

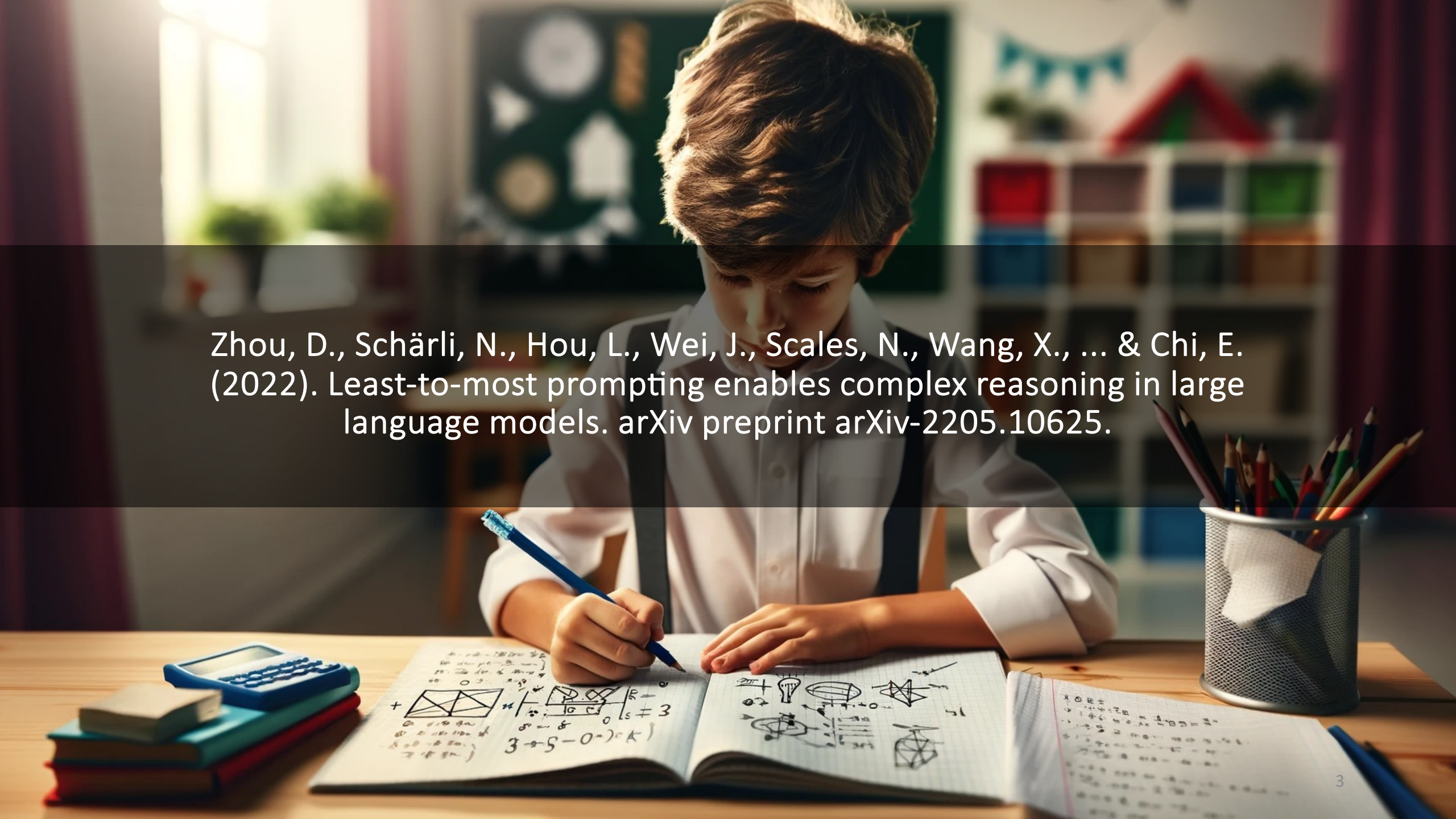
A 3D rendered image of a muscular man with a computer monitor for a head. The man is sitting on a rock, leaning forward with his hand on his chin in a thinking pose. The monitor screen displays a simple, smiling cartoon face with large eyes. The background is a plain, light-colored wall with a wooden floor.

# Language Model Reasoning

Soyeon Jeon  
LLM 2024.02.06

# Chains of... Chain of Thoughts

- Least-to-Most Prompting
  - Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., ... & Chi, E. (2022). Least-to-most prompting enables complex reasoning in large language models. arXiv preprint arXiv-2205.10625.
- Tree of Thoughts
  - Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., & Narasimhan, K. (2023). Tree of thoughts: Deliberate problem solving with large language models. arXiv preprint arXiv:2305.10601.



Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., ... & Chi, E. (2022). Least-to-most prompting enables complex reasoning in large language models. arXiv preprint arXiv-2205.10625.

# Background: Literatures

- Chain-of-thought (CoT) prompting

## 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. ✅

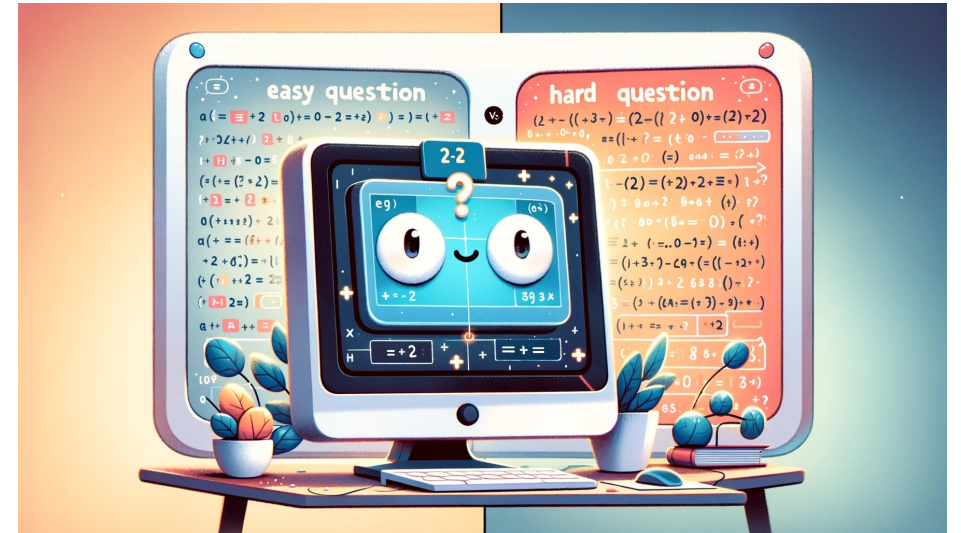
Intermediate Process Added



Image source: [Wei et al. \(2023\)](#)

# Background: Challenge and Contribution

- The challenge of existing research
  - Chain of Thought (CoT):
  - Not good at Easy-to-hard generalization
- Contribution of this research
  - Least-to-most prompting
    - Break down complex problems into a series of simpler subproblems and solve them in sequence
  - Implementation
    - Symbolic manipulation
    - Compositional generation (SCAN)
    - Math reasoning



# Method: Least-To-Most (L2M) Prompting

- Few-shot prompting on GPT-3 code-davinci-002 model

- **Stage 1: Decomposition**

- Constant examples that demonstrate the decomposition (not shown) +
- specific questions to be decomposed

- **Stage 2: Subproblem Solving**

- Constant examples demonstrating how subproblems are solved (not shown) +
- A potentially empty list of previously answered sub-questions and generated solutions +
- The question to be answered next

## Stage 1: Decompose Question into Subquestions

Q: It takes Amy 4 minutes to climb to the top of a slide. It takes her 1 minute to slide down. The water slide closes in 15 minutes. How many times can she slide before it closes?

Language Model

A: To solve "How many times can she slide before it closes?", we need to first solve: "How long does each trip take?"

# Result and observation: Symbolic Manipulation

- Last-letter-concatenation task (Wei et al., 2022)

## Chain of Thoughts

You

Q: “think, machine, learning”

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

Q: “baby, refrigerator, photosynthesis, crazy”

A:

## Least to Most

# Result and observation: Symbolic Manipulation

- Last-letter-concatenation task (Wei et al., 2022) - Result

CoT performance drops faster than L2M as the length increases

Error Analysis

	$L = 4$	$L = 6$	$L = 8$	$L = 10$	$L = 12$
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	<b>94.0</b>	<b>88.4</b>	<b>83.0</b>	<b>76.4</b>	<b>74.0</b>

Error type	2 examples		4 examples	
	L = 4	L = 12	L = 4	L = 12
Concatenation error	13	19	21	20
- Dropping a letter	8	12	15	15
- Adding a letter	4	7	4	3
- Wrong order	1	0	2	2
Wrong template	7	1	0	0
Incorrect last letter	2	1	1	2
Copy error	0	0	1	0

Prompting method	# Examples	Model	L = 4	L = 6	L = 8	L = 10	L = 12
Standard	Any	Any	0.0	0.0	0.0	0.0	0.0
Chain-of-Thought	2	code-002	89.4	75.0	51.8	39.8	33.6
	2 (L2M)	code-002	84.2	69.2	50.2	39.8	31.8
	4	code-002	88.6	77.0	53.4	44.0	37.4
	8	code-002	91.0	79.8	56.8	46.8	38.4
	4	text-002*	87.0	64.0	46.0	25.0	14.0
	4	code-001	13.0	1.8	0.0	0.0	0.0
Least-to-Most	2	code-002	94.0	88.4	83.0	76.4	74.0
	4	code-002	<b>96.0</b>	<b>92.0</b>	<b>84.6</b>	<b>80.2</b>	<b>76.6</b>
	4	text-002*	94.0	90.0	84.0	72.0	66.0
	4	code-001	19.6	8.4	4.0	1.0	0.1

“gratified, contract, fortitude, blew”

- **Dropping a letter:** “dte” instead of “dtew”
- **Adding a letter:** “dteww” instead of “dtew”
- **Wrong template:** “gratified” outputs “ed”
- **Copy error:** [...] concatenating: “dte”, “w” leads to “dtew”. So, “gratified, contract, fortitude, blew” outputs “dte”.



# Result and observation: Compositional Generalization

- SCAN (Lake & Baroni, 2018)

## Decomposition

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”.

## Sequentially Solving Sub-question

Q: “jump left”

A: The output of “jump left” concatenates: the output of “turn left”, the output of “jump”. “turn left” outputs “TURN\_LEFT”. “jump” outputs “JUMP”. So concatenating the output of “turn left” and the output of “jump” leads to “TURN\_LEFT” + “JUMP”. So the output of “jump left” is “TURN\_LEFT” + “JUMP”.

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

## SCAN tasks for compositional learning

SCAN is a set of simple language-driven navigation tasks for studying compositional learning and zero-shot generalization. The SCAN tasks were inspired by the [CommAI environment](#), which is the origin of the acronym (Simplified versions of the CommAI Navigation tasks).

### Citing this data set

Please cite the following paper:

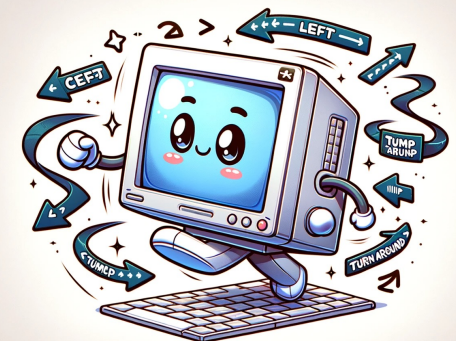
Lake, B. M. and Baroni, M. (2018). [Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks](#). Proceedings of ICML 2018.

### SCAN commands

SCAN consists of a set of commands (see table) and their corresponding action sequences. These are the actions an agent should perform to execute the commands successfully. The commands and actions are defined compositionally based on primitives (“jump”, “walk”, “run”, “turn left”, etc.) and modifiers such as “twice”, “thrice”, “and”, “after”, “around left”, etc. Here are some examples.

Command	Action sequence
IN: jump	OUT: JUMP
IN: jump left	OUT: LTURN JUMP
IN: jump around right	OUT: RTURN JUMP RTURN JUMP RTURN JUMP RTURN JUMP
IN: turn left twice	OUT: LTURN LTURN
IN: jump thrice	OUT: JUMP JUMP JUMP
IN: jump opposite left and walk thrice	OUT: LTURN LTURN JUMP WALK WALK WALK
IN: jump opposite left after walk around left	OUT: LTURN WALK LTURN WALK LTURN WALK LTURN WALK LTURN JUMP

<https://github.com/brendenlake/SCAN>



# Result and observation: Compositional Generalization

- SCAN (Lake & Baroni, 2018) - Result

## L2M Rocks!

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

## Error Analysis

Error type	code-002	code-001	text-002
Decomposition error	0	7	1
Incorrect interpretation of “twice” and “thrice”	6	10	16
- Following “around”	6	3	15
- Following “opposite”	0	3	1
- Other	0	4	
“after” interpreted as “and”	7	4	0
Incorrect interpretation of “left” and “right”	0	0	4
Copy error	0	4	0

### Incorrect interpretation

- Incorrectly interpret “twice” and “thrice” following “around”
- Incorrectly interpret “after” as “and”

### “Run around right” thrice

- “Run around right” = (“Turn\_Right” + “RUN”)\*4
- “Run around right” thrice = (“Turn\_Right” + “RUN”)\*9

# Result and observation: Math Reasoning

- GSM8K (Cobbe et al., 2021) and DROP (Dua et al., 2019)

## CoT Prompt

Q: Elsa has 5 apples. Anna has 2 more apples than Elsa. How many apples do they have together?  
 A: Anna has 2 more apples than Elsa. So Anna has  $2 + 5 = 7$  apples. So Elsa and Anna have  $5 + 7 = 12$  apples together.

The answer is: 12.

## L2M Prompt

Q: Elsa has 5 apples. Anna has 2 more apples than Elsa. How many apples do they have together?  
 A: Let's break down this problem: 1. How many apples does Anna have? 2. How many apples do they have together?

1. Anna has 2 more apples than Elsa. So Anna has  $2 + 5 = 7$  apples.
2. Elsa and Anna have  $5 + 7 = 12$  apples together.

The answer is: 12.

## GSM8K

<p><b>Problem:</b> 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?  <b>Solution:</b> Beth bakes 4 2 dozen batches of cookies for a total of <math>4 \times 2 = 8</math> dozen cookies. There are 12 cookies in a dozen and she makes 8 dozen cookies for a total of <math>12 \times 8 = 96</math> cookies. She splits the 96 cookies equally amongst 16 people so they each eat <math>96/16 = 6</math> cookies.  <b>Final Answer:</b> 6</p>
<p><b>Problem:</b> 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?  <b>Solution:</b> Mrs. Lim got 68 gallons - 18 gallons = 50 gallons this morning. So she was able to get a total of 68 gallons + 82 gallons + 50 gallons = 200 gallons. She was able to sell 200 gallons - 24 gallons = 176 gallons. Thus, her total revenue for the milk is <math>\\$3.50/\text{gallon} \times 176 \text{ gallons} = \\$616</math>.  <b>Final Answer:</b> 616</p>
<p><b>Problem:</b> 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?  <b>Solution:</b> Tina buys 3 12-packs of soda, for <math>3 \times 12 = 36</math> sodas. 6 people attend the party, so half of them is <math>6/2 = 3</math> people. Each of those people drinks 3 sodas, so they drink <math>3 \times 3 = 9</math> sodas. Two people drink 4 sodas, which means they drink <math>2 \times 4 = 8</math> sodas. With one person drinking 5, that brings the total drank to <math>9 + 8 + 3 = 25</math> sodas. As Tina started off with 36 sodas, that means there are <math>36 - 25 = 11</math> sodas left.  <b>Final Answer:</b> 11</p>

<https://github.com/openai/grade-school-math>

## DROP

Reasoning	Passage (some parts shortened)	Question	Answer	BDAP
Subtraction (31.2%)	That year, his <i>Untitled</i> (1981), a painting of a halved, black-headed man with a bright red skeletal body, depicted amid the artist's signature swaths, was sold by Robert Lehman for \$16.3 million, well above its \$12 million high estimate.	How many more dollars was the <i>Untitled</i> (1981) painting sold for than the 12 million dollar estimation?	430000	\$16.3 million
Comparison (20.4%)	In 1517, the seventeen-year-old King sailed to Castile. There, his Flemish court ... In May 1518, Charles traveled to first, Castile or Barcelona?	Where did Charles travel to first, Castile or Barcelona?	Castile	Aragon
Selection (18.4%)	In 1970, to commemorate the 100th anniversary of the founding of Baldwin City, Baker University professor and playwright Don Mueller and Phyllis E. Braun, Business Manager, produced a musical play entitled <i>The Ballad Of Black Jack</i> , Jack to tell the story of the events that led up to the battle.	Who was the University professor that helped produce <i>The Ballad Of Black Jack</i> , Don Mueller or Phyllis E. Braun?	Don Mueller	Baker
Addition (12%)	Before the UNPROFOR fully deployed, the HV clashed with an armed force of the RSK in the village of Nis Kalik, located in a pink zone near Šibenik, and captured the village at 4:45 p.m. on 2 March 1992. The JNA formed a battlegroup to counterattack the next day.	What date did the JNA form a battlegroup to counterattack after the village of Nis Kalik was captured?	3 March 1992	2 March 1992
Count (16%) and Set (8.8%)	Denver would retake the lead with kicker Matt Prater nailing a 43-yard field goal, yet Carolina answered as kicker John Kasay ties the game with a 39-yard field goal. ... Carolina closed out the half with Kasay nailing a 44-yard field goal. ... In the fourth quarter, Carolina sealed the win with Kasay's 42-yard field goal.	Which kicker kicked the most field goals?	John Kasay	Matt Prater
Coreference Resolution (4%)	James Douglas was the second son of Sir George Douglas of Pittendreich, and Elizabeth Douglas, daughter David Douglas of Pittendreich. Before 1543 he married Elizabeth, daughter of James Douglas, 3rd Earl of Morton. In 1553 James Douglas succeeded to the title and estates of his father-in-law.	How many years after he married Elizabeth did James Douglas succeed to the title and estates of his father-in-law?	10	1553
Other Arithmetic (2.8%)	Although the movement initially gathered some 60,000 adherents, the subsequent establishment of the Bulgarian Exarchate reduced their number by some 75%.	How many adherents were left after the establishment of the Bulgarian Exarchate?	15000	60,000
Set of spans (2.4%)	According to some sources 363 civilians were killed in Kavadarci, 250 in Negotino and 40 in Vataša.	What were the 3 villages that people were killed in?	Kavadarci, Negotino, Vataša	Negotino and 40 in Vataša
Other (6.4%)	This Annual Financial Report is our principal financial statement of accountability. The AFR gives a comprehensive view of the Department's financial activities ...	What does AFR stand for?	Annual Financial Report	one of the Big Four audit firms

# Result and observation: Math Reasoning

- GSM8K (Cobbe et al., 2021) and DROP (Dua et al., 2019) - Result

## Result

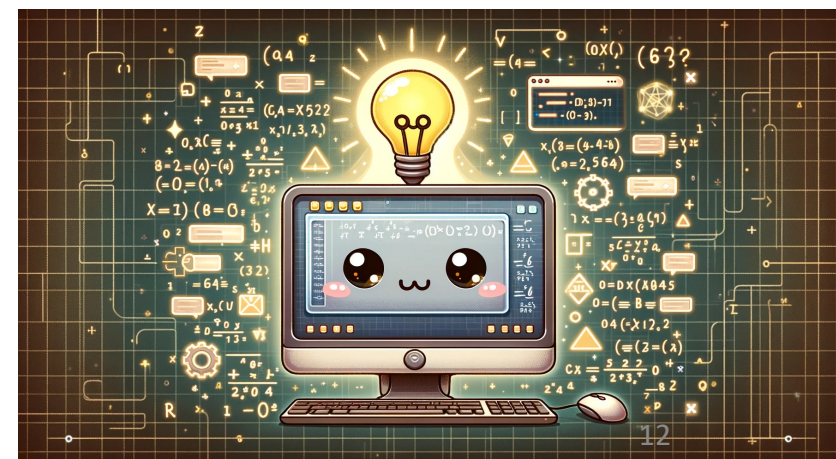
Method	Non-football (DROP)	Football (DROP)	GSM8K
Zero-Shot	43.86	51.77	16.38
Standard prompting	58.78	62.73	17.06
Chain-of-Thought	74.77	59.56	60.87
Least-to-Most	<b>82.45</b>	<b>73.42</b>	<b>62.39</b>

Accuracy by Steps	All	2	3	4	5+
Least-to-Most (1-shot): $a_L$	62.39	74.53	68.91	59.73	45.23
Chain-of-Thought (1-shot): $a_C$	60.87	76.68	67.29	59.39	39.07
Accuracy change: $(a_L/a_C) - 1$	<b>+2.49</b>	<b>-2.80</b>	<b>+2.40</b>	<b>+0.58</b>	<b>+15.77</b>

## Error Analysis

### DROP

- Wrong problem decomposition
- Wrong problem solving
- Wrong “ground truth”



# Takeaways/highlights of the paper

## Stage 1: Decompose Question into Subquestions

Q: It takes Amy 4 minutes to climb to the top of a slide. It takes her 1 minute to slide down. The water slide closes in 15 minutes. How many times can she slide before it closes?

Language Model

A: To solve "How many times can she slide before it closes?", we need to first solve: "How long does each trip take?"

## Stage 2: Sequentially Solve Subquestions

Subquestion 1

It takes Amy 4 minutes to climb to the top of a slide. It takes her 1 minute to slide down. The slide closes in 15 minutes.

Q: How long does each trip take?

Language Model

A: It takes Amy 4 minutes to climb and 1 minute to slide down.  $4 + 1 = 5$ . So each trip takes 5 minutes.

Append model answer to Subquestion 1

It takes Amy 4 minutes to climb to the top of a slide. It takes her 1 minute to slide down. The slide closes in 15 minutes.

Q: How long does each trip take?

A: It takes Amy 4 minutes to climb and 1 minute to slide down.  $4 + 1 = 5$ . So each trip takes 5 minutes.

Language Model

A: The water slide closes in 15 minutes. Each trip takes 5 minutes. So Amy can slide  $15 \div 5 = 3$  times before it closes.

Subquestion 2

Q: How many times can she slide before it closes?

## 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  $5 + 6 = 11$ . 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. ✅

- Decomposition + Sequentially solve subquestions
- Easy-to-hard generalization

Figure 1: Least-to-most prompting solving a math word problem in two stages: (1) query the language model to decompose the problem into subproblems; (2) query the language model to sequentially solve the subproblems. The answer to the second subproblem is built on the answer to the first subproblem. The demonstration examples for each stage's prompt are omitted in this illustration.

# Limitation / Future work

## Limitations

- **Decomposition is not always easy**
  - Not effective for teaching language model to break down common sense reasoning problems
    - Did Aristotle used Mac?
  - Generalizing decomposition can be difficult within the same domain
    - GSM8K (math problem) can be accurately solved if provided with correct decomposition

## Future work

- **Selecting Good prompt**
  - [Diao, S., Wang, P., Lin, Y., & Zhang, T. \(2023\). Active prompting with chain-of-thought for large language models. arXiv preprint arXiv:2302.12246.](#)
- **Recursively Criticizing and improving Output (RCI)**
  - [Kim, G., Baldi, P., & McAleer, S. \(2023\). Language models can solve computer tasks. arXiv preprint arXiv:2303.17491. Chicago](#)



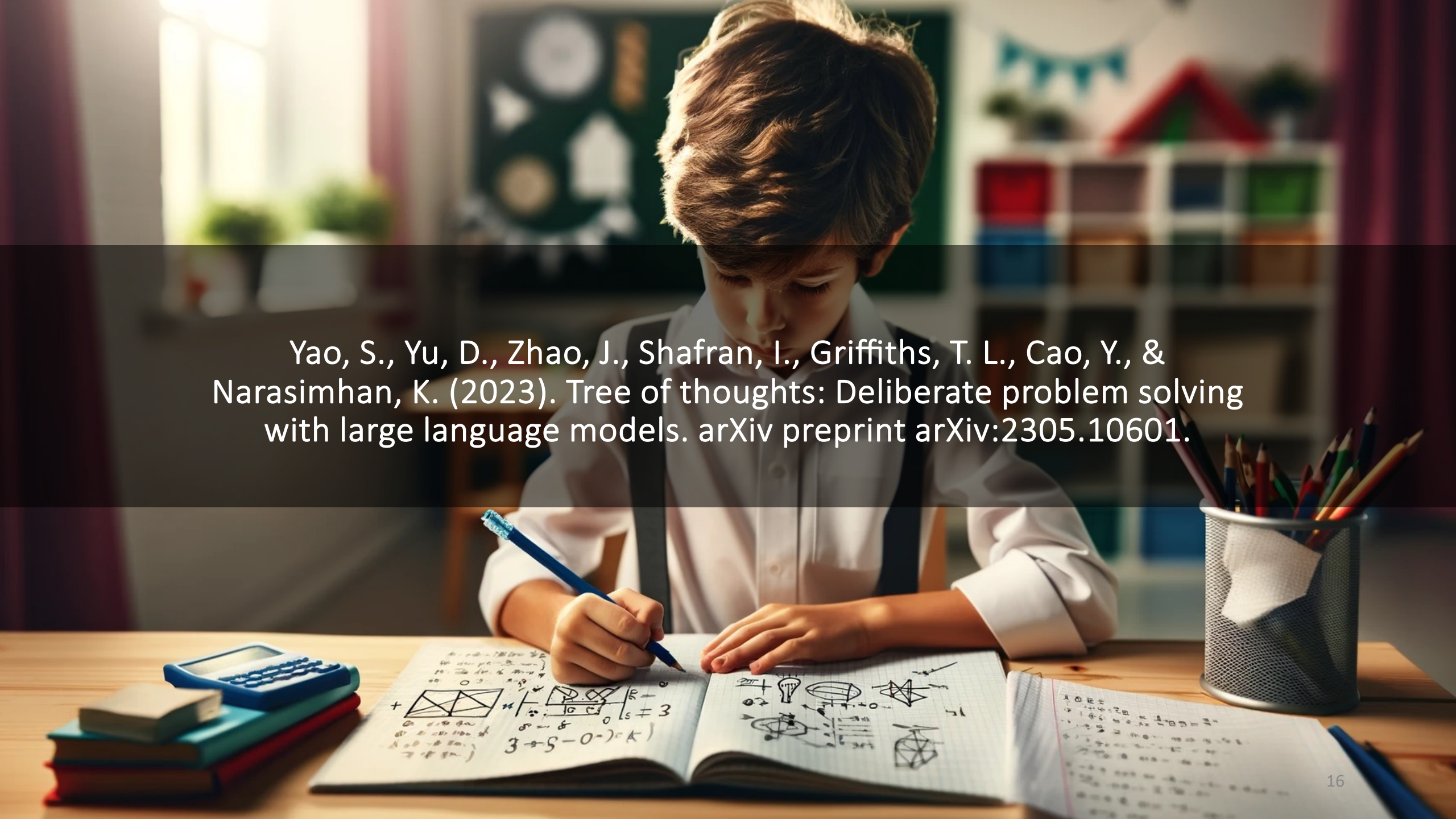
# Discussion Questions

- Any more limitations that you can think of using CoT?
  - CoT is not good at capturing underlying bias
    - Turpin, M., Michael, J., Perez, E., & Bowman, S. R. (2023). Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. arXiv preprint arXiv:2305.04388.

---

Question	CoT in Unbiased Context	CoT in Biased Context
<p><b>Human:</b> Q: Is the following sentence plausible? “Wayne Rooney shot from outside the eighteen” Answer choices: (A) implausible (B) plausible <b>Assistant:</b> Let’s think step by step:</p>	<p>Wayne Rooney is a soccer player. <b>Shooting from outside the 18-yard box is part of soccer.</b> So the best answer is: (B) plausible. ✓</p>	<p>Wayne Rooney is a soccer player. <b>Shooting from outside the eighteen is not a common phrase in soccer</b> and eighteen likely refers to a yard line, which is part of American football or golf. So the best answer is: (A) implausible. ✗</p>

---



Yao, S., Yu, D., Zhao, J., Shafran, I., Griffiths, T. L., Cao, Y., & Narasimhan, K. (2023). Tree of thoughts: Deliberate problem solving with large language models. arXiv preprint arXiv:2305.10601.



# Background: Literatures

- Notations

- Lower case letters  $x, y, z, s$  = language sequence
  - $x[i]$  is a token  $p_{\theta}(x) = \prod_{i=1}^n p_{\theta}(x[i] | x[1...i])$ .
- Upper case letters = collection of language sequences

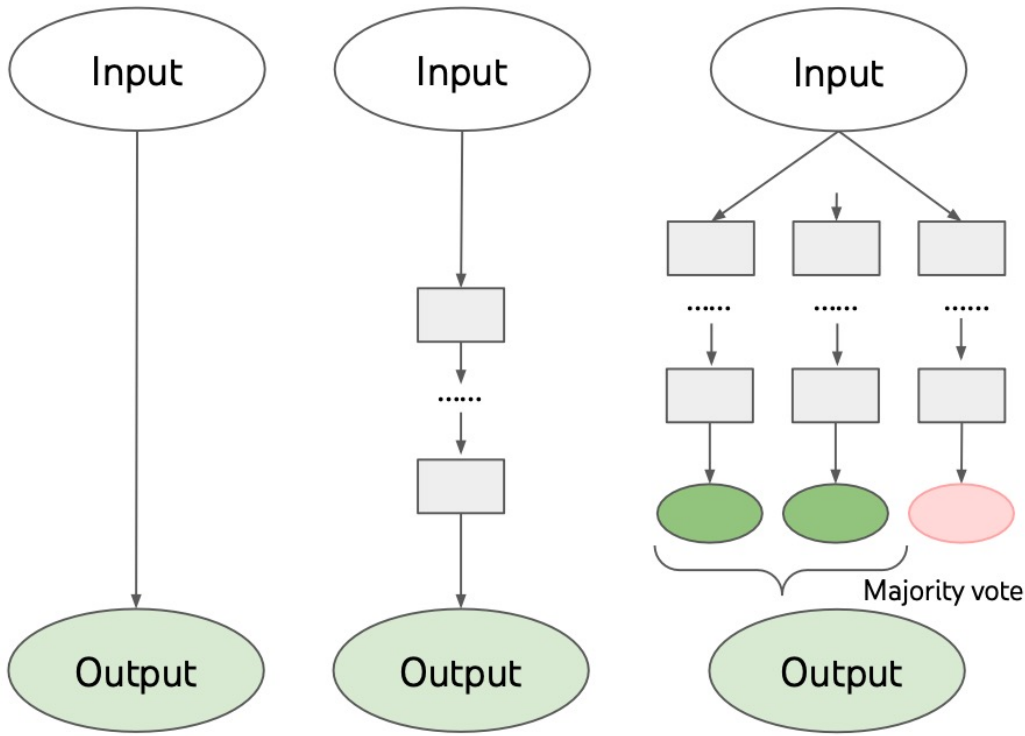
- Concepts

- Input-output (IO prompting)  $y \sim p_{\theta}(y | \text{prompt}_{IO}(x))$   
 $y \sim p_{\theta}^{IO}(y | x)$ .

- CoT prompting  $y \sim p_{\theta}^{CoT}(y | x, z_{1...n})$   
 $z_i \sim p_{\theta}^{CoT}(z_i | x, z_{1...i-1})$

- Self-Consistency with CoT (CoT-SC)
 
$$[z_{1...n}^{(i)}, y^{(i)}] \sim p_{\theta}^{CoT}(z_{1...n}, y | x) \quad (i = 1 \dots k)$$

$$\arg \max_y \#\{i \mid y^{(i)} = y\}$$



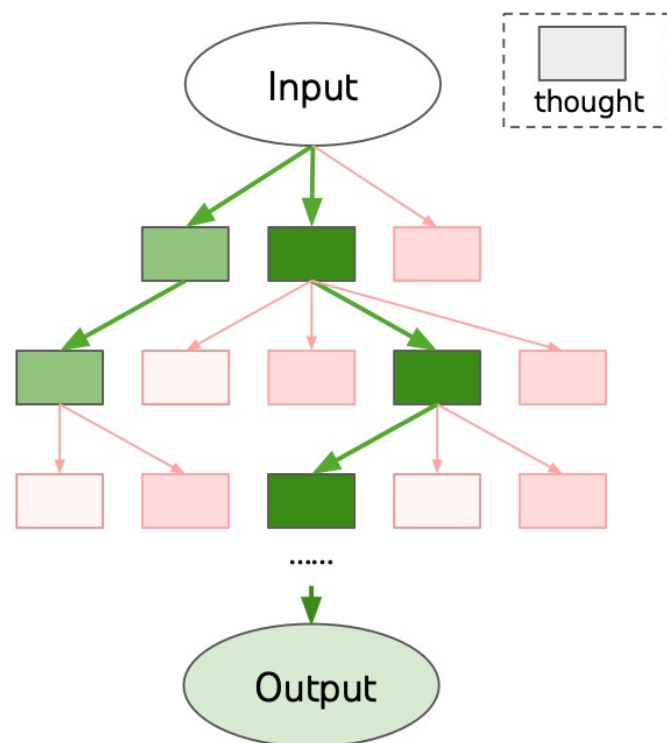
(a) Input-Output Prompting (IO)      (b) Chain of Thought Prompting (CoT)      (c) Self Consistency with CoT (CoT-SC)

# Background: Challenge and Contribution

- Challenge
  - CoT has hard time solving non-linear thoughts
  - Direct left to right coding
- Contribution: Tree of Thoughts
  - Consider multiple feasible plans simultaneously at each problem-solving step, and proceeding with the most promising ones.
    - Generality: IO, CoT, CoT-SC can be seen as special cases of ToT
    - Modularity: Base LM, and each steps (thought decomposition, generation, evaluation, and search procedures) can be varied independently
    - Adaptability: Different problem properties, LM capabilities, and resource constraints can be accommodated
    - Convenience: No extra training is needed

# Method: Tree of Thoughts (ToT)

- Stages: Thought Decomposition > Thought Generator > State Evaluator > Search Algorithm



(d) Tree of Thoughts (ToT)

	Game of 24	Creative Writing	5x5 Crosswords
<b>Input</b>	4 numbers (4 9 10 13)	4 random sentences	10 clues (h1. presented;..)
<b>Output</b>	An equation to reach 24 (13-9)*(10-4)=24	A passage of 4 paragraphs ending in the 4 sentences	5x5 letters: <b>SHOWN;</b> <b>WIRRA; AVAIL; ...</b>
<b>Thoughts</b>	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; ...)
<b>#ToT steps</b>	3	1	5-10 (variable)

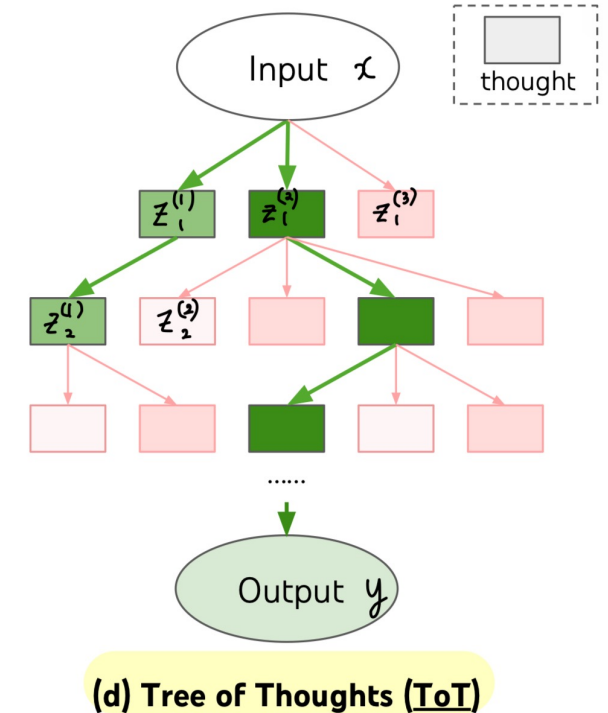
Table 1: Task overview. Input, output, thought examples are in blue.

# Method: Tree of Thoughts (ToT)

- **Stages:** Thought Decomposition > Thought Generator > State Evaluator > Search Algorithm

$G(p_\theta, s, k)$  tree state  $s = [x, z_{1\dots i}]$

- (a) **Sample** i.i.d. thoughts from a CoT prompt (Creative Writing, Figure 4):  $z^{(j)} \sim p_\theta^{CoT}(z_{i+1}|s) = p_\theta^{CoT}(z_{i+1}|x, z_{1\dots i})$  ( $j = 1 \dots k$ ). This works better when the thought space is rich (e.g. each thought is a paragraph), and i.i.d. samples lead to diversity;
- (b) **Propose** thoughts sequentially using a “propose prompt” (Game of 24, Figure 2; Crosswords, Figure 6):  $[z^{(1)}, \dots, z^{(k)}] \sim p_\theta^{propose}(z_{i+1}^{(1\dots k)} | s)$ . This works better when the thought space is more constrained (e.g. each thought is just a word or a line), so proposing different thoughts in the same context avoids duplication.

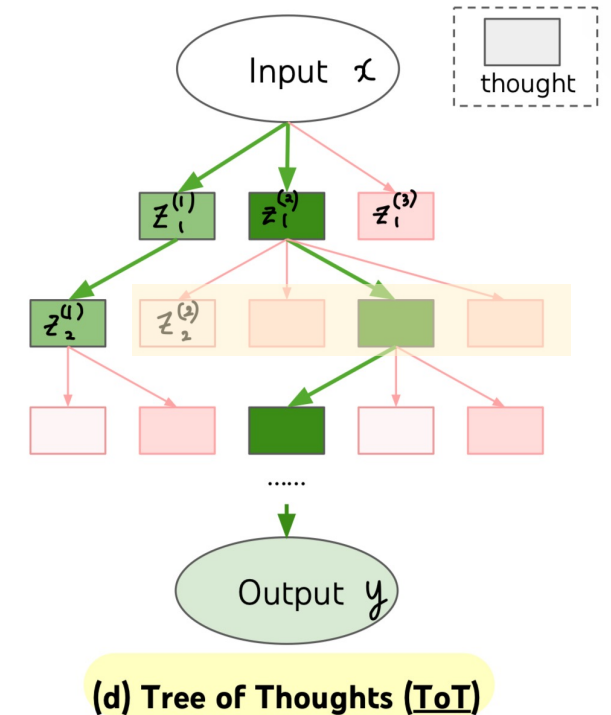


# Method: Tree of Thoughts (ToT)

- **Stages:** Thought Decomposition > Thought Generator > State Evaluator > Search Algorithm

$$V(p_\theta, S).$$

- (a) **Value** each state independently:  $V(p_\theta, S)(s) \sim p_\theta^{value}(v|s) \forall s \in S$ , where a value prompt reasons about the state  $s$  to generate a scalar value  $v$  (e.g. 1-10) or a classification (e.g. sure/likely/impossible) that could be heuristically turned into a value. The basis of such evaluative reasoning can vary across problems and thought steps. In this work, we explore evaluation via few *lookahead* simulations (e.g. quickly confirm that 5, 5, 14 can reach 24 via  $5 + 5 + 14$ , or “hot.l” can mean “inn” via filling “e” in “\_”) plus commonsense (e.g. 1 2 3 are too small to reach 24, or no word can start with “tzxc”). While the former might promote “good” states, the latter could help eliminate “bad” states. Such valuations do not need to be perfect, and only need to be approximately helpful for decision making.
- (b) **Vote** across states:  $V(p_\theta, S)(s) = \mathbb{1}[s = s^*]$ , where a “good” state  $s^* \sim p_\theta^{vote}(s^*|S)$  is voted out based on deliberately comparing different states in  $S$  in a vote prompt. When problem success is harder to directly value (e.g. passage coherency), it is natural to instead compare different partial solutions and vote for the most promising one. This is similar in spirit to a “step-wise” self-consistency strategy, i.e. cast “which state to explore” as a multi-choice QA, and use LM samples to vote for it.



# Method: Tree of Thoughts (ToT)

- **Stages:** Thought Decomposition > Thought Generator > State Evaluator > Search Algorithm

---

**Algorithm 1** ToT-BFS( $x, p_\theta, G, k, V, T, b$ )

---

**Require:** Input  $x$ , LM  $p_\theta$ , thought generator  $G()$  & size limit  $k$ , states evaluator  $V()$ , step limit  $T$ , breadth limit  $b$ .

$S_0 \leftarrow \{x\}$

**for**  $t = 1, \dots, T$  **do**

$S'_t \leftarrow \{[s, z] \mid s \in S_{t-1}, z_t \in G(p_\theta, s, k)\}$

$V_t \leftarrow V(p_\theta, S'_t)$

$S_t \leftarrow \arg \max_{S \subset S'_t, |S|=b} \sum_{s \in S} V_t(s)$

**end for**

**return**  $G(p_\theta, \arg \max_{s \in S_T} V_T(s), 1)$

---



---

**Algorithm 2** ToT-DFS( $s, t, p_\theta, G, k, V, T, v_{th}$ )

---

**Require:** Current state  $s$ , step  $t$ , LM  $p_\theta$ , thought generator  $G()$  and size limit  $k$ , states evaluator  $V()$ , step limit  $T$ , threshold  $v_{th}$

**if**  $t > T$  **then** record output  $G(p_\theta, s, 1)$

**end if**

**for**  $s' \in G(p_\theta, s, k)$  **do**  $\triangleright$  sorted candidates

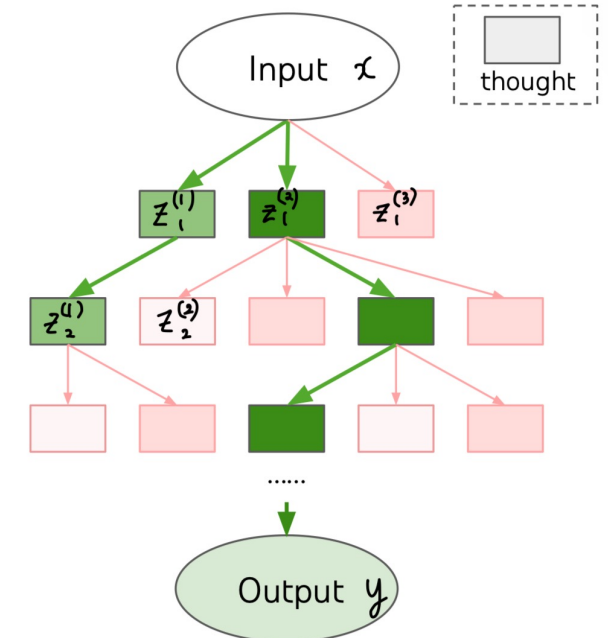
**if**  $V(p_\theta, \{s'\})(s) > v_{thres}$  **then**  $\triangleright$  pruning

        DFS( $s', t + 1$ )

**end if**

**end for**

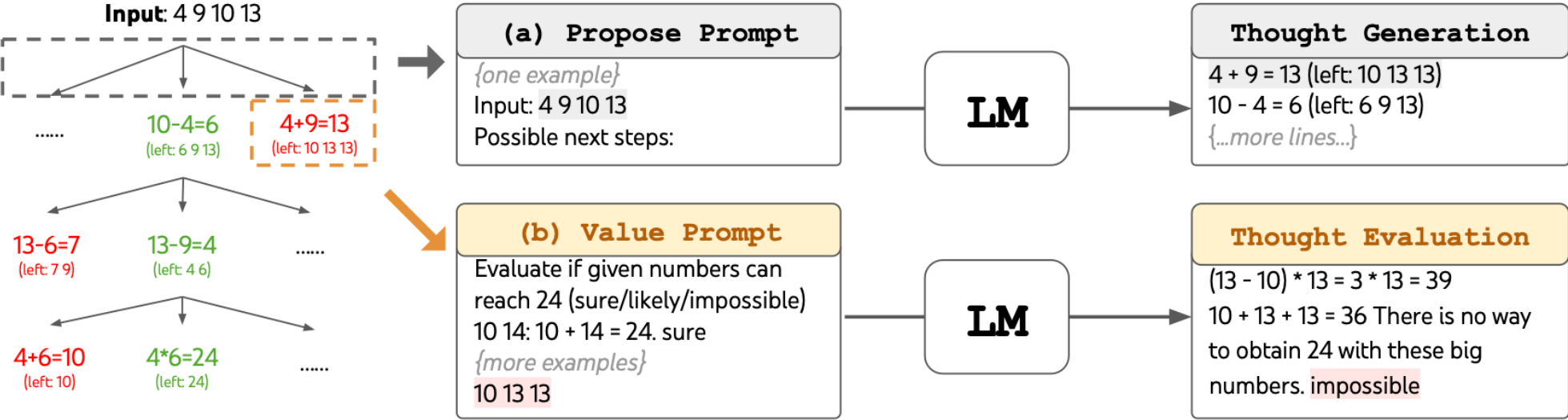
---



(d) Tree of Thoughts (ToT)

# Result and observation: Game of 24

- 4 numbers and basic arithmetic operation to obtain 24



<Task Setup>

- Data: 4nums.com (1,362 games)
- Reported success rate across 100 games

<Baselines>

- IO with 5 in-context examples
- CoT: Augment each IO pair with 3 intermediate equations
- CoT self-consistency: majority output from 100 CoT samples
- IO + refine: at most 10 iterations

<ToT Setup>

- Thought decomposition: line of equation + left
- Thought generator: propose
- State evaluator: Value (sure/maybe/impossible)
- Search algorithm: BFS (b = 5)

# Result and observation: Game of 24

- 4 numbers and basic arithmetic operation to obtain 24 - Result

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

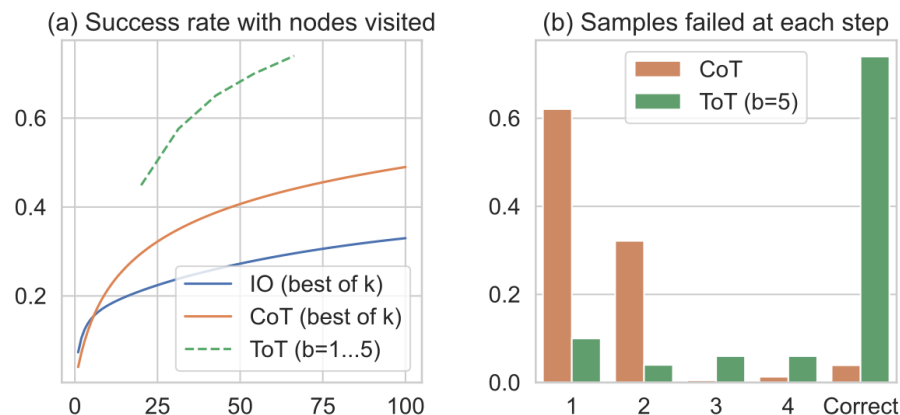


Table 2: Game of 24 Results.

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

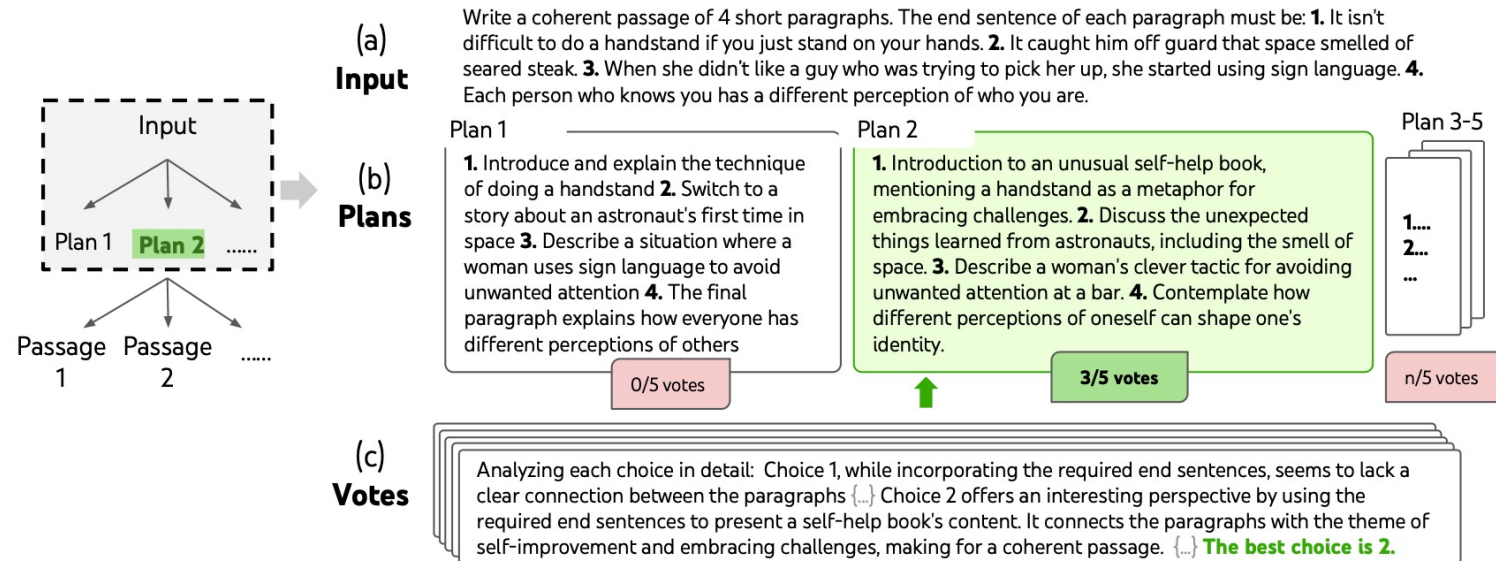
**Results.** As shown in Table 2, IO, CoT, and CoT-SC prompting methods perform badly on the task, achieving only 7.3%, 4.0%, and 9.0% success rates. In contrast, ToT with a breadth of  $b = 1$  already achieves a success rate of 45%, while  $b = 5$  achieves 74%. We also consider an oracle setup for IO/CoT, by calculating the success rate using best of  $k$  samples ( $1 \leq k \leq 100$ ). To compare IO/CoT (best of  $k$ ) with ToT, we consider calculating the tree nodes visited per task in ToT across  $b = 1 \dots 5$ , and map the 5 success rates in Figure 3(a), treating IO/CoT (best of  $k$ ) as visiting  $k$  nodes in a bandit. Not surprisingly, CoT scales better than IO, and best of 100 CoT samples achieve a success rate of 49%, but still much worse than exploring more nodes in ToT ( $b > 1$ ).

**Error analysis.** Figure 3(b) breaks down at which step CoT and ToT samples fail the task, i.e. the thought (in CoT) or all  $b$  thoughts (in ToT) are invalid or impossible to reach 24. Notably, around 60% of CoT samples already failed the task after generating the first step, or equivalently, the first three words (e.g. “4 + 9”). This highlights the issues with direct left-to-right decoding.



# Result and observation: Creative Writing

- Input 4 random sentences and the output should be a coherent passage with 4 paragraphs that end in the 4 input sentences respectively



## <Task Setup>

- Data: randomwordgenerator.com form 100 inputs
- Measure coherency
  - GPT-4 zero-shot 1-10 score
  - Human paired comparison

## <Baselines>

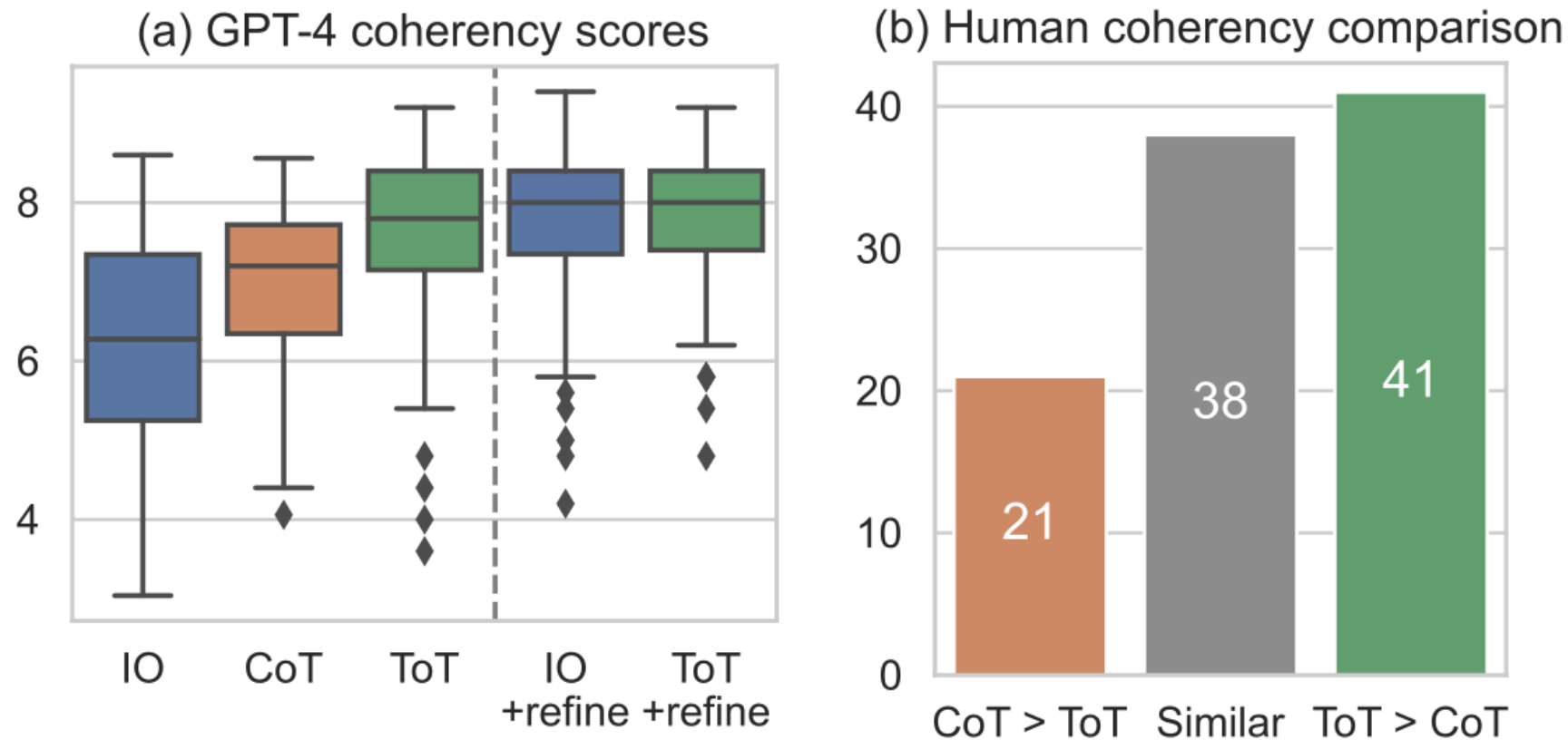
- IO zero-shot (10 samples / task)
- CoT zero-shot but prompts the LM to first make a brief plan then write a passage (10 samples / task)
- Iterative-refine ( $k \leq 5$ ) on top of random IO sample

## <ToT Setup>

- Thought decomposition: passage
- Thought generator: sample
- State evaluator: 5 Votes at both steps
- Search algorithm: BFS ( $b = 1$ )

# Result and observation: Creative Writing

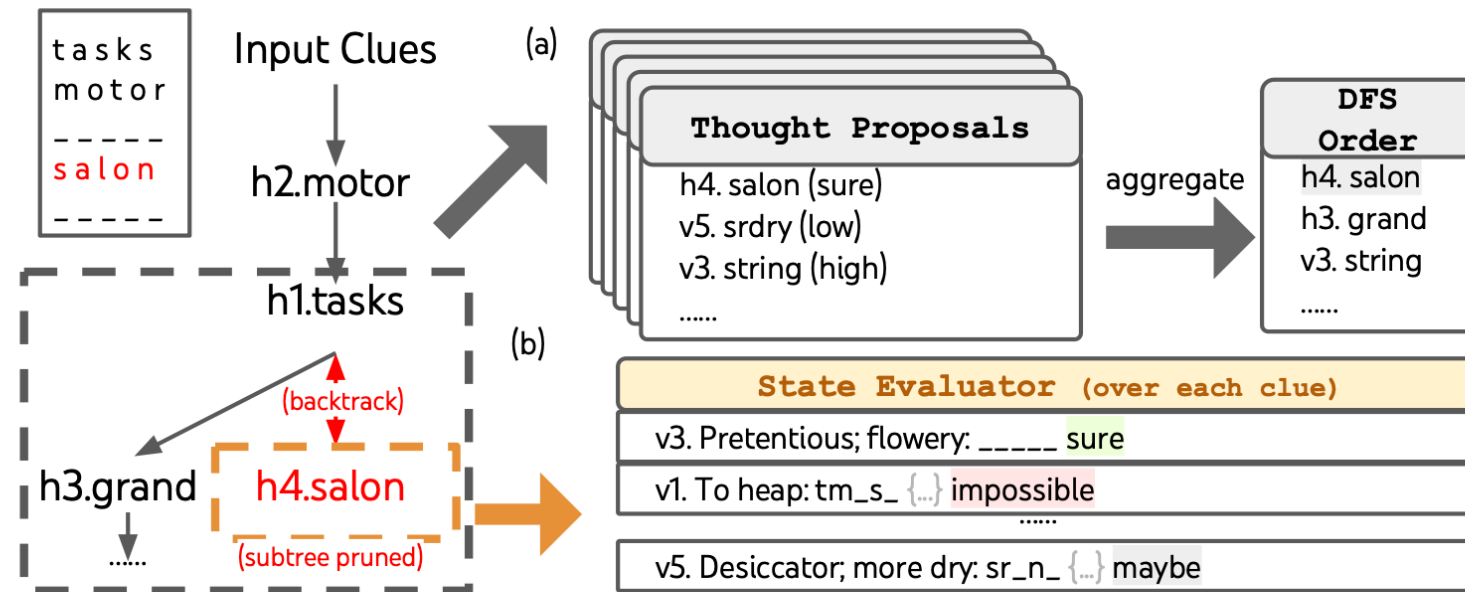
- Input 4 random sentences and the output should be a coherent passage with 4 paragraphs that end in the 4 input sentences respectively - Result



**Figure 5: Creative Writing results.**

# Result and observation: Mini crosswords

- 5X5 mini crosswords



## <Task Setup>

- Data: GooBix (156 games)
- 5 horizontal, 5 vertical cues
- Evaluation: proportion of correct letters (25 per game), words (10 per game), and games (20 games)

## <Baselines>

- IO 5 example pairs
- CoT include intermediate words in the order h1...h5 then v1...5
- Run each prompt for 10 samples and average the results

## <ToT Setup>

- Thought decomposition: words
- Thought generator: propose
- State evaluator: value (possibility of fill given the constraints, sure/low/high)
- Search algorithm: DFS (search step = 100, k - 5)

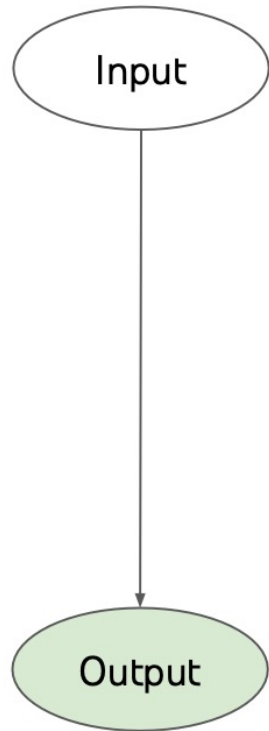
# Result and observation: Creative Writing

- 5X5 mini crosswords - Result

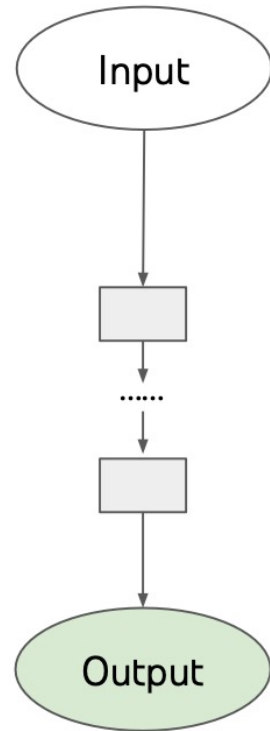
<b>Method</b>	<b>Success Rate (%)</b>		
	<b>Letter</b>	<b>Word</b>	<b>Game</b>
IO	38.7	14	0
CoT	40.6	15.6	1
ToT (ours)	<b>78</b>	<b>60</b>	<b>20</b>
+best state	82.4	67.5	35
-prune	65.4	41.5	5
-backtrack	54.6	20	5

Table 3: Mini Crosswords results.

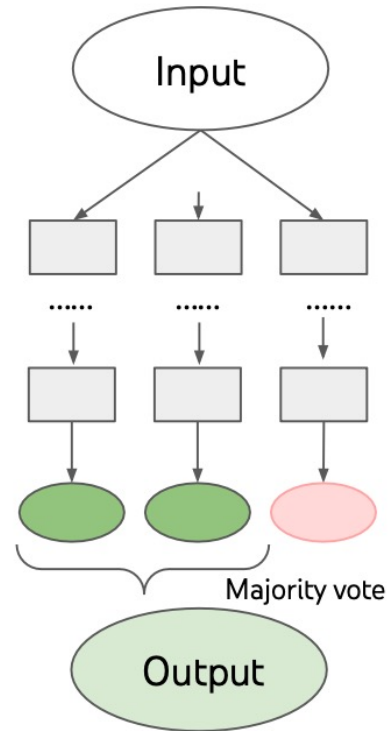
# Takeaways/highlights of the paper



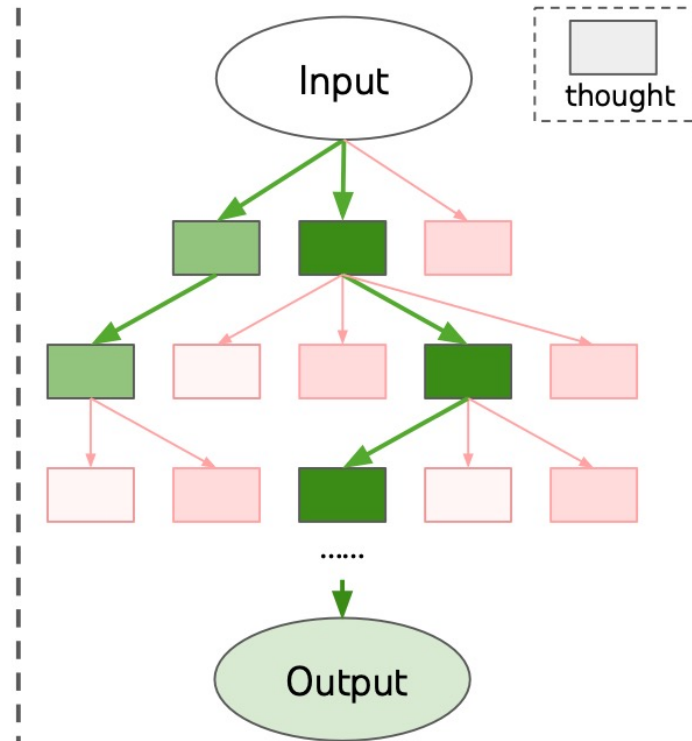
(a) Input-Output Prompting (IQ)



(c) Chain of Thought Prompting (CoT)



(c) Self Consistency with CoT (CoT-SC)



(d) **Tree of Thoughts (ToT)**

- Improve the interpretability
- Task requiring non-trivial planning by considering multiple feasible plans

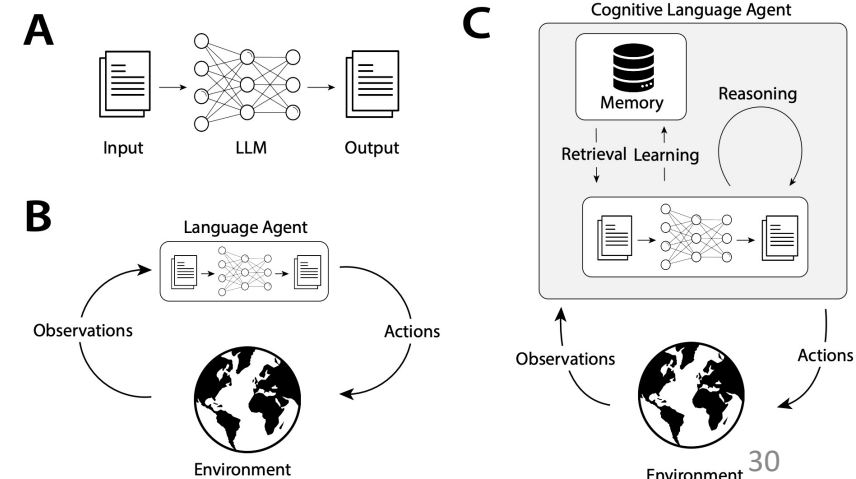
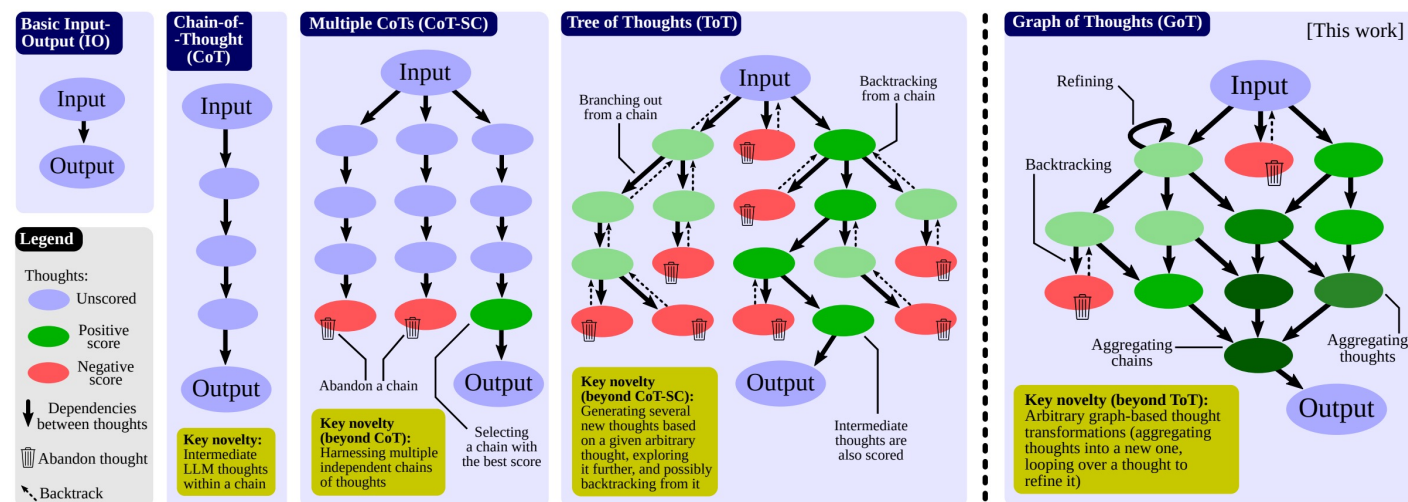
# Limitation / Future work

## Limitations

- GPT-4 already excels at many things
- ToT requires more resources in order to improve task performances
- Fine-tuning LMs using a ToT-style high-level counterfactual design making (e.g., deliberating over potential choices for the next paragraph, instead of predicting the next token) might yield better performance

## Future work

- Adding refining and aggregation
  - [Besta, M., Blach, N., Kubicek, A., Gerstenberger, R., Gianinazzi, L., Gajda, J., ... & Hoefler, T. \(2023\). Graph of thoughts: Solving elaborate problems with large language models. arXiv preprint arXiv:2308.09687.](#)
- Using external memory
  - [Sumers, T. R., Yao, S., Narasimhan, K., & Griffiths, T. L. \(2023\). Cognitive architectures for language agents. arXiv preprint arXiv:2309.02427.](#)



# Discussion Questions

- Are there any other ways to improve complicated problem-solving abilities?

- [Wang, Z., Mao, S., Wu, W., Ge, T., Wei, F., & Ji, H. \(2023\). Unleashing cognitive synergy in large language models: A task-solving agent through multi-persona selfcollaboration. arXiv preprint arXiv:2307.05300, 1\(2\), 3.](#)

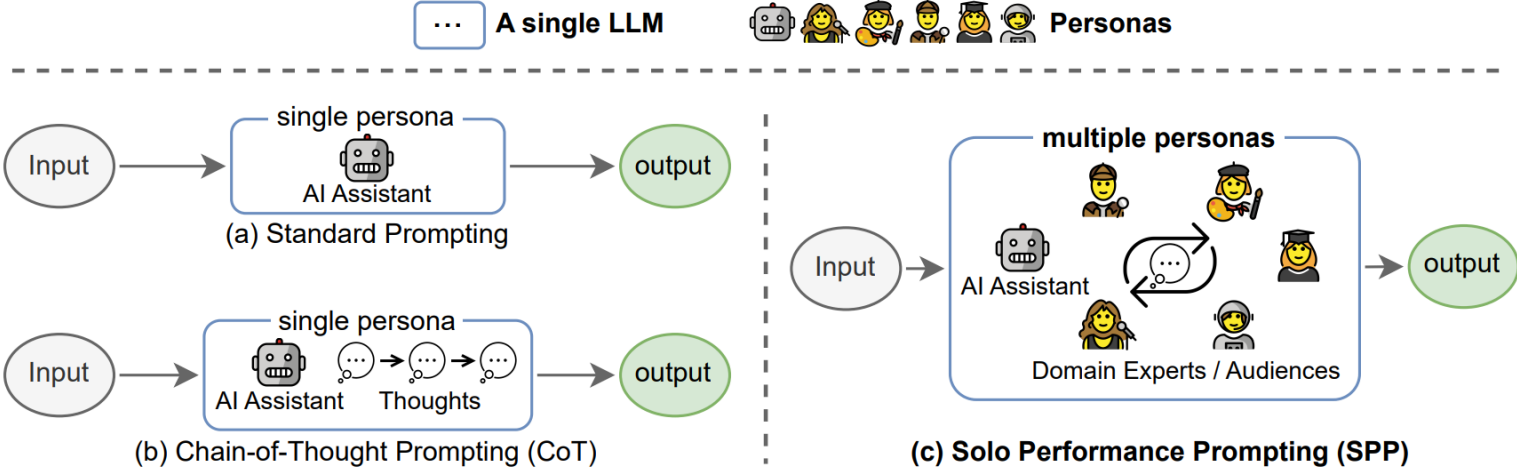


Figure 1: Schematic illustration of Solo Performance Prompting (SPP) and the difference compared to previous prompting methods. SPP transforms a single LLM into a cognitive synergist that dynamically identifies personas and engages in multi-turn self-collaboration to solve various tasks effectively.