VIDY Reading Group
Symbolic reasoning and learning

Shuaicheng Zhang and Longfeng Wu

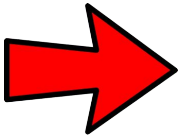
CS, Virginia Tech

Dec 7th, 2023

VLOG

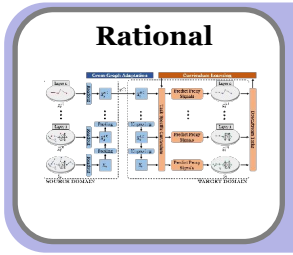
COMPUTER SCIENCE
VIRGINIA TECH

Roadmap



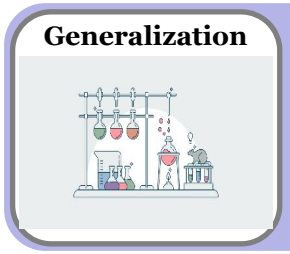
I. Background

- Large Language Models
- Chain-of-thought



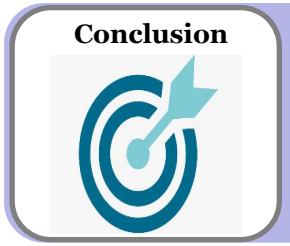
II. Self-consistent Prompting

- Motivations
- Method
- Experiments



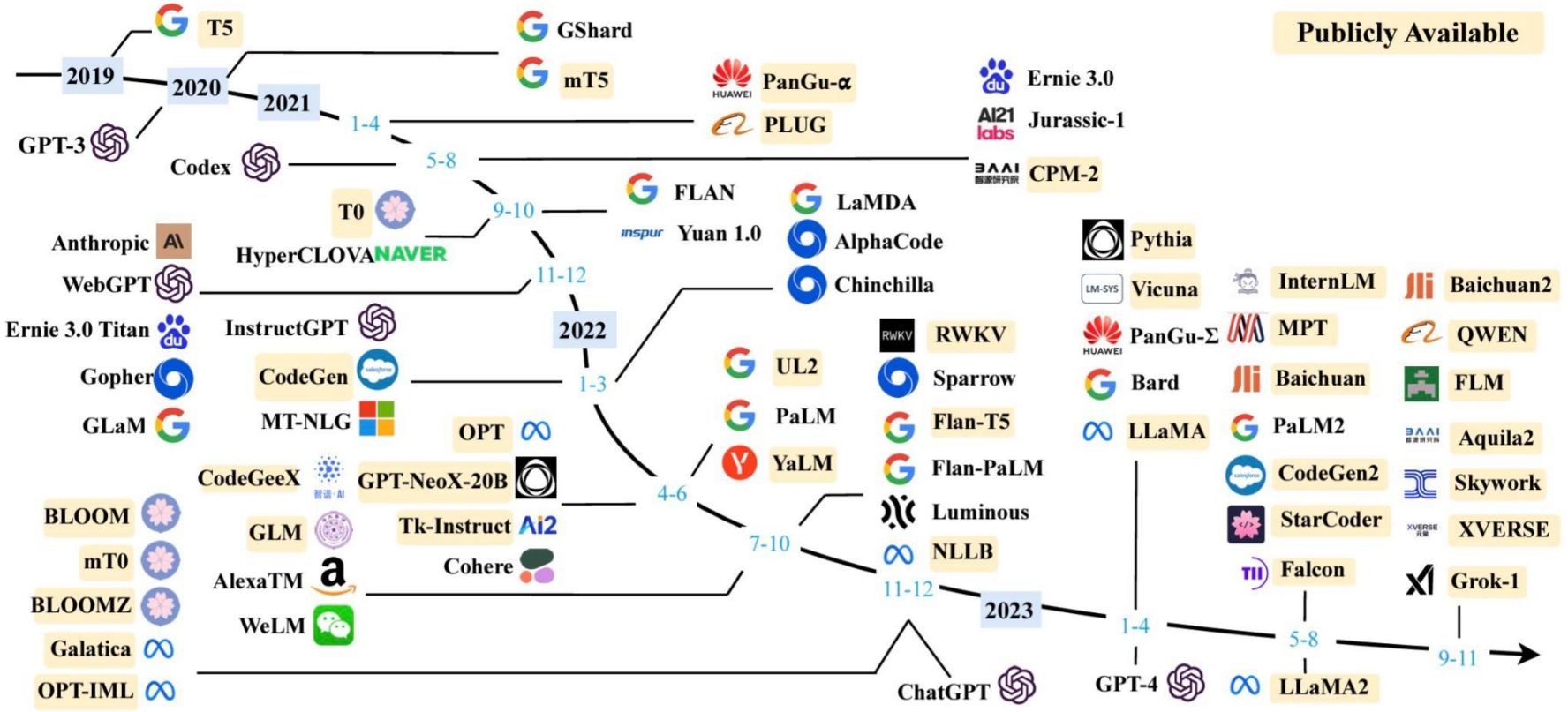
III. Least-to-most Prompting

- Motivations
- Method
- Experiments



IV. Conclusion

Large Language Models



1. Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. ACM Comput. Surv. 55, 9, Article 195 (September 2023), 35 pages. https://doi.org/10.1145/3560815

Large Language Model Basics

- Decoder only model
- LM Objective
  - next token prediction

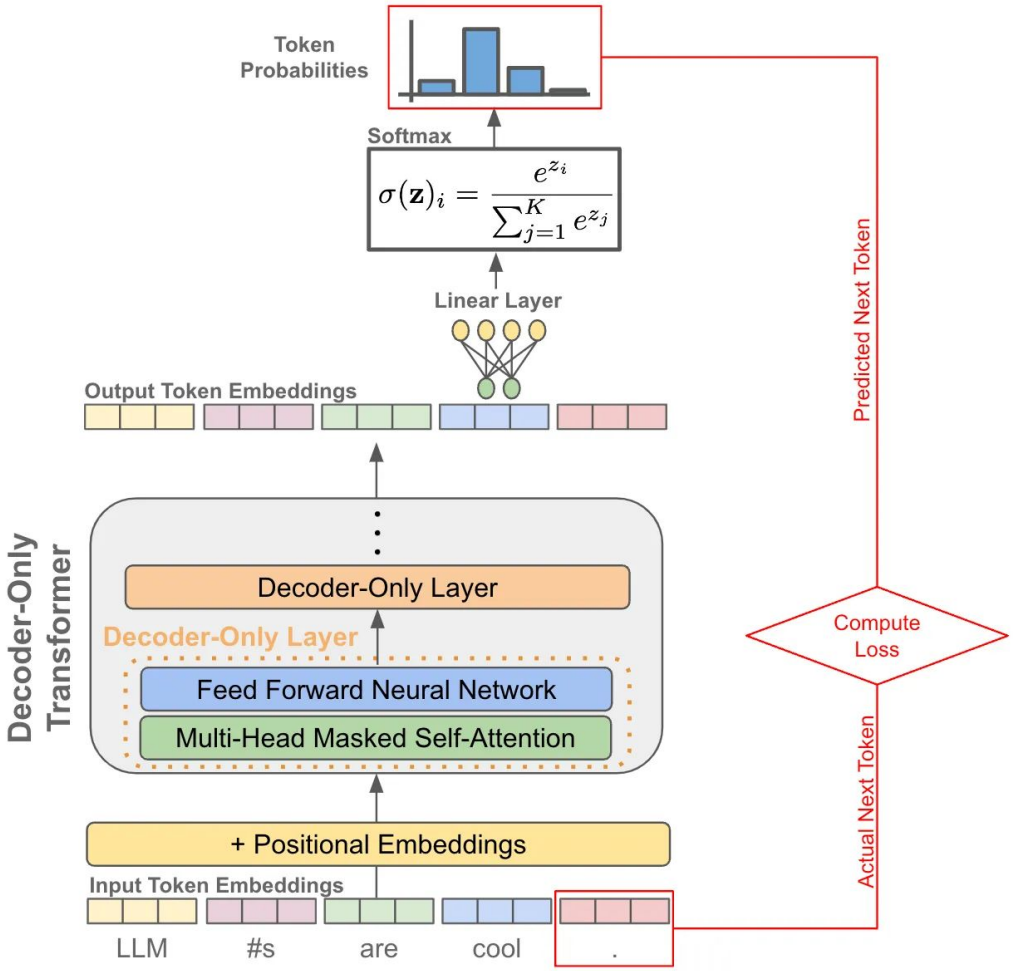
$$\mathcal{L}_{LM}(\mathbf{x}) = \sum_{i=1}^{n} \log P(x_i|\mathbf{x}_{<i}).$$

- Decoding strategy
  - Deterministic

$$x_i = \arg\max_x P(x|\mathbf{x}_{<i}),$$

  - Temperature-based

$$P(x_j|\mathbf{x}_{<i}) = \frac{\exp(l_j/t)}{\sum_{j'} \exp(l_{j'}/t)},$$

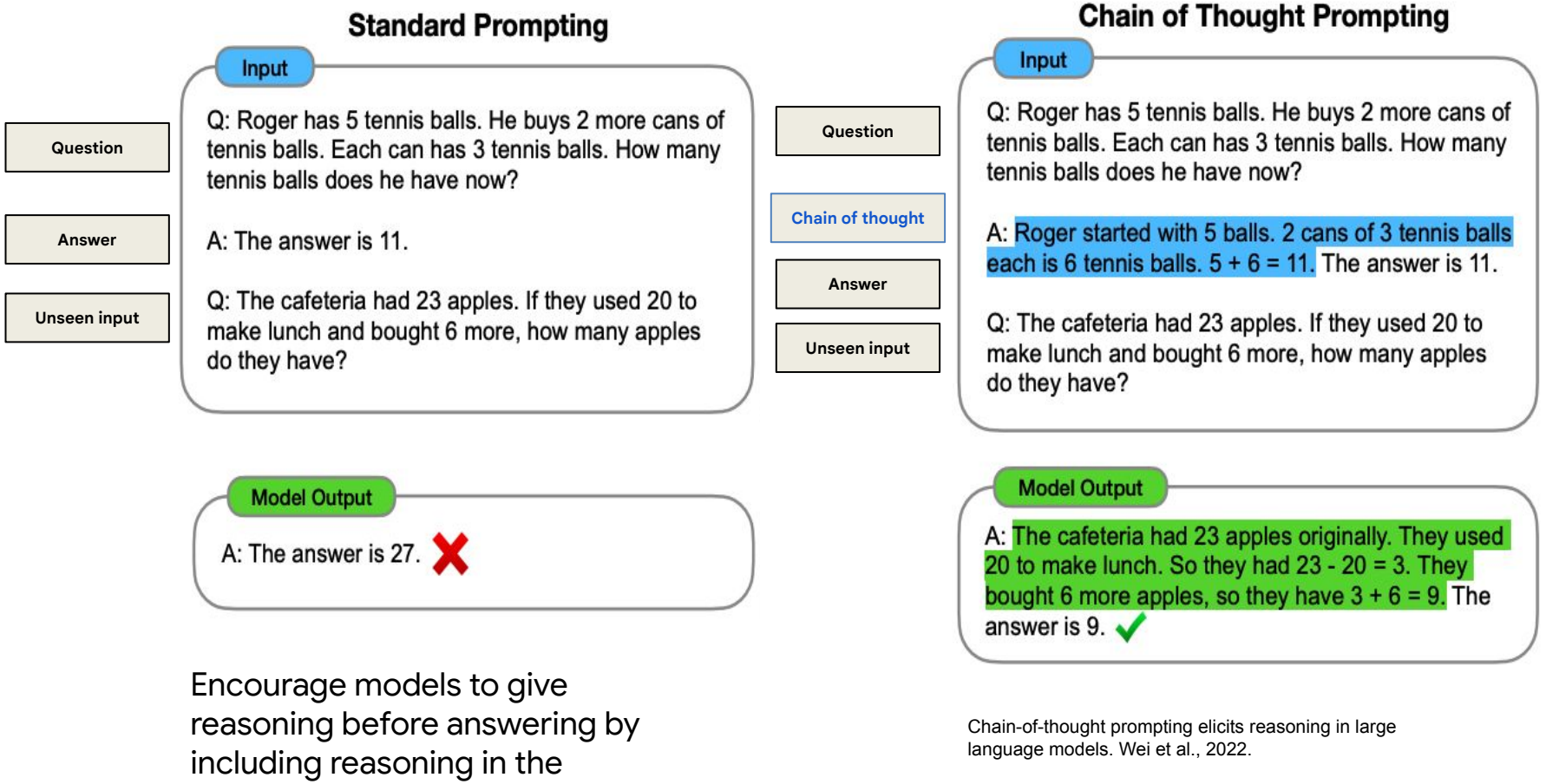


Prompt Basics

Name	Notation	Example	Description
Input	$\boldsymbol{x}$	I love this movie.	One or multiple texts
Output	$\boldsymbol{y}$	++ (very positive)	Output label or text
Prompting Function	$f_{\text{prompt}}(\boldsymbol{x})$	`[X]` Overall, it was a `[Z]` movie.	A function that converts the input into a specific form by inserting the input $\boldsymbol{x}$ and adding a slot `[Z]` where answer $\boldsymbol{z}$ may be filled later.
Prompt	$\boldsymbol{x}'$	I love this movie. Overall, it was a `[Z]` movie.	A text where `[X]` is instantiated by input $\boldsymbol{x}$ but answer slot `[Z]` is not.
Filled Prompt	$f_{\text{fill}}(\boldsymbol{x}', \boldsymbol{z})$	I love this movie. Overall, it was a bad movie.	A prompt where slot `[Z]` is filled with any answer.
Answered Prompt	$f_{\text{fill}}(\boldsymbol{x}', \boldsymbol{z}^*)$	I love this movie. Overall, it was a good movie.	A prompt where slot `[Z]` is filled with a true answer.
Answer	$\boldsymbol{z}$	"good", "fantastic", "boring"	A token, phrase, or sentence that fills `[Z]`

1. Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing. ACM Comput. Surv. 55, 9, Article 195 (September 2023), 35 pages. https://doi.org/10.1145/3560815

Chain of Thought Prompting



Encourage models to give reasoning before answering by including reasoning in the in-context examples.

Chain-of-thought prompting elicits reasoning in large language models. Wei et al., 2022.

Background

- **Motivations**
  - Arithmetic reasoning -> generate natural language rationale.
  - In context few-shot learning via prompt.

- **Limitations of existing works**
  - High cost to create a large set of high quality rationales.
  - Does not improve with the scale of language model[1].

- **Research questions**
  - How large language models can learn via a few examples with natural language data about the task?
  - How much a chain-of-thought prompting can outperforms standard prompting on arithmetic, commonsense, and symbolic reasoning?

[1] Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. "Scaling Language Models: Methods, Analysis & Insights from Training Gopher." arXiv preprint arXiv:2112.11446.

Examples

COT format

<

input,

chain of thought,

output

>

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.

Math Word Problems (multiple choice)

Q: How many keystrokes are needed to type the numbers from 1 to 500?
Answer Choices: (a) 1156 (b) 1392 (c) 1480 (d) 1562 (e) 1788

A: There are 9 one-digit numbers from 1 to 9. There are 90 two-digit numbers from 10 to 99. There are 401 three-digit numbers from 100 to 500. 9 + 90(2) + 401(3) = 1392. The answer is (b).

CSQA (commonsense)

Q: Sammy wanted to go to where the people were. Where might he go?
Options: (a) race track (b) populated areas (c) desert (d) apartment (e) roadblock

A: The answer must be a place with a lot of people. Race tracks, desert, apartments, and roadblocks don't have a lot of people, but populated areas do. So the answer is (b).

StrategyQA

Q: Yes or no: Would a pear sink in water?

A: The density of a pear is about 0.6 g/cm^3, which is less than water. Thus, a pear would float. So the answer is no.

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.

Sports Understanding

Q: Is the following sentence plausible? "Joao Moutinho caught the screen pass in the NFC championship."

A: Joao Moutinho is a soccer player. The NFC championship is part of American football, not soccer. So the answer is no.

SayCan (Instructing a robot)

Human: How would you bring me something that isn't a fruit?

Explanation: the user wants something to eat that isn't a fruit. An energy bar is not a fruit, so I will bring the user an energy bar.
Plan: 1. find(energy bar) 2. pick(energy bar) 3. find(user) 4. put(energy bar) 5. done().

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.

Coin Flip (state tracking)

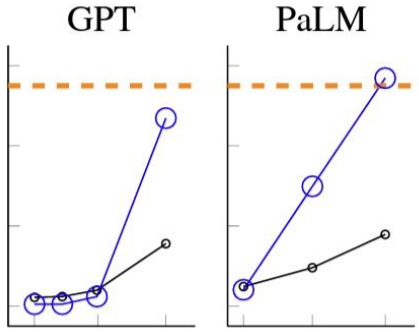
Q: A coin is heads up. Maybelle flips the coin. Shalonda does not flip the coin. Is the coin still heads up?

A: The coin was flipped by Maybelle. So the coin was flipped 1 time, which is an odd number. The coin started heads up, so after an odd number of flips, it will be tails up. So the answer is no.

Observations of CoT

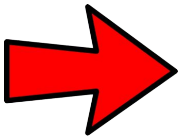
- **Empirical Observations**
  - Chain-of-thoughts prompting **improves performance** by a large margin on arithmetic reasoning.[1]
  - Chain-of-thoughts prompting has a **linguistic nature** for commonsense reasoning.
  - Chain-of-thoughts prompting facilitates **OOD generalization** to longer sequence lengths.

- **Relevance to model scale**
  - Standard prompting has a flat scaling curve but Chain-of-thoughts prompting has a **dramatically increasing** curve.[1]



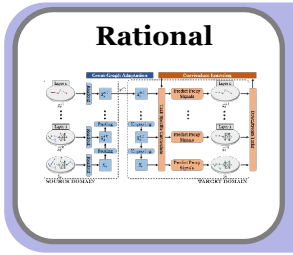
[1] Brown, Tom, et al. "Language models are few-shot learners." *Advances in neural information processing systems* 33 (2020): 1877-1901.

Roadmap



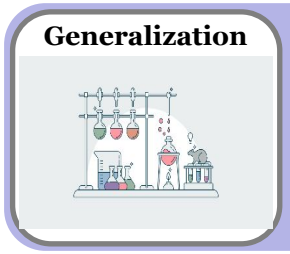
I. Background

- Large Language Models
- Chain-of-thought



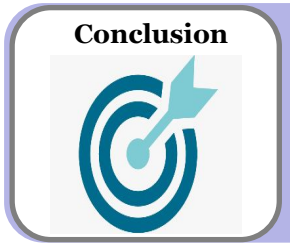
II. Self-consistent Decoding

- Motivations
- Method
- Experiments



III. Least-to-most Prompting

- Motivations
- Method
- Experiments



IV. Conclusion

Self-Consistency Decoding

Self-Consistency Improves Chain of Thought Reasoning in Language Models

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, Denny Zhou

- A **decoding strategy** to replace greedy decoding used in CoT prompting.
- A complex reasoning problem admits multiple different ways of thinking to its **unique** answer. [1]
- Significantly **boosts the performance** of CoT prompting.

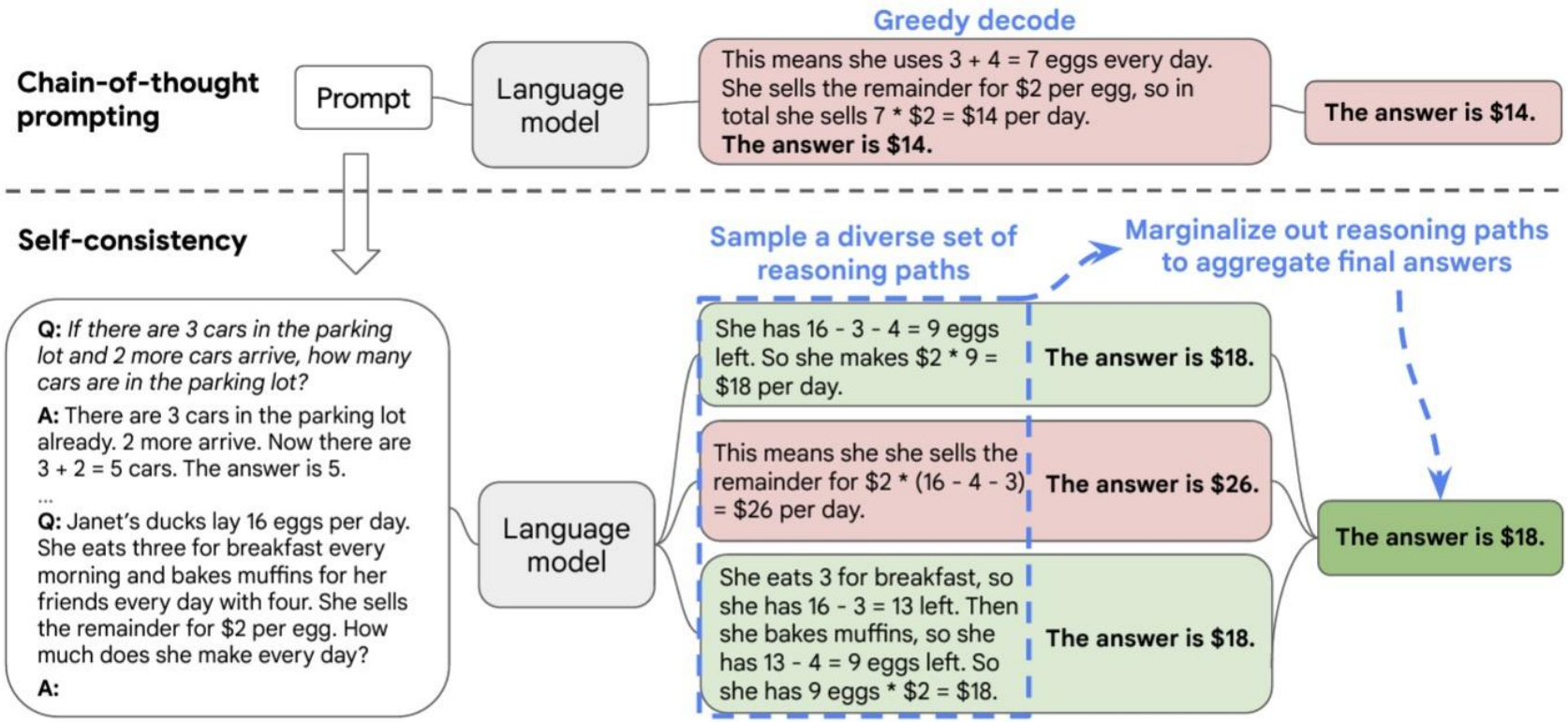
[1] Stanovich, Keith E., and Richard F. West. "24. Individual Differences in Reasoning: Implications for the Rationality Debate?." *Behavioural and Brain Science* 23.5 (2000): 665-726.

Motivations

- **Inconsistency in Reasoning**
  - Traditional CoT reasoning methods often result in inconsistent reasoning paths.
  - Unreliable outputs from Language Models (LMs).

- **Lack of Self-Evaluation Mechanisms**
  - Current LMs do not possess robust mechanisms for self-assessment and correction.
  - Compromise of the reliability of LM reasoning processes.

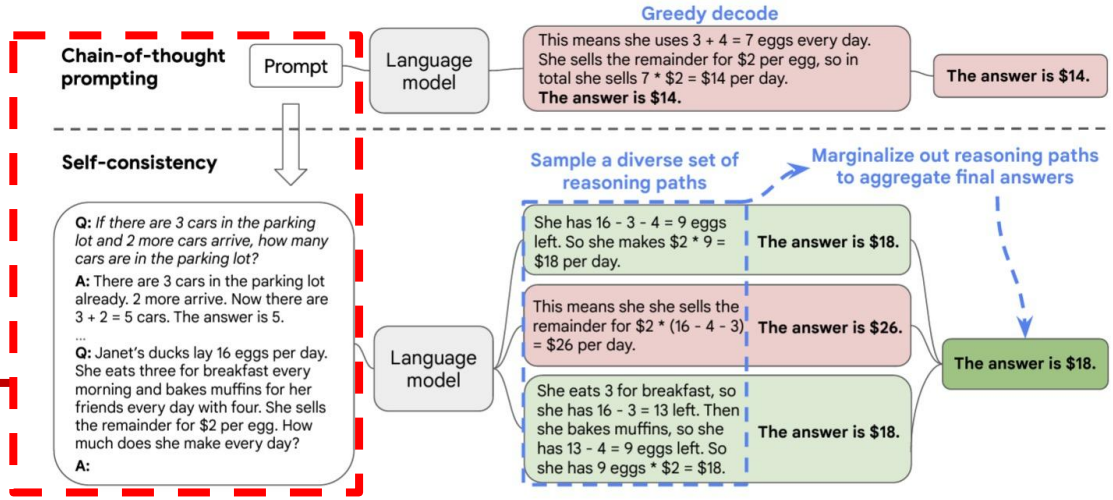
Method Overview



Greedy decode -> Sample a diverset of reasoning paths
-> Marginalize out reasoning paths

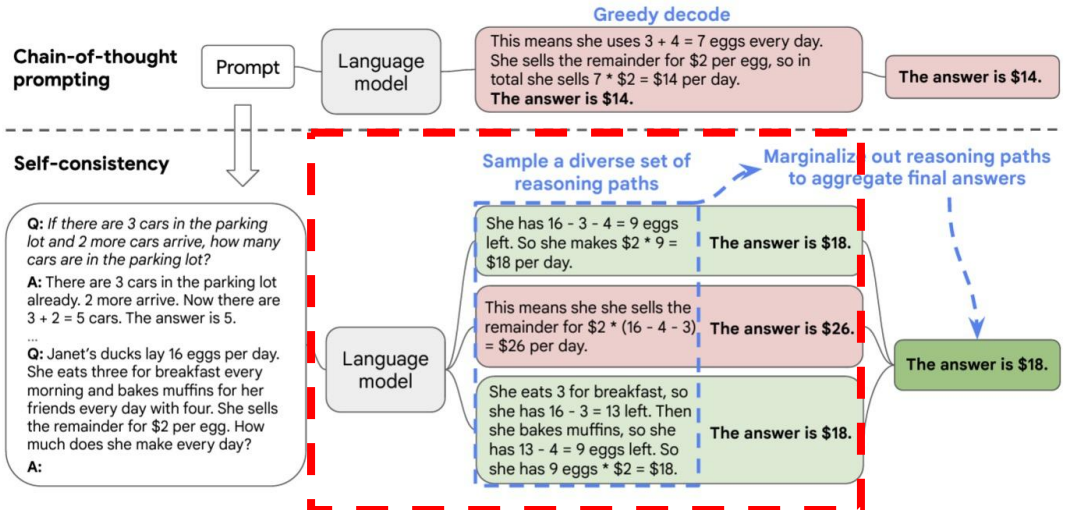
Method Overview

1. Chain-of-thought prompts.



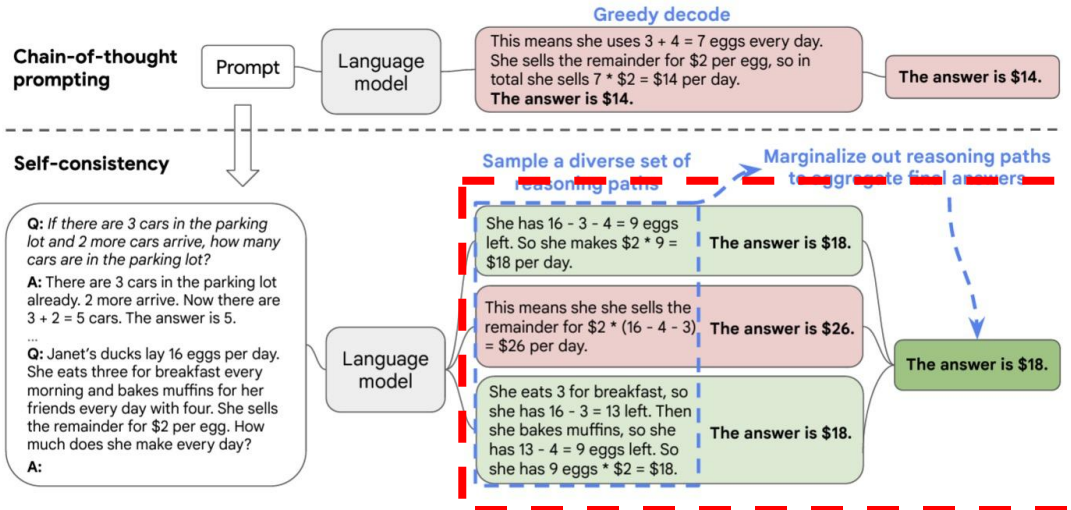
Method Overview

1. Chain-of-thought prompts.
2. Sample a set of candidate reasoning paths from LLM decoder where r is reasoning path and a is answer. $(\mathbf{r}_i, \mathbf{a}_i)$



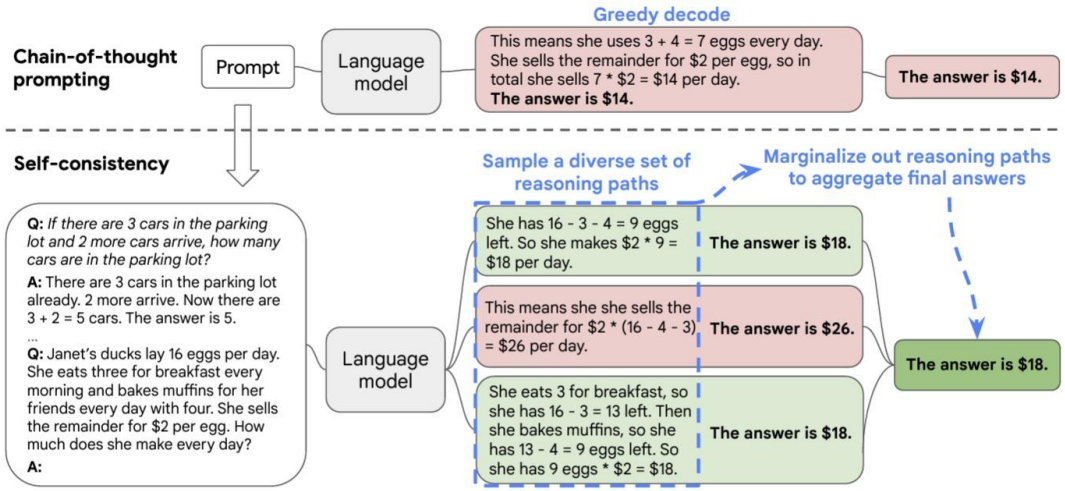
Method Overview

1. Chain-of-thought prompts.
2. Sample a set of candidate reasoning paths from LLM decoder where r is reasoning path and a is answer. $(\mathbf{r}_i, \mathbf{a}_i)$
3. Marginalize r by taking advantage of majority vote of $\arg\max_a \sum_{i=1}^{m} \mathbb{1}(\mathbf{a}_i = a)$
4. Additionally, each $P(\mathbf{r}_i, \mathbf{a}_i \mid \text{prompt}, \text{question})$ can be weighted by $P(\mathbf{r}_i, \mathbf{a}_i \mid \text{prompt}, \text{question}) = \exp \frac{1}{K} \sum_{k=1}^{K} \log P(t_k \mid \text{prompt}, \text{question}, t_1, \ldots, t_{k-1})$



Method Overview

1. Chain-of-thought prompts.
2. Sample a set of candidate reasoning paths from LLM decoder where r is reasoning path and a is answer. $(\mathbf{r}_i, \mathbf{a}_i)$
3. Marginalize r by taking advantage of majority vote of $\arg\max_a \sum_{i=1}^{m} \mathbb{1}(\mathbf{a}_i = a)$
4. Additionally, each $P(\mathbf{r}_i, \mathbf{a}_i \mid \text{prompt}, \text{question})$ can be weighted by $P(\mathbf{r}_i, \mathbf{a}_i \mid \text{prompt}, \text{question}) = \exp \frac{1}{K} \sum_{k=1}^{K} \log P(t_k \mid \text{prompt}, \text{question}, t_1, \ldots, t_{k-1})$,



Tasks and Datasets

Arithmetic Reasoning:

- Math Word Problem Repository: A collection of various arithmetic problems.
- AddSub: Problems involving basic addition and subtraction.
- MultiArith: Tasks requiring multi-step arithmetic reasoning.
- ASDiv: A dataset for arithmetic and symbolic division problems.
- AQUA-RAT: Problems based on grade-school mathematics.
- GSM8K: A set of grade-school math problems.
- SVAMP: A challenge dataset with math word problems.

Commonsense Reasoning:

- CommonsenseQA: A dataset for answering questions that require commonsense knowledge.
- StrategyQA: Questions requiring strategic reasoning skills.
- AI2 Reasoning Challenge (ARC): A challenge for answering science questions that necessitate reasoning.

Symbolic Reasoning:

- Last Letter Concatenation (e.g., “Elon Musk” to “nk”)
- Coinflip (e.g., determining the position of a coin after several flips)

Language Models and Prompts

- **UL2-20b**
  - Encoder-decoder
- **GPT3-175b**
  - Decoder only
- **Lambda-137b**
  - Decoder only.
  - pre-trained on a mixture of web documents, dialog data, and Wikipedia.
- **PaLM-540b**
  - Decoder only.
  - Pretrained on a high-quality corpus of 780 billion tokens, including filtered webpages, books, Wikipedia, news articles, source code, and social media conversations

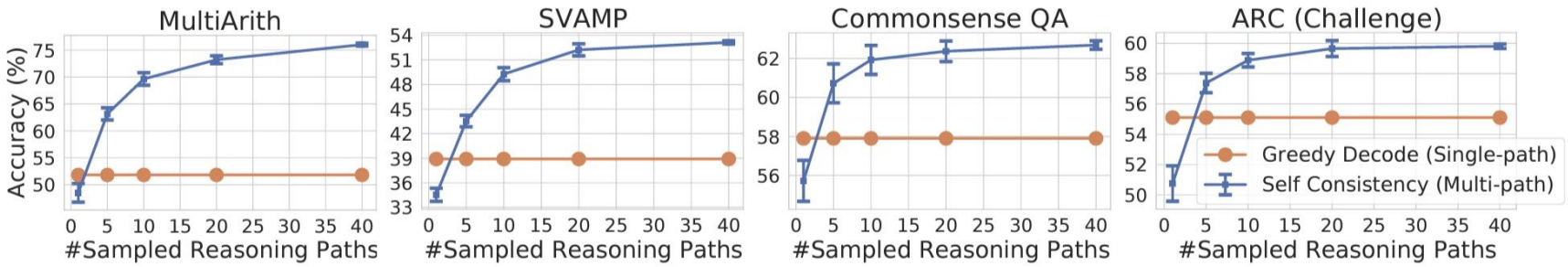
Experiments: Arithmetic Reasoning

	Method	AddSub	MultiArith	ASDiv	AQuA	SVAMP	GSM8K
	Previous SoTA	**94.9**[a]	60.5[a]	75.3[b]	37.9[c]	57.4[d]	35[e] / 55[g]
UL2-20B	CoT-prompting	18.2	10.7	16.9	23.6	12.6	4.1
	Self-consistency	24.8 (+6.6)	15.0 (+4.3)	21.5 (+4.6)	26.9 (+3.3)	19.4 (+6.8)	7.3 (+3.2)
LaMDA-137B	CoT-prompting	52.9	51.8	49.0	17.7	38.9	17.1
	Self-consistency	63.5 (+10.6)	75.7 (+23.9)	58.2 (+9.2)	26.8 (+9.1)	53.3 (+14.4)	27.7 (+10.6)
PaLM-540B	CoT-prompting	91.9	94.7	74.0	35.8	79.0	56.5
	Self-consistency	93.7 (+1.8)	99.3 (+4.6)	81.9 (+7.9)	48.3 (+12.5)	86.6 (+7.6)	74.4 (+17.9)
GPT-3 Code-davinci-001	CoT-prompting	57.2	59.5	52.7	18.9	39.8	14.6
	Self-consistency	67.8 (+10.6)	82.7 (+23.2)	61.9 (+9.2)	25.6 (+6.7)	54.5 (+14.7)	23.4 (+8.8)
GPT-3 Code-davinci-002	CoT-prompting	89.4	96.2	80.1	39.8	75.8	60.1
	Self-consistency	91.6 (+2.2)	**100.0** (+3.8)	**87.8** (+7.6)	**52.0** (+12.2)	**86.8** (+11.0)	**78.0** (+17.9)

Experiments: Commonsense/Symbolic Reasoning

	Method	CSQA	StrategyQA	ARC-e	ARC-c	Letter (4)	Coinflip (4)
	Previous SoTA	**91.2**[a]	73.9[b]	86.4[c]	75.0[c]	N/A	N/A
UL2-20B	CoT-prompting	51.4	53.3	61.6	42.9	0.0	50.4
	Self-consistency	55.7 (+4.3)	54.9 (+1.6)	69.8 (+8.2)	49.5 (+6.8)	0.0 (+0.0)	50.5 (+0.1)
LaMDA-137B	CoT-prompting	57.9	65.4	75.3	55.1	8.2	72.4
	Self-consistency	63.1 (+5.2)	67.8 (+2.4)	79.3 (+4.0)	59.8 (+4.7)	8.2 (+0.0)	73.5 (+1.1)
PaLM-540B	CoT-prompting	79.0	75.3	95.3	85.2	65.8	88.2
	Self-consistency	80.7 (+1.7)	**81.6** (+6.3)	**96.4** (+1.1)	**88.7** (+3.5)	70.8 (+5.0)	91.2 (+3.0)
GPT-3 Code-davinci-001	CoT-prompting	46.6	56.7	63.1	43.1	7.8	71.4
	Self-consistency	54.9 (+8.3)	61.7 (+5.0)	72.1 (+9.0)	53.7 (+10.6)	10.0 (+2.2)	75.9 (+4.5)
GPT-3 Code-davinci-002	CoT-prompting	79.0	73.4	94.0	83.6	70.4	99.0
	Self-consistency	81.5 (+2.5)	79.8 (+6.4)	96.0 (+2.0)	87.5 (+3.9)	**73.4** (+3.0)	**99.5** (+0.5)

Experiments: Accuracy vs # of Paths



	GSM8K	StrategyQA
Question	*Henry made two stops during his 60-mile bike trip. He first stopped after 20 miles. His second stop was 15 miles before the end of the trip. How many miles did he travel between his first and second stops?*	*Yes or no: Is Albany, Georgia the most populous US Albany?*
Greedy Decode	He traveled 60 miles in total. He stopped after 20 miles, so he traveled 60 - 20 = 40 miles between the first and second stops. The answer is 40.	Albany, Georgia is the most populous US Albany. Thus, the answer is yes.
Sampled Path 1	Henry travelled a total of 60 miles. His first stop was 20 miles in, and his second stop was 15 miles before the end. So between his first and second stops he travelled 60 - 20 -15 = 25 miles. The answer is 25.	The most populous US Albany is Albany, New York. Thus, Albany, Georgia is not the most populous US Albany. So the answer is no.
Sampled Path 2	He made two stops during a 60-mile trip. The first was 20 miles into the trip. The second was 15 miles before the end of the trip. This means the second stop was 60 - 15 = 45 miles into the trip. Since he made the stops in order, the second stop must have been 45 - 20 = 25 miles after the first stop. The answer is 25.	Albany, Georgia has a population of about 88,000. Albany, New York has a population of about 95,000. Thus, Albany, Georgia is not the most populous US Albany. So the answer is no.

Experiments: When CoT doesn't work

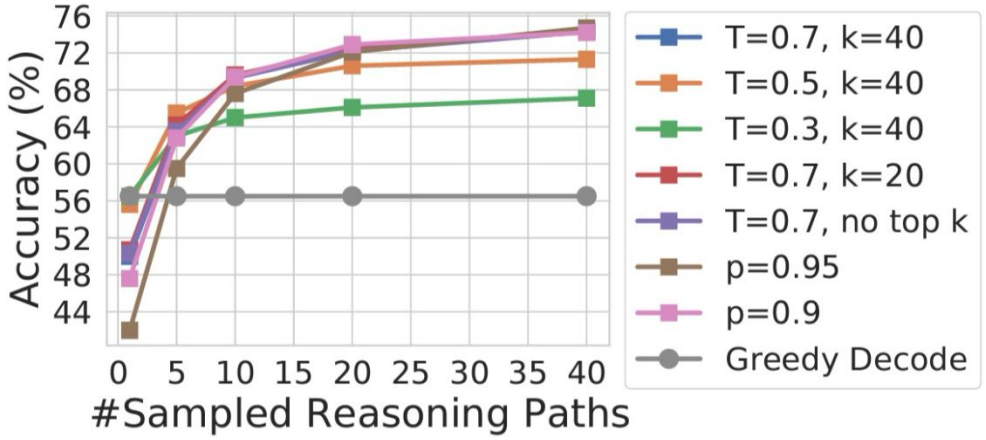
	ANLI R1 / R2 / R3	e-SNLI	RTE	BoolQ	HotpotQA (EM/F1)
Standard-prompting (no-rationale)	69.1 / 55.8 / 55.8	85.8	84.8	71.3	27.1 / 36.8
CoT-prompting (Wei et al., 2022)	68.8 / 58.9 / 60.6	81.0	79.1	74.2	28.9 / 39.8
Self-consistency	**78.5 / 64.5 / 63.4**	**88.4**	**86.3**	**78.4**	**33.8 / 44.6**

Experiments: Beam Search & Ensemble

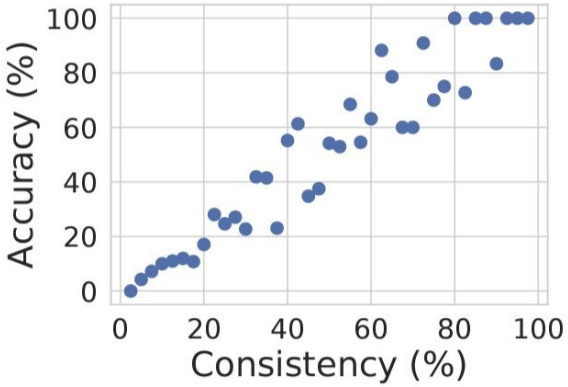
	Beam size / Self-consistency paths	1	5	10	20	40
AQuA	Beam search decoding (top beam)	23.6	19.3	16.1	15.0	10.2
	Self-consistency using beam search	23.6	19.8 ± 0.3	21.2 ± 0.7	24.6 ± 0.4	24.2 ± 0.5
	Self-consistency using sampling	19.7 ± 2.5	**24.9 ± 2.6**	**25.3 ± 1.8**	**26.7 ± 1.0**	**26.9 ± 0.5**
MultiArith	Beam search decoding (top beam)	10.7	12.0	11.3	11.0	10.5
	Self-consistency using beam search	10.7	11.8 ± 0.0	11.4 ± 0.1	12.3 ± 0.1	10.8 ± 0.1
	Self-consistency using sampling	9.5 ± 1.2	11.3 ± 1.2	**12.3 ± 0.8**	**13.7 ± 0.9**	**14.7 ± 0.3**

	GSM8K	MultiArith	SVAMP	ARC-e	ARC-c
CoT (Wei et al., 2022)	17.1	51.8	38.9	75.3	55.1
Ensemble (3 sets of prompts)	18.6 ± 0.5	57.1 ± 0.7	42.1 ± 0.6	76.6 ± 0.1	57.0 ± 0.2
Ensemble (40 prompt permutations)	19.2 ± 0.1	60.9 ± 0.2	42.7 ± 0.1	76.9 ± 0.1	57.0 ± 0.1
Self-Consistency (40 sampled paths)	**27.7 ± 0.2**	**75.7 ± 0.3**	**53.3 ± 0.2**	**79.3 ± 0.3**	**59.8 ± 0.2**

Experiments: Scaling & Consistency



LaMDA-137B	Prompt with correct chain-of-thought	17.1
	Prompt with imperfect chain-of-thought	14.9
	+ Self-consistency (40 paths)	**23.4**
	Prompt with equations	5.0
	+ Self-consistency (40 paths)	**6.5**
PaLM-540B	Zero-shot CoT (Kojima et al., 2022)	43.0
	+ Self-consistency (40 paths)	**69.2**



Key takeaways

- CoT's greedy decoding sometimes fall short to lead the correct output.
  - Suffers from repetitiveness and local-optimality.

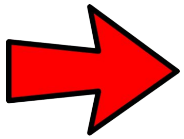
- Decoding a batch of candidate reasoning path with majority votes help the answer be more consistent.
  - Correct reasoning path may lead to the wrong answer, vice versa.
  - Diversity is the essence to recover the answer[1].

- Limitations
  - Computation cost to generate extra candidate reasoning paths.

- Future works
  - self-consistency can be used to generate better supervised data to fine-tune the model.

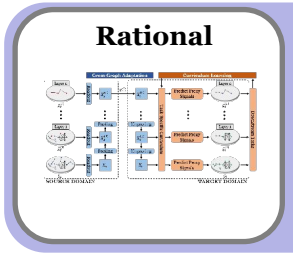
[1] Evans, Jonathan St BT. "Intuition and reasoning: A dual-process perspective." *Psychological Inquiry* 21.4 (2010): 313-326.

Roadmap



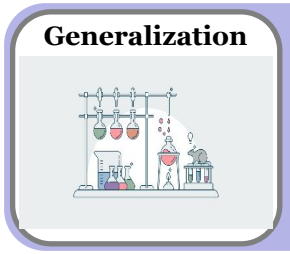
I. Background

- Large Language Models
- Chain-of-thought



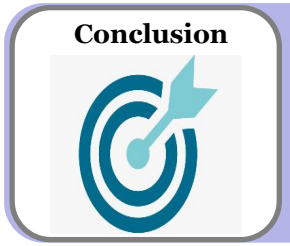
II. Self-consistent Prompting

- Motivations
- Method
- Experiments



III. Least-to-most Prompting

- Motivations
- Method
- Experiments



IV. Conclusion

Motivations

- **Differences between human and machine learning**

  - **Human**: Accomplish a new task from only a few examples.
  - **Machine Learning**: Requires a large amount of labeled data.

  - **Human**: Explain the underlying rationale for their decisions.
  - **Machine Learning**: A black box essentially.

  - **Human**: Solve problems more difficult than they have seen.
  - **Machine Learning**: Train and test at the same level of difficulty.

Motivations

- **Differences between human and machine learning**

Chain-of-thought

  - **Human**: Accomplish a new task from only a few examples.
  - **Machine Learning**: Requires a large amount of labeled data.

Self-consistency

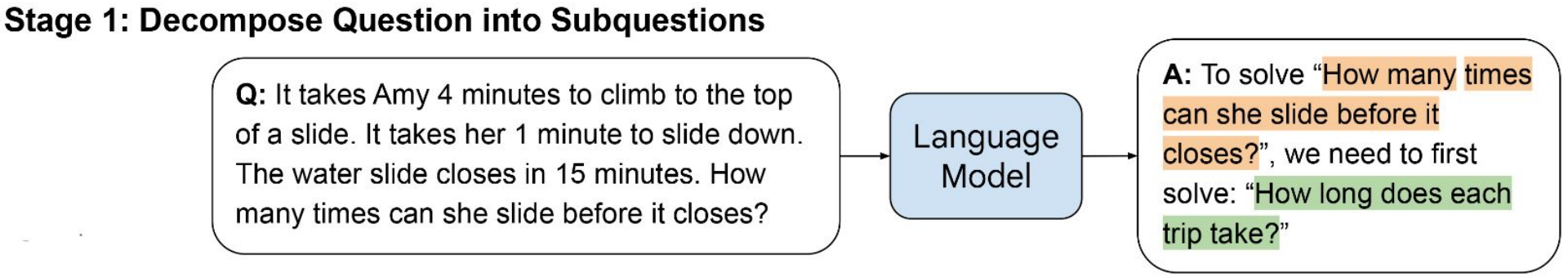
  - **Human**: Explain the underlying rationale for their decisions.
  - **Machine Learning**: A black box essentially.

Least-to-most

  - **Human**: Solve problems more difficult than they have seen.
  - **Machine Learning**: Train and test at the same level of difficulty.

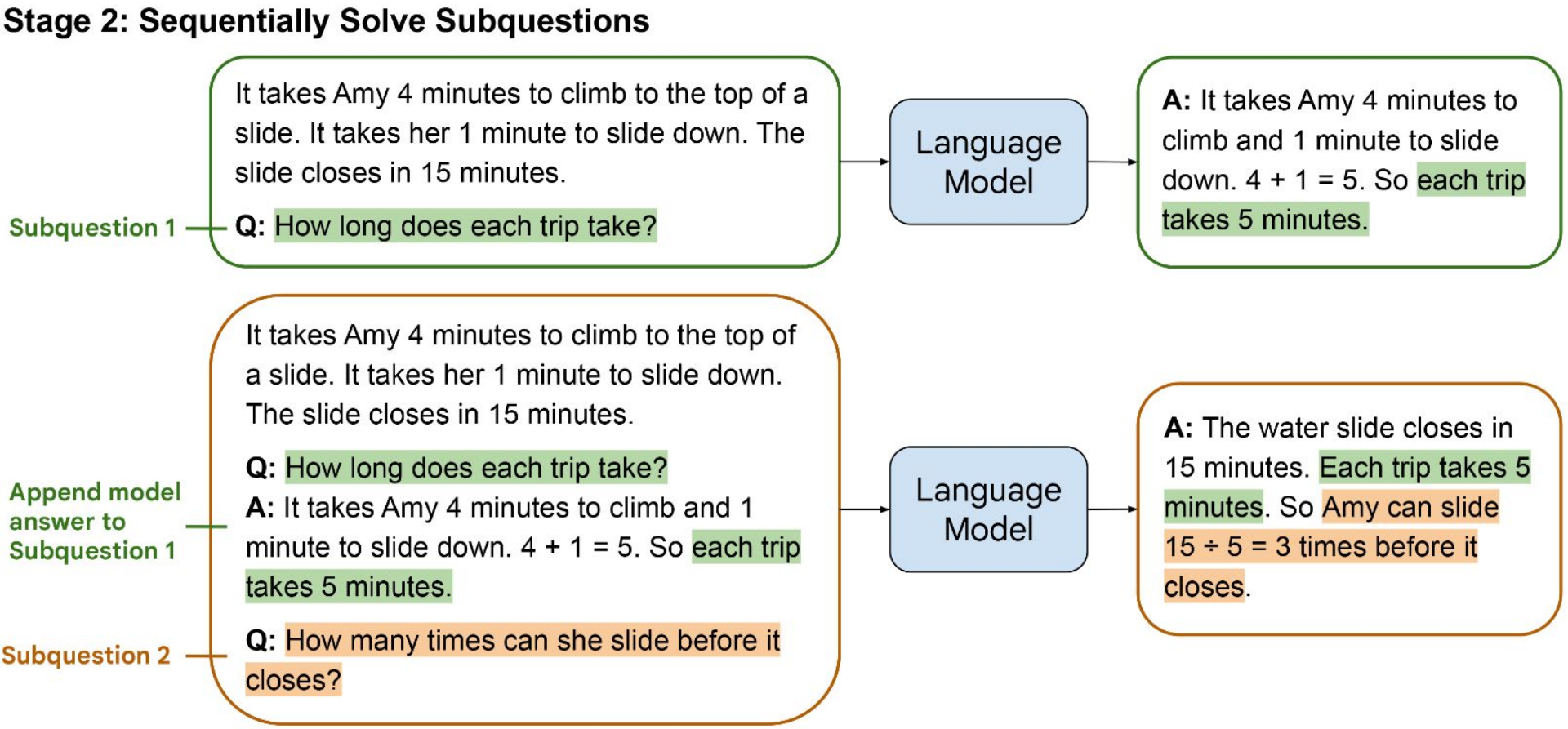
Method Overview

- Main ideas:
  - Decomposition.
    - Query the LLM to decompose a complex problem into a list of easier subproblems (with few examples).



Method Overview

- Main ideas:
  - Subproblem solving.
    - Constant examples + a potentially empty list of previously subproblem and generated solutions + next question.



Experiment Results

- Symbolic manipulation
  - Last-letter-concatenation (e.g., “Elon Musk” to “nk”)

- Prompting techniques:
  - Standard few-shot prompting

Q: “think, machine”
A: “ke”.

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

  - Chain-of-thought prompting

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: 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”.

Experiment Results

– Least-to-most prompting

Stage 1

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

Stage 2

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”.

Experiment Results

- Model
  - GPT-3 (code-davinci-002)
- Testing set
  - Random sample words from Wiktinoary with length from 4-12.
- Results

	$L = 4$	$L = 6$	$L = 8$	$L = 10$	$L = 12$
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**

- Observations
  - Standard prompting completely fails on this task.
  - Performance of Cot drops much faster than least-to-most as the length increases.

Experiment Results

- Compositional generalization
  - 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

- Prompting techniques
  - Least-to-most prompting

Stage 1

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".

Experiment Results

– Chain-of-thought prompting

Stage 2

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".

Q: "run and look twice"
A: The output of "run and look twice" concatenates: the output of "run", the output of "look twice". "run" outputs "RUN". "look twice" outputs "LOOK" * 2. So concatenating the output of "run" and the output of "look twice" leads to "RUN" + "LOOK" * 2. So the output of "run and look twice" is "RUN" + "LOOK" * 2.

- Results

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**

- Observations
  - Least-to-most prompting achieves an accuracy of 99.7% for various splits.

Experiment Results

- Math reasoning
  - Datasets: GSM8K, DROP
  - Difficulty: number of solving steps.
- Prompting techniques
  - Least-to-most prompting

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.

  - Chain-of-thought prompting

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.

Experiment Results

- Results

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**

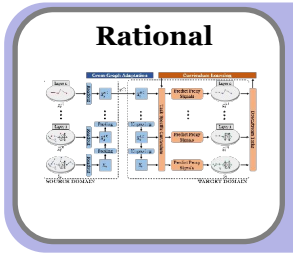
Accuracy by Steps (GSM8K)	All	2 Steps	3 Steps	4 steps	$\geq$ 5 steps
Least-to-Most	**62.39**	74.53	**68.91**	**59.73**	**45.23**
Chain-of-Thought	60.87	**76.68**	67.29	59.39	39.07

- Observations
  - Least-to-most outperforms Cot by a large margin for DROP, and slightly improve Cot for GSM8K.
  - Least-to-most improves Cot in solving problems which need at least 5 steps to be solved.

Roadmap

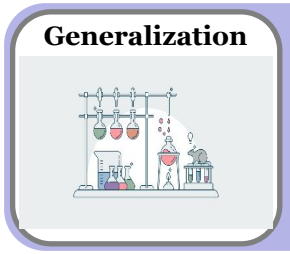
I. Background

- Large Language Models
- Chain-of-thought



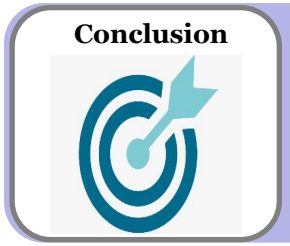
II. Self-consistent Prompting

- Motivations
- Method
- Experiments



III. Least-to-most Prompting

- Motivations
- Method
- Experiments



IV. Conclusion

Conclusion

Prompt Engineering	Pros	Cons
Chain-of-thought	• Significantly improves performance. • Linguistic features for reasoning. • Generalize to OOD samples. • Scales drastically with model size.	• Greedy decoding may suffer from repetitiveness and local-optimality.
Self-consistency	• Diverse reasoning paths lead to a more confident answer. • Empirical analysis shows significant better performance.	• More computation cost due to diversity.
Least-to-most	• Tackle easy-to-hard generalization issues. • Combine with other techniques (Cot, SC) • Unidirectional communication → bidirectional conversations	• Highly dependent on decomposition. • Decomposition don't generalize well across different domains.


THANK
YOU!


Q&A