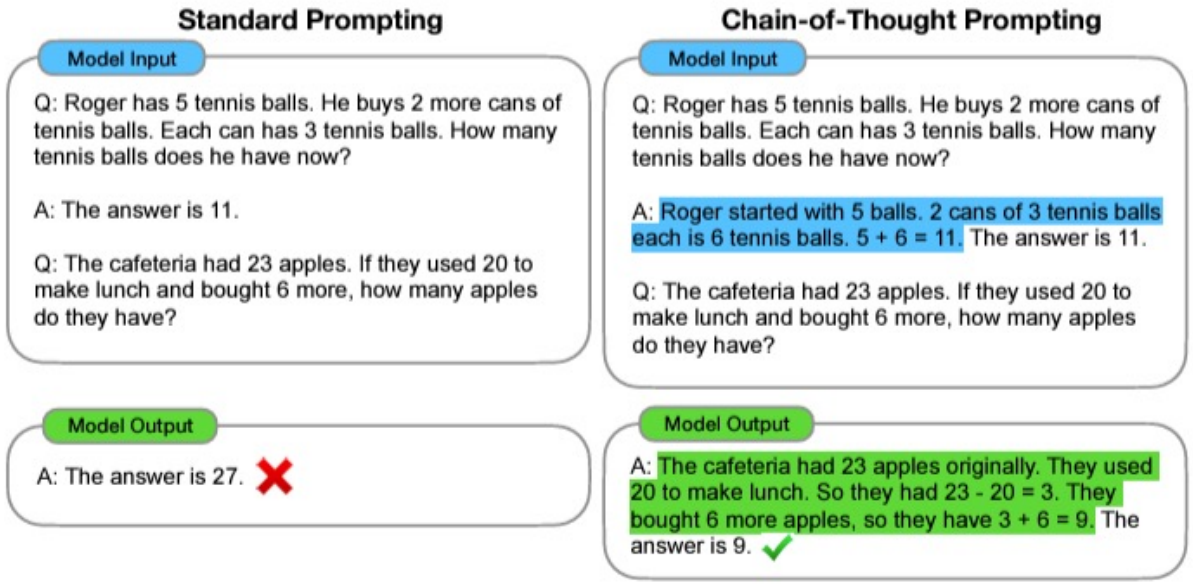
Language model reasoning II

CSE 561

Anxu (Ben) Wang

Rationales can improve LLM performance




```
Input:
6 2 4 + 2 5 9
Target:
<scratch>
6 2 4 + 2 5 9 , C: 0
2 + 5 , 3  C: 1
6 + 2 , 8 3  C: 0
, 8 8 3  C: 0
0 8 8 3
</scratch>
8 8 3
```

Current issues with rationale generation

- Construction of a fine-tuning dataset of rationales
  - Manually by human annotators
    - Expensive
    - Infeasible to construct for each problem
  - Automatically with hand-crafted templates
    - Only work when a general solution is already known
- Leverage in-context learning by including only a few rationale examples in the language model prompt
  - Underperform models fin-tuned to directly predict answers using large datasets

Theme: Leverage LLM output to reduce human input

- STaR: Self-Taught Reasoner Bootstrapping Reasoning with Reasoning

- Large Language Models can Self-Improve

Self-Taught Reasoner (STaR)

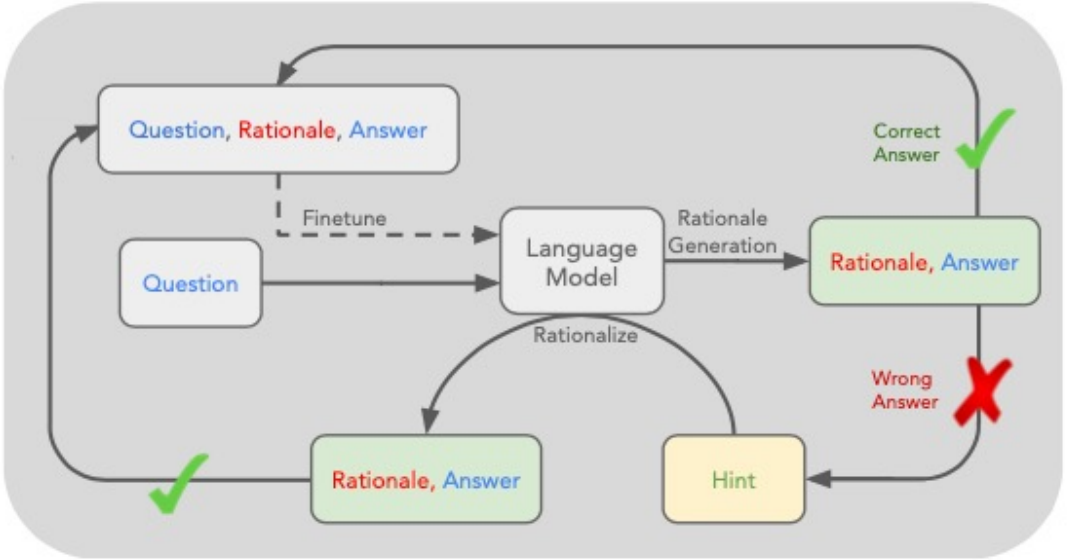


Figure 1: An overview of STaR and a STaR-generated rationale on CommonsenseQA. We indicate the fine-tuning outer loop with a dashed line. The questions and ground truth answers are expected to be present in the dataset, while the rationales are generated using STaR.

Two steps for the method

- Rationale Generation Bootstrapping
- Rationalization

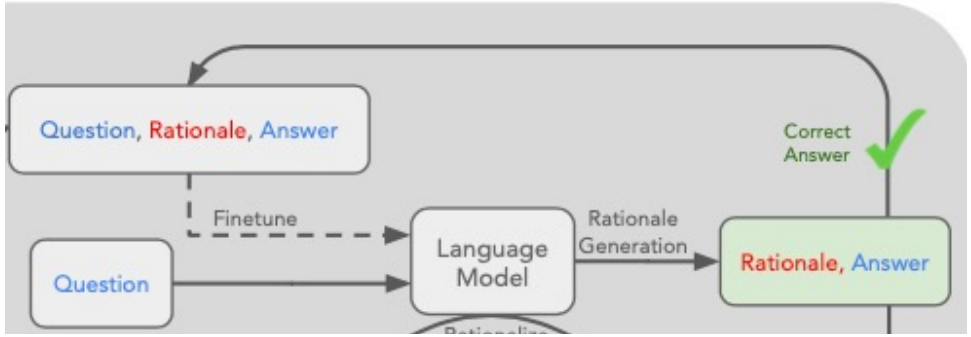
Algorithm 1 STaR

Input M: a pretrained LLM; dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{D}$ (w/ few-shot prompts)

$M_0 \leftarrow M$ # Copy the original model
for n **in** $1...N$ **do** # Outer loop
  $(\hat{r}_i, \hat{y}_i) \leftarrow M_{n-1}(x_i) \quad \forall i \in [1, D]$ # Perform rationale generation
  $(\hat{r}_i^{\text{rat}}, \hat{y}_i^{\text{rat}}) \leftarrow M_{n-1}(\text{add\_hint}(x_i, y_i)) \quad \forall i \in [1, D]$ # Perform rationalization
  $\mathcal{D}_n \leftarrow \{(x_i, \hat{r}_i, y_i) \mid i \in [1, D] \wedge \hat{y}_i = y_i\}$ # Filter rationales using ground truth answers
  $\mathcal{D}_n^{\text{rat}} \leftarrow \{(x_i, \hat{r}_i^{\text{rat}}, y_i) \mid i \in [1, D] \wedge \hat{y}_i \neq y_i \wedge \hat{y}_i^{\text{rat}} = y_i\}$ # Filter rationalized rationales
  $M_n \leftarrow \text{train}(M, \mathcal{D}_n \cup \mathcal{D}_n^{\text{rat}})$ # Finetune the original model on correct solutions - inner loop
end for

Two steps for the method

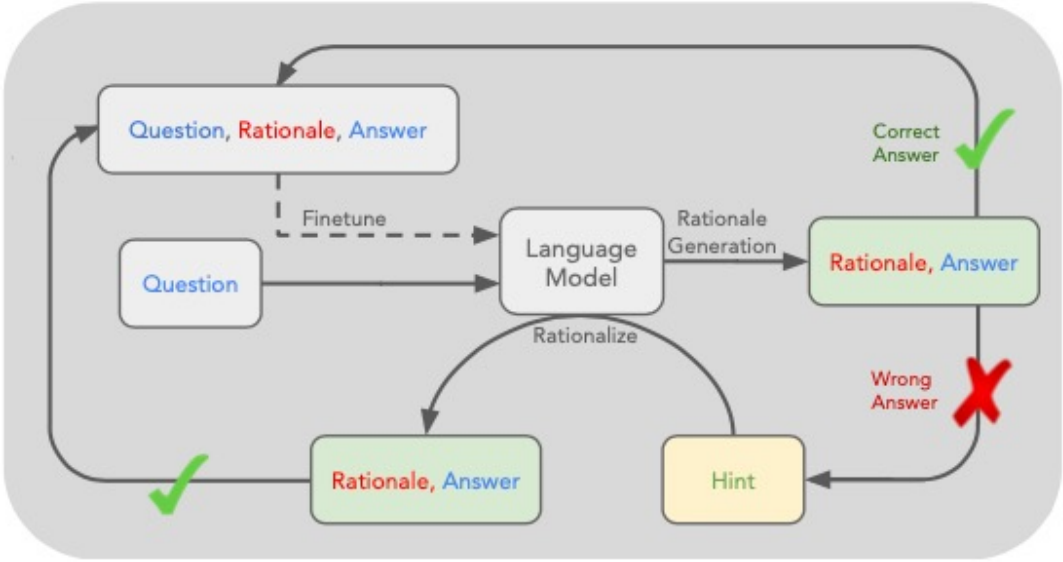
- Rationale Generation Bootstrapping
  - Pretrained LLM *M*, dataset *D* = {(xi, yi)}$_{i=1}^{D}$
  - P = {(x_i^p ,r_i^p y_i^p)}$_{i=1}^{P}$, where P ≪ D (e.g. P = 10)
  - Produce rationale r_i for each x_i
  - Filter the generated rationales to include only the ones which result in correct answer
  - Fine-tune M on filtered rationales
- Rationalization



Two steps for the method

- Rationale Generation Bootstrapping
- Rationalization
  - Improvement from bootstrap plateaus

```
Q: Where do you put your grapes just
before checking out?
Answer Choices:
(a) mouth
(b) grocery cart (CORRECT)
(c) super market
(d) fruit basket
(e) fruit market
A: The answer should be the place
where grocery items are placed before
 checking out. Of the above choices,
grocery cart makes the most sense for
 holding grocery items. Therefore,
the answer is grocery cart (b).
```



Two steps for the method

- Rationale Generation Bootstrapping
- Rationalization

Algorithm 1 STaR

Input M: a pretrained LLM; dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{D}$ (w/ few-shot prompts)

$M_0 \leftarrow M$ # Copy the original model
for n **in** $1...N$ **do** # Outer loop
  $(\hat{r}_i, \hat{y}_i) \leftarrow M_{n-1}(x_i) \quad \forall i \in [1, D]$ # Perform rationale generation
  $(\hat{r}_i^{\text{rat}}, \hat{y}_i^{\text{rat}}) \leftarrow M_{n-1}(\text{add\_hint}(x_i, y_i)) \quad \forall i \in [1, D]$ # Perform rationalization
  $\mathcal{D}_n \leftarrow \{(x_i, \hat{r}_i, y_i) \mid i \in [1, D] \wedge \hat{y}_i = y_i\}$ # Filter rationales using ground truth answers
  $\mathcal{D}_n^{\text{rat}} \leftarrow \{(x_i, \hat{r}_i^{\text{rat}}, y_i) \mid i \in [1, D] \wedge \hat{y}_i \neq y_i \wedge \hat{y}_i^{\text{rat}} = y_i\}$ # Filter rationalized rationales
  $M_n \leftarrow \text{train}(M, \mathcal{D}_n \cup \mathcal{D}_n^{\text{rat}})$ # Finetune the original model on correct solutions - inner loop
end for

Experiments Set Up and Datasets

- GPT-J (6B-parameter model)
  - Large enough to generate rationales of non-trivial quality to be bootstrapped from
- Arithmetic
- CommonsenseQA
- Grade School Math (GSM8K)

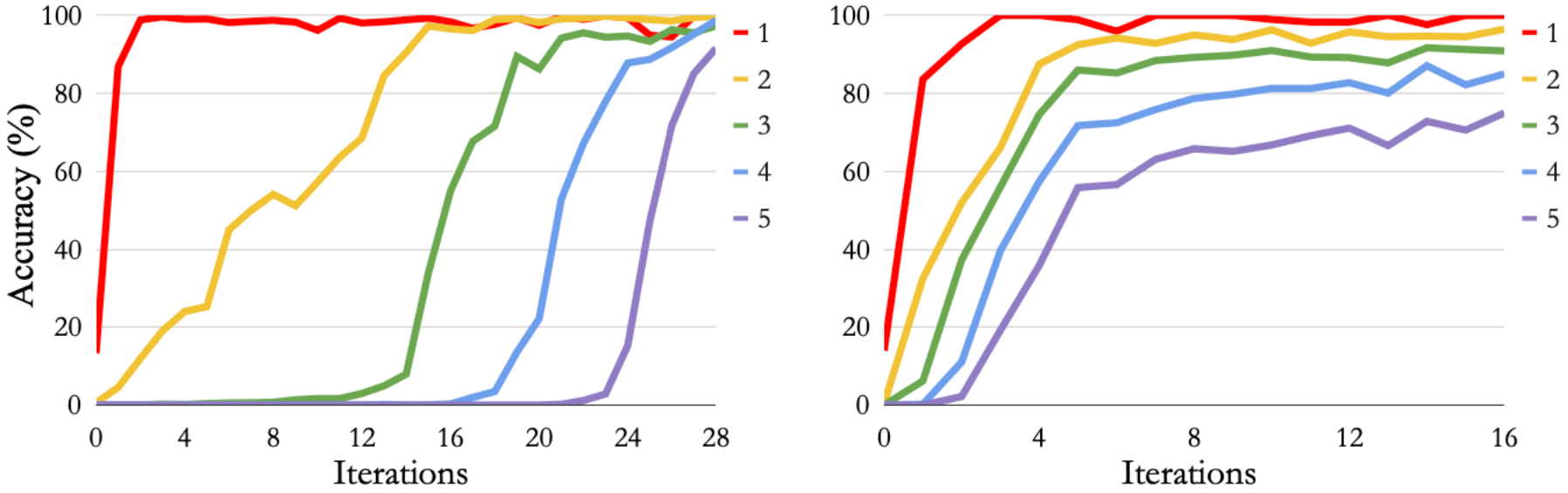
Experiments Set Up and Datasets

- GPT-J (6B-parameter model)
- Arithmetic
  - Calculate the sum of two n-digit integers
  - Everything up to and including “Target” in prompt
  - Asked to generate the scratchpad
  - Include few-shot prompts for 1-5 digits
- CommonsenseQA
- Grade School Math (GSM8K)

```
Input:
6 2 4 + 2 5 9
Target:
<scratch>
6 2 4 + 2 5 9 , C: 0
2 + 5 , 3  C: 1
6 + 2 , 8 3  C: 0
, 8 8 3  C: 0
0 8 8 3
</scratch>
8 8 3
```

Results: Symbolic Reasoning

- Without iterations, few-shot accuracy on arithmetic problems is less than 1% even with rationales



(a) Without rationalization

(b) With rationalization

Figure 4: A visualization of the accuracy of n-digit summation with each iteration of STaR with and without rationalization for arithmetic. Each series corresponds to the accuracy of summing two n-digit numbers.

Experiments Set Up and Datasets

- CommonsenseQA
  - Diverse set of questions which require commonsense reasoning ability
  - Multiple choice questions with 20% by chance

```
Q: What home entertainment equipment requires cable?
Answer Choices:
(a) radio shack
(b) substation
(c) television
(d) cabinet
(e) desk
A: The answer must require cable. Cable is used to provide satellite channels to
televisions. Therefore, the answer is television (c).

Q: Why do I eat just enough when eating breakfast?
Answer Choices:
(a) gas
(b) full stomach
(c) feeling satisfied (CORRECT)
(d) have energy
(e) bloated
A: The answer must be a reason for eating just enough when eating breakfast. Eating
just enough is a way to feel satisfied. Therefore, the answer is feeling satisfied (
c).
```

Results: Natural Language Reasoning

- STaR without rationalization outperformed GPT-J fine-tuned despite less training data
- Inclusion of rationalization performed close to 30x larger GPT-3

	CQA Dev Set Accuracy (%)	Train Data Used (%)
GPT-3 Direct Finetuned [29]	*73.0*	100
Few-shot Direct GPT-J	20.9	∼0
Few-shot CoT GPT-J [3]	36.6	∼0
Few-shot CoT LaMDA 137B [6]	55.6	∼0
GPT-J Direct Finetuned	60.0	100
STaR without rationalization	68.8	69.7
STaR with rationalization	**72.5**	86.7

Results: Natural Language Reasoning

- STaR improve CQA rationale quality
  - Crowdworkers rank STaR-generated rationales higher than the few-shot rationales

Q: They prided themselves on being a wealth of knowledge, and that's why many chose to attend their what?
Answer Choices:
(a) book store (b) university (c) meeting (d) class (e) encyclopedia
~~A: The answer must be a place where people go to learn about things. The answer is university (b).~~
A: The answer must be a place where people go to learn about things. Universities are places where people go to learn about things. Therefore, the answer is university (b).

Results: Natural Language Reasoning

- STaR improve CQA rationale quality
  - Crowdworkers rank STaR-generated rationales higher than the few-shot rationales
- Failure cases
  - Standard logical fallacies

```
Q: What might someone get from learning about science?
Answer Choices:
(a) headache
(b) see things differently
(c) increased knowledge
(d) accidents
(e) appreciation of nature
A: The answer must be something that someone would get from learning about
science. Learning about science would increase knowledge. Therefore, the
answer is increased knowledge (c).
```

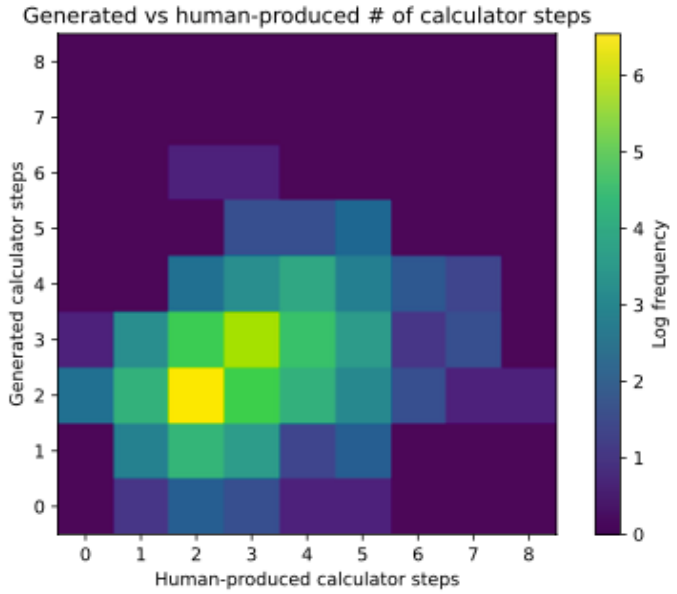
These rationales, while perhaps useful to the model, read to us as opaque and unexplanatory.

Experiments Set Up and Datasets

- Mathematical reasoning in language
  - Grade School Math (GSM8L) dataset
  - Posed in natural language and require two to eight calculation steps
  - Combines the skills needed for arithmetic and commonsense reasoning

Results: Mathematical Reasoning in Language

	GSM8K Test Accuracy (%)	Train Data Used (%)
Few-shot Direct GPT-J	3.0	~0
Few-shot CoT GPT-J	3.1	~0
GPT-J Direct Finetuned	5.8	100
STaR without rationalization	10.1	25.0
STaR with rationalization	**10.7**	28.7



Summary

- Proposed a bootstrapping mechanism to iteratively generate a rationale dataset from a few initial examples with rationales—without needing to check new rationales' correctness
- Complemented rationale generation with rationalization, where a model is tasked with justifying an answer and then fine-tuned as if it had come up with the rationale without any hint. It is shown rationalization accelerates and improves the bootstrapping process

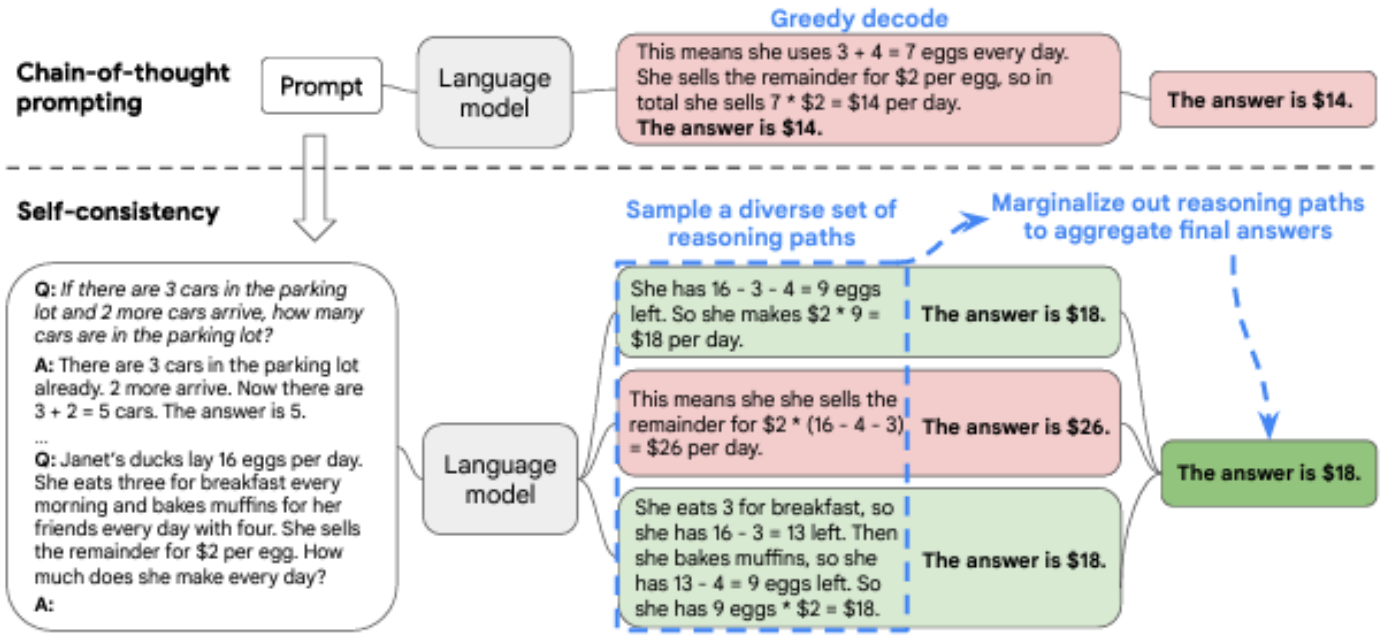
Theme: Leverage LLM output to reduce human input

- STaR: Self-Taught Reasoner Bootstrapping Reasoning with Reasoning

- Large Language Models can Self-Improve
  - This work is similar to Zelikman et al. (2022) where they both propose to fine-tune a model on self-generated CoT data, but their method does not require ground truth labels and shows stronger empirical results with multi-task generalization.

Theme: Leverage LLM output to reduce human input

- Large Language Models can Self-Improve



Large language model can improve itself

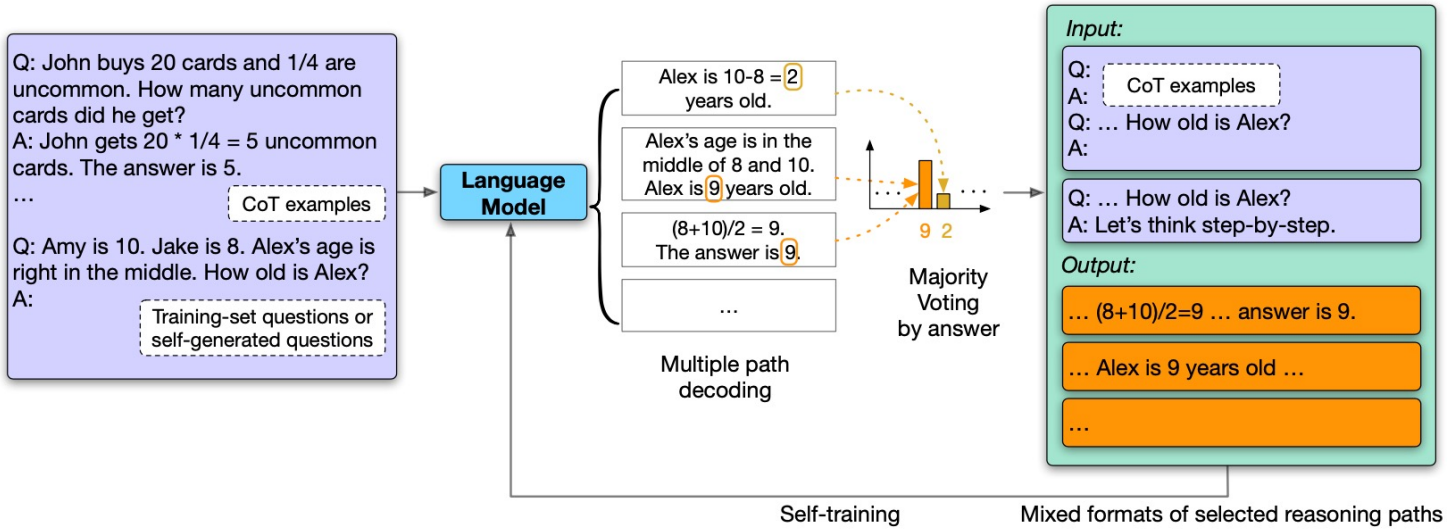
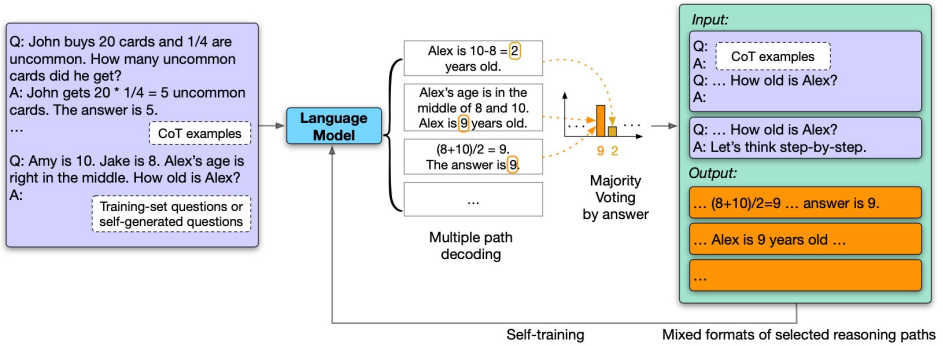


Figure 1: Overview of our method. With Chain-of-Thought (CoT) examples as demonstration (Wei et al., 2022b), the language model generates multiple CoT reasoning paths and answers (temperature $T > 0$) for each question. The most consistent answer is selected by majority voting (Wang et al., 2022b). The "high-confidence" CoT reasoning paths that lead to the majority answer are augmented by mixed formats as the final training samples to be fed back to the model for fine-tuning.

Method

- Generate and filter multiple reasoning paths
- Train with mixed formats
- Generate questions and prompts



Generate and filter multiple reasoning paths

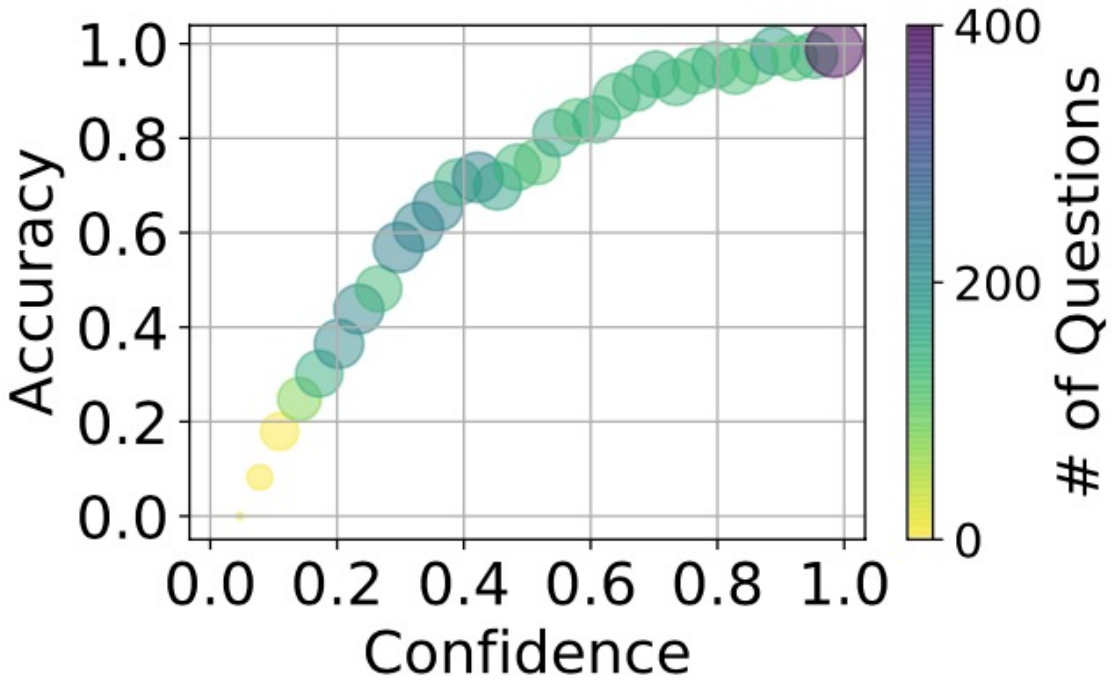
- For each training question x_i, sample m CoT reasoning paths
- The most consistent answer is selected as y_i
- The paths that leads to y_i will be put into the self-training data

Question: Stefan goes to a restaurant with his family. They order an appetizer that costs $10 and 4 entrees that are $20 each. If they tip 20% of the total, what is the total amount of money that they spend?
Multiple Path Decoding: **Output 1**: The appetizer costs $10. The entrees cost $20 each so in total 4 * $20 = $80. This means the total cost is $10 + $80 = $90. They tip 20% of it, so the total amount they spend is $90 * 1.2 = $108. The answer is 108. **Output 2**: The appetizer costs $10 and the entrees are $20 each. There are 4 entrees so the sum is $20 * 4 = $80. The waiter gets 20% of the total. 20% of $80 is $80 * .2 = $16. The answer is $80 + $16 = $96. (*Incorrect reasoning path*) **Output 3**: The appetizer costs $10. The entrees cost 4 * $20 = $80. The tip is 20% of the total, so it is 20% of the $90 they have spent. The tip is 0.2 * 90 = $18. The total they spent is $90 + $18 = $108. The answer is 108.

Table 1: Examples of 3 self-generated CoT reasoning paths given a question. Output 1 and 3 are the most consistent reasoning paths based on majority voting and kept as self-training data.

Generate and filter multiple reasoning paths

- Did not use any ground truth labels to filter out wrong cases
- Consistent CoT paths leads to accurate answers



Train with mixed formats

- Prevent the language model from overfitting to specific prompts or answer styles

Question: Amy is 10 years old. Jake is 8 years old. Alex's age is right in the middle. How old is Alex?
Selected Chain-of-Thought: Amy is 10 years old. Jake is 8 years old. Alex's age is in the middle of Amy and Jake, so Alex is (8 + 10) / 2 = 9 years old. The answer is 9.

Mixed-formats of training data:
Format 1: **Input**: *[CoT prompting examples]* + '\n' + *[Question]* + '\n' + 'A:'
Output: Amy is 10 years old. Jake is 8 years old. Alex's age is in the middle of Amy and Jake, so Alex is (8 + 10) / 2 = 9 years old. The answer is 9.

Format 2: **Input**: *[Standard prompting examples]* + '\n' + *[Question]* + '\n' + 'A:'
Output: The answer is 9.

Format 3: **Input**: *[Question]* + '\n' + 'A: Let's think step by step.'
Output: Amy is 10 years old. Jake is 8 years old. Alex's age is in the middle of Amy and Jake, so Alex is (8 + 10) / 2 = 9 years old. The answer is 9.

Format 4: **Input**: *[Question]* + '\n' + 'A:'
Output: The answer is 9.

Table 2: An example of how a reasoning path is augmented into four formats of training data with different prompts (in input) and answer styles (in output). Specifically, the *CoT prompting examples* used for each tasks are listed in Appendix A.2. The *Standard prompting examples* are the same question-answer pairs with *CoT prompting examples*, except that reasoning is removed.

Generating questions and prompts

- Question generation
  - Randomly select existing questions
  - Concatenate them in a random order as input prompt
  - Let language model generate consecutive sequences as new questions
  - Repeat and use self-consistency to keep confident answers
- Prompt generation
  - Generate CoT paths using the model itself
  - Start the answer with “A: Let’s think step by step”
  - Let the language model generate the consecutive reasoning paths
  - Use those as examples for few-shot CoT prompting

Experiments set up and datasets

- PaLM (540B-parameter model)
  - M = 32 reasoning path for each question in a training set
  - Each reasoning path is augmented into 4 formats
- Arithmetic
  - GSM8K – math problem set
  - DROP – reading comprehension benchmark which requires numerical reasong
- CommonsenseQA
  - OpenBookAQ
  - AI2 Reasoning Challenge (ARC)
- Natural Language Inference
  - Adversarial NIL subset
  - ANLI-A2 and ANLI-A3, more challenging than ANLI-A1

Results: Main results

- Results before and after language model after self-improvement (LMSI)

	Prompting Method	GSM8K	DROP	ARC-c	OpenBookQA	ANLI-A2	ANLI-A3
	Previous SOTA	82.3[a]	84.9[b]	88.7[c]	91.0[d]	64.9[d]	66.0[d]
w/o **LMSI**	Standard-Prompting	17.9	60.0	87.1	84.4	55.8	55.8
	CoT-Prompting	56.5	70.6	85.2	86.4	58.9	60.6
	Self-Consistency	74.4	78.2	88.7	90.0	64.5	63.4
LMSI	Standard-Prompting	32.2	71.7	87.2	92.0	64.8	66.9
	CoT-Prompting	73.5	76.2	88.3	93.0	65.3	67.3
	Self-Consistency	82.1	83.0	**89.8**	**94.4**	**66.5**	**67.9**

Results: Out of Domain and CoT importance

- Multi-task self-training for unseen tasks

	Self-training data	AQUA	SVAMP	StrategyQA	ANLI-A1	RTE	MNLI-M/MM
w/o **LMSI**	-	35.8	79.0	75.3	68.8	79.1	72.0/74.0
LMSI	GSM8K + DROP + ...	39.0	82.8	77.8	79.2	80.1	81.8/82.2

- Importance of training with CoT formats

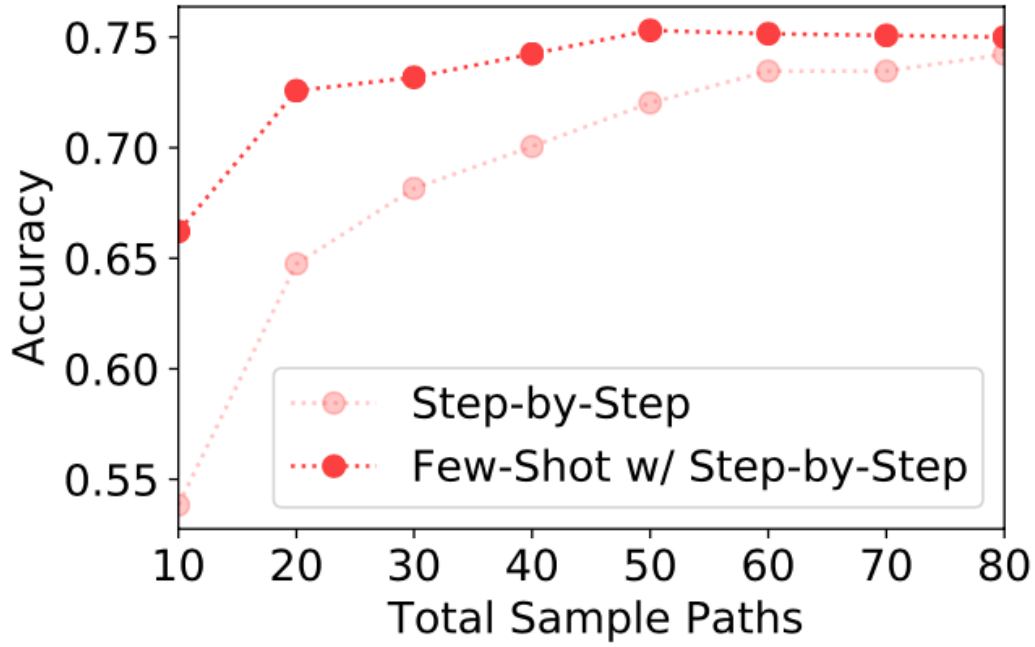
	Results on **GSM8K**	
	Standard Prompting	CoT Prompting
w/o **LMSI**	17.9	56.5
LMSI w/o CoT formats	23.6	61.6
LMSI	32.2	73.5

Push the limit of self-improvement

- Self-generating questions

	Questions used for Self-Training	Results on **GSM8K** CoT-Prompting	Self-Consistency
w/o **LMSI**	-	56.5	74.4
LMSI	Generated Questions	66.2	78.1
LMSI	Training-set Questions	**73.5**	**82.1**

- Self-generating few-shot CoT prompt

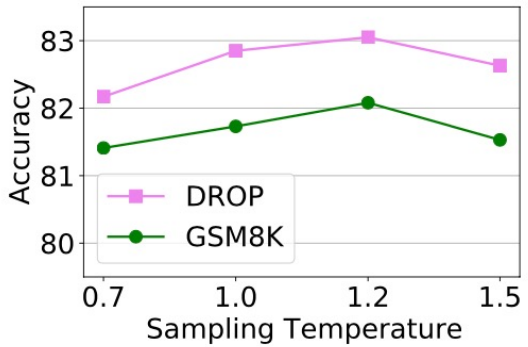


Distillation and hyperparameter study

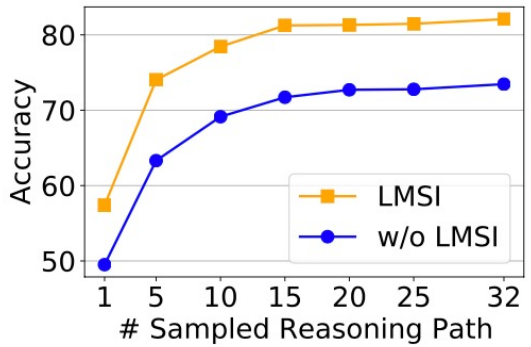
- Distillation to smaller model

	Results on **GSM8K**		
	8 billion	62 billion	540 billion
w/o **LMSI**	5.0	29.7	56.5
Distilled from **LMSI** 540 billion	33.4	57.4	-

- Temperature T = 1.2 benefits the datasets the most
- Sampled reasoning path = 15 to achieve a reasonably good accurarcy



(a) Accuracy results of **LMSI** on GSM8K and DROP test set when different sampling temperatures are applied for Self-Consistency.



(b) Accuracy results with or without **LMSI** on GSM8K test set using different numbers of sampled reasoning path for Self-Consistency.

Summary

- A large language model can self-improve by taking datasets without ground truth outputs, by leveraging CoT reasoning and self-consistency
- Achieved competitive in-domain multi-task performances as well as out-of-domain generalization
  - Achieved state-of-the-art-level results on ARC, OpenBookQA, and ANLI datasets.