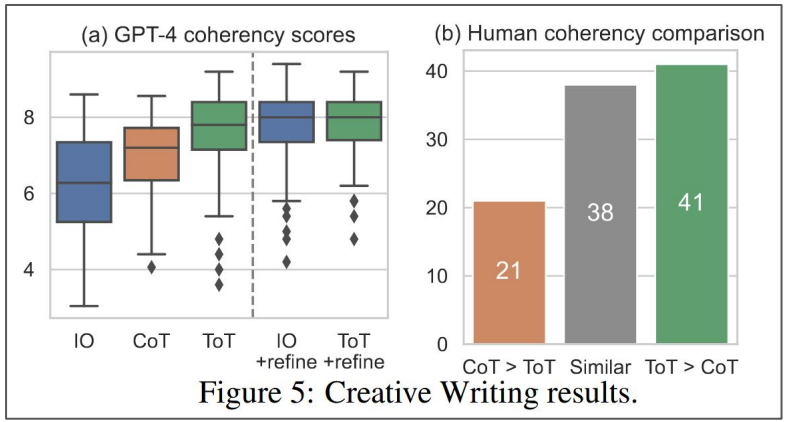


Figure 5: Creative Writing results.

Fig source: http://arxiv.org/abs/2305.10601

Limitations

- ToT might not be necessary for many existing tasks that current LLMs already excels at, like Math, Code
- ToT costs a lot of computation
- Defining a step of reasoning can be difficult for some real-world tasks

Thanks for your time!

Q&A