



Chain of Thoughts

In-Context Learning

**ART: Automatic Multi-step
Reasoning and Tool-use for
Large Language Models**

Auto-Prompting

Tool Use

Overview

- ▶ In this paper, we present **Automatic Reasoning and Tool use (ART)**, a framework that automatically generates decompositions (multi-step reasoning) for instances of new tasks.
- ▶ The framework also selects and **uses the most appropriate available tools** (like search engines, and code execution) in individual steps.
- ▶ Given a new task, **ART retrieves demonstrations of related tasks** from a *task library* to enable few-shot decomposition and tool use.
- ▶ These demonstrations follow a flexible but **structured query language**, which enables cross-task demonstrations
- ▶ This encourages the model to **generalize from demonstrations** to decompose a new task and use tools in appropriate places, zero-shot.

Overview

Task: Translate into Pig Latin **Input:** albert goes home

A Select Examples

Task Library

Task: Anachronisms
Input: George HW ... Gulf War
Q1: [search] When was George H. W. Bush, president?
#1: From 1989-1993 ...
Q2: [EOQ]
Ans: True

Task: Arithmetic
Input: Viola bought 167 books...
Q1: [gen code] Write arithmetic as python code
#1: viola =167, nancy = 137
ans = viola - nancy
Q2: [exec code] Execute code
Q3: [EOQ]
Ans: 30

B Run Program

Frozen LLM

Q1: [search] How to write english as pig latin?
#1: Add "yay" if it starts with a vowel ...
Q2: [gen code] Write code to translate "albert goes driving" to pig latin.
#2: for w in ["albert", "goes", "home"]:
if w[0] in "aeiou":
print(w + "yay") ...
Q3: [exec] Execute snippet
#3: alberyay oesgay rivingday
Q4: [EOQ]
Ans: alberyay oesgay rivingday

Task Library examples

Tool Output LLM Output

C Fix Mistakes (optional)

Q1: [search] How to write english as pig latin?
#1: Add "yay" if it starts with a vowel ...
Q2: [gen code] Write code to translate "albert ...
#2: for w in ["albert", "goes", "home"]:
if w[0] in "aeiou":
print(w + "yay") ...
consentent_cluster = find_clstr(w)
Q3: [exec code] Execute snippet
#3: alberyay oesgay rivingday
Q4: [EOQ]
Ans: alberyay oesgay rivingday

Human Feedback

Literature

- ▶ Scaled finetuning for low-resource adaptation
 - ▶ Finetuning on an aggregated mixture of tasks (CoT, dialog, and code datasets) together with scaling models to 540B parameters achieves state-of-the-art in-context learning performance
- ▶ Prompting with intermediate reasoning steps
 - ▶ LLMs to automatically generate such **Chain of Thoughts** style prompts—**AutoCoT**— which are competitive with hand-crafted prompts in their performance on arithmetic and commonsense reasoning tasks.
- ▶ Tool Use
 - ▶ An alternative line of recent work uses self-supervision to teach LLMs to use search, translation, and a calculator ([Schick et al., 2023](#))— **Toolformer**

CoT: Chain of Thoughts

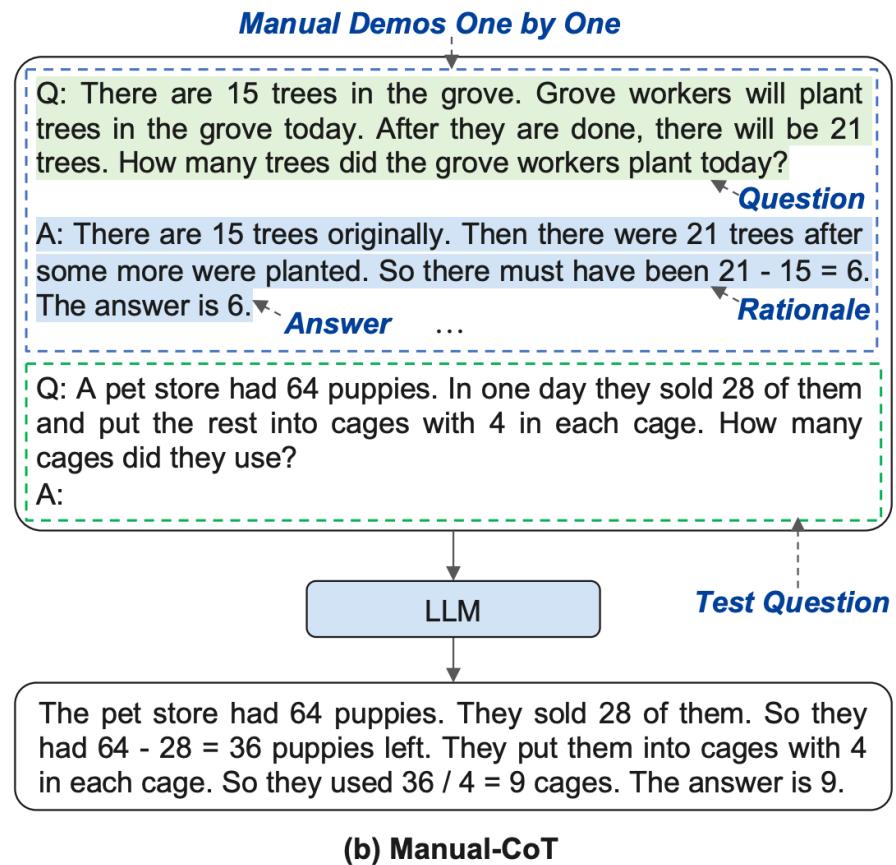
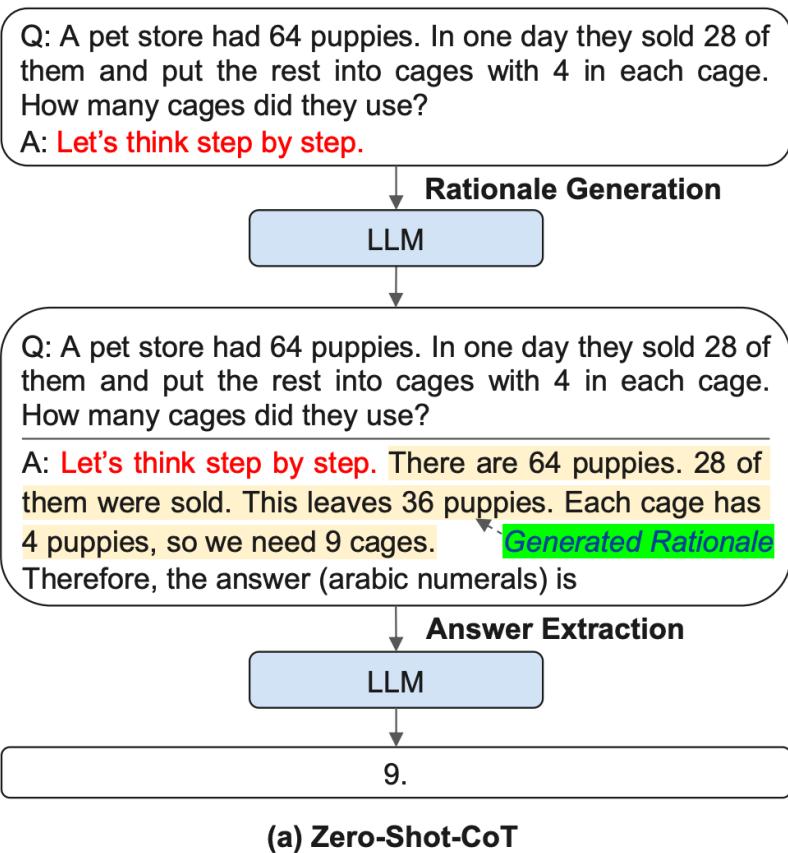
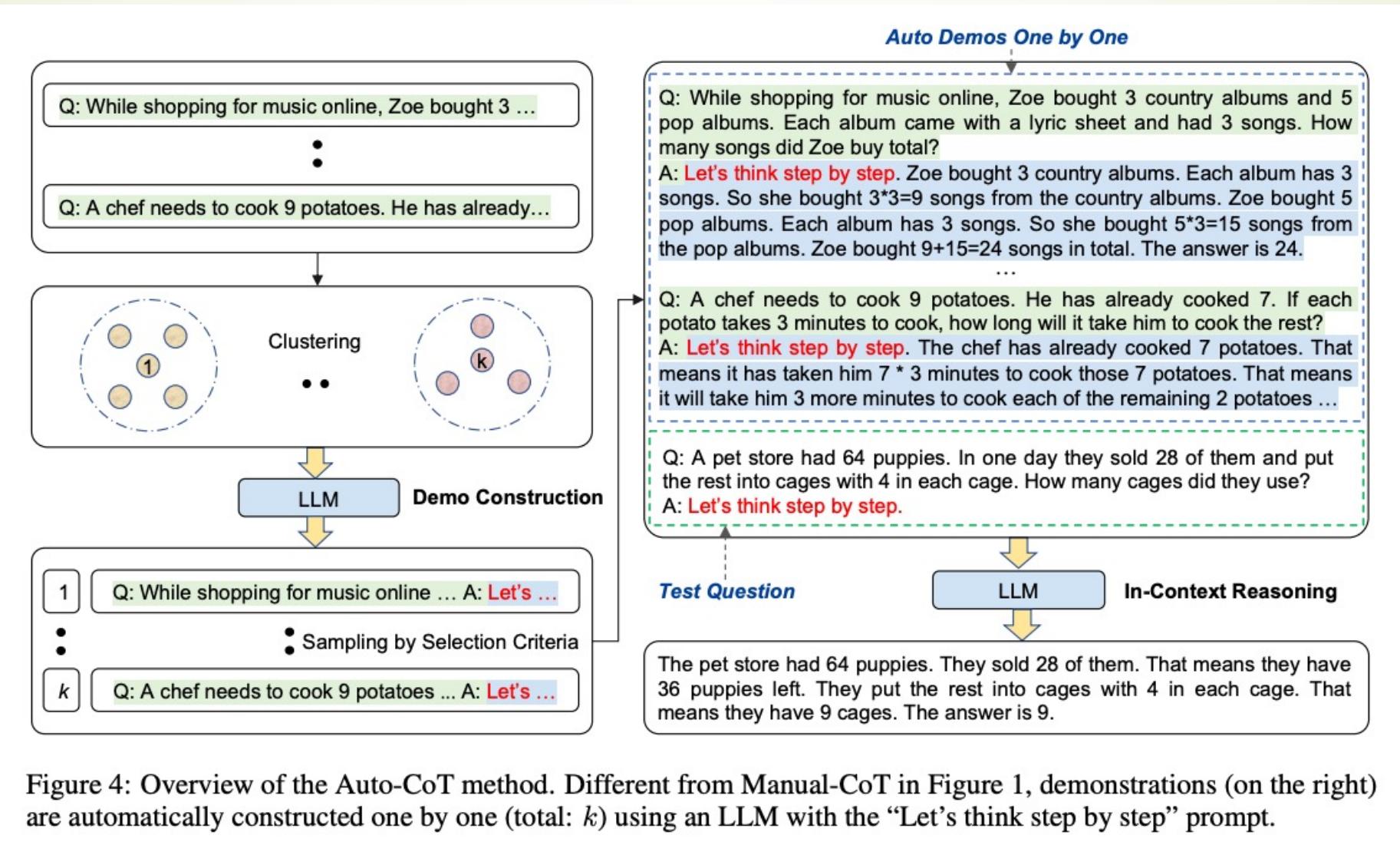


Figure 1: Zero-Shot-CoT [Kojima et al., 2022] (using the “Let’s think step by step” prompt) and Manual-CoT [Wei et al., 2022a] (using manually designed demonstrations one by one) with example inputs and outputs of an LLM.

AutoCoT



Toolformer

Your task is to add calls to a Question Answering API to a piece of text. The questions should help you get information required to complete the text. You can call the API by writing "[QA(question)]" where "question" is the question you want to ask. Here are some examples of API calls:

Input: Joe Biden was born in Scranton, Pennsylvania.

Output: Joe Biden was born in [QA("Where was Joe Biden born?")] Scranton, [QA("In which state is Scranton?")] Pennsylvania.

Input: Coca-Cola, or Coke, is a carbonated soft drink manufactured by the Coca-Cola Company.

Output: Coca-Cola, or [QA("What other name is Coca-Cola known by?")] Coke, is a carbonated soft drink manufactured by [QA("Who manufactures Coca-Cola?")] the Coca-Cola Company.

Input: \mathbf{x}

Output:

Figure 3: An exemplary prompt $P(\mathbf{x})$ used to generate API calls for the question answering tool.

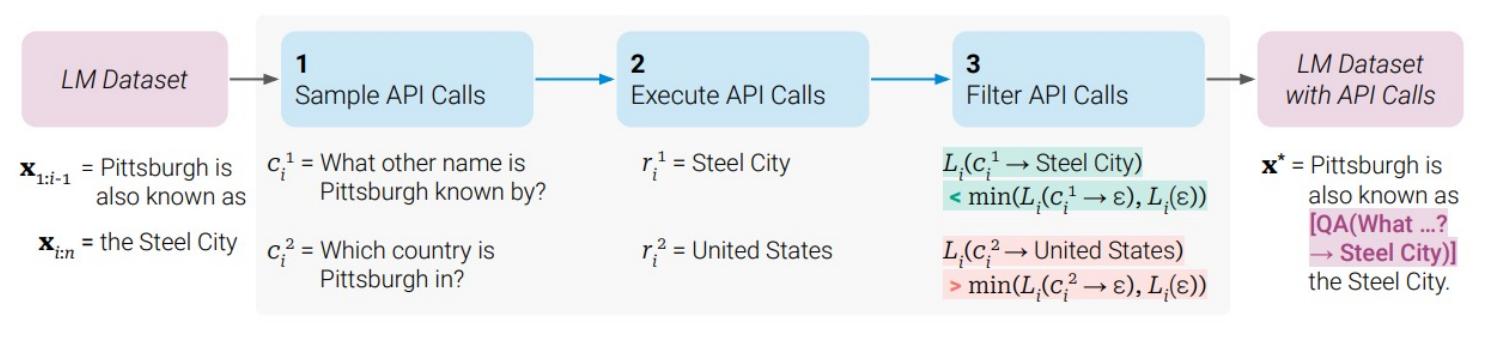


Figure 2: Key steps in our approach, illustrated for a *question answering* tool: Given an input text \mathbf{x} , we first sample a position i and corresponding API call candidates $c_i^1, c_i^2, \dots, c_i^k$. We then execute these API calls and filter out all calls which do not reduce the loss L_i over the next tokens. All remaining API calls are interleaved with the original text, resulting in a new text \mathbf{x}^* .

Toolformer

API Name	Example Input	Example Output
Question Answering	Where was the Knights of Columbus founded?	New Haven, Connecticut
Wikipedia Search	Fishing Reel Types	Spin fishing > Spin fishing is distinguished between fly fishing and bait cast fishing by the type of rod and reel used. There are two types of reels used when spin fishing, the open faced reel and the closed faced reel.
Calculator	$27 + 4 * 2$	35
Calendar	ϵ	Today is Monday, January 30, 2023.
Machine Translation	sûreté nucléaire	nuclear safety

Table 1: Examples of inputs and outputs for all APIs used.

ART: Overview

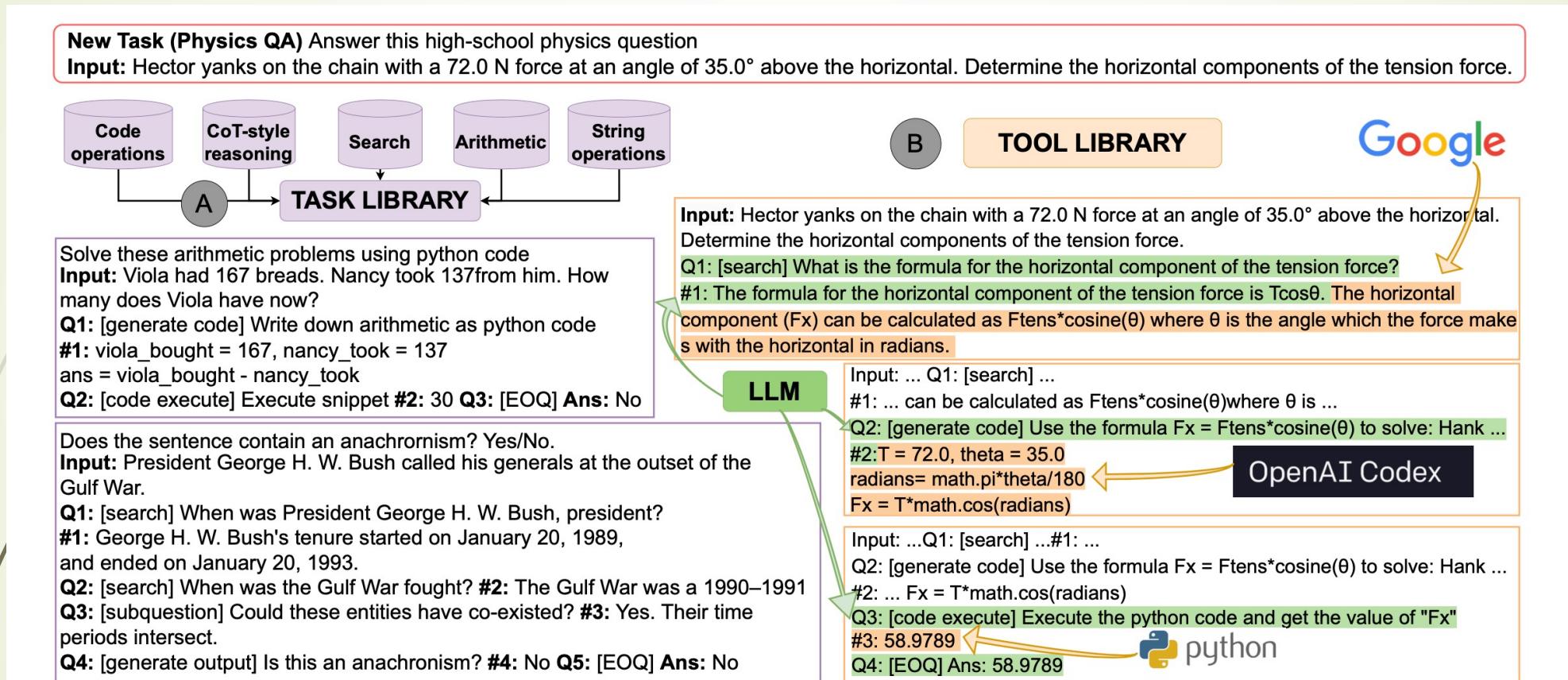
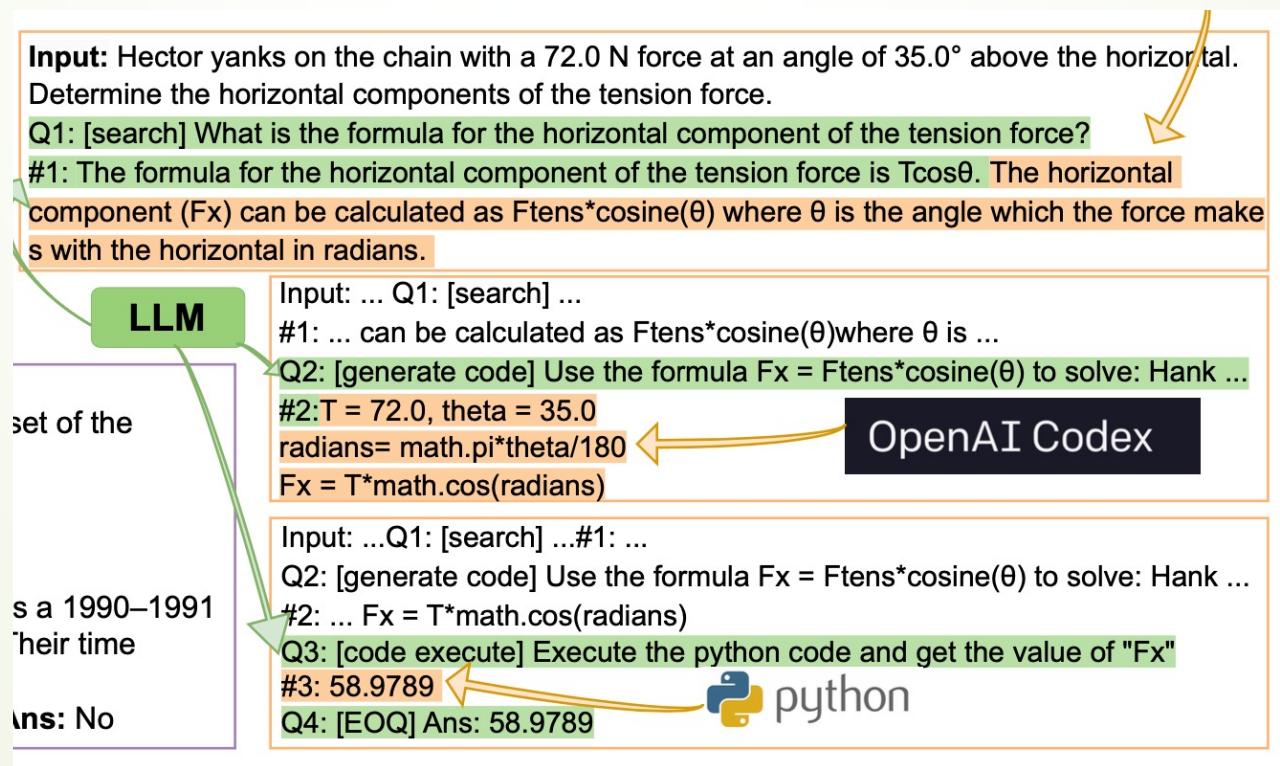


Figure 2: A run-through of ART on a new task, Physics QA. (A) Programs of related tasks like anachronisms and Math QA provide few-shot supervision to the LLM — related sub-steps and tools in these programs can be used by the LLM for cross-task generalization (shown in purple). (B) Tool use: Search is used to find the appropriate physics formula, and code generation and execution are used to substitute given values and compute the answer (shown in orange).

ART: Overview

The input node is followed by a sequence of sub-task nodes, represented as a (query, answer) pair

“Qi : ..., #i : ...”. The program ends with a dummy sub-task (“Q4: [EOQ]”), followed by a final answer node (“Ans: 59N”)



Task Library

- ▶ Beyond the Imitation Game benchmark (BIG-bench) : ~200 tasks
 - ▶ 80% of benchmark tasks are JSON, containing list of examples of inputs and targets.
 - ▶ 20% are programmatic in Python. Able to interact with model over multiple query rounds.
- ▶ ART: Focus on text classification and short answer generation in English: 91 tasks
- ▶ ART: Focus on five most used skills: over 50 tasks
 - ▶ **Arithmetic:** arithmetic and algebra problems.
 - ▶ **Code:** Generating and executing python code.
 - ▶ **Search and question decomposition:** Single or multi-step questions that require search
 - ▶ **Free-form reasoning:** Explaining step-by-step reasoning in natural language
 - ▶ **String Operations:** Reformatting/editing strings, checking string entailment, etc.
- ▶ ART: Randomly select 2-4 tasks from each of these 5 task clusters and **author decomposed programs** with appropriate tool use for these tasks. This results in a total of 15 tasks that compose the task library.

Demonstration Program

String Operations

In these examples, you are given a task description and an input. Break the input down into subtasks in order to solve the task. You can use string operations like splitting, reformatting, editing or merging. You can also use other operations like arithmetic and logic.

Description: (Date Understanding) Find the required date in MM/DD/YYYY using information about related events and dates in the input. Clue: First find what day is today.

Input: The deadline is Jun 1, 2021, which is 2 days away from now. What is the date 24 hours later in MM/DD/YYYY?

Q1: [string reformat] Jun 1, 2021 in MM/DD/YYYY

#1: 06/01/2021

Q2: [arithmetic] 06/01/2021 is 2 days away from now. What date is today?

#2: Today is 04/01/2021

Q3: [arithmetic] What date is 24 hours later than today?

#3: 05/01/2021

Q4: [EOQ]

Ans: 05/31/2021

Description: (Language games) Translate English into Pig Latin.

Input: (English) Sami made his way across the bar and hugged Layla.

Q1: [string split] What are the words in "Sami made his way across the bar and hugged Layla."?

#1: ["Sami", "made", "his", "way", "across", "the", "bar", "and", "hugged", "Layla", "."]

Q2: [string edit] Transfer the initial consonant of each word to the end of the word and adding "ay" after it.

#2: ["Amisay", "ademay", "ishay", "ayway", "acrossyay", "ethay", "arbay", "andyay", "uggedhay", "Aylalay", "."]

Q3: [string merge] Concatenate #2 into a full sentence.

#3: Amisay ademay ishay ayway acrossyay ethay arbay andyay uggedhay Aylalay.

Q4: [EOQ]

Ans: Amisay ademay ishay ayway acrossyay ethay arbay andyay uggedhay Aylalay.

Description: (Kth letter concatenation) Take the letters at position 3 of the words in a list of words and concatenate them using a space.

Input: Take the letters at position 3 of the words in "Savita Saeed Ramos Sato Yadav" and concatenate them using a space.

Q1: [string split] What are the words in "Savita Saeed Ramos Sato Yadav"?

#1: ["Savita", "Saeed", "Ramos", "Sato", "Yadav"]

Q2: [string index] What is the third letter of words in the list in #1?

#2: ["v", "e", "m", "t", "d"]

Q3: [string merge] Concatenate #2 with spaces

#3: "v e m t d"

Q4: [EOQ]

Ans: v e m t d

Description: %s

Input: %s

Q1:

Arithmetic

In these examples, you are given a task description and an input. Break the input down into subtasks in order to solve the task. You can generate python code to solve arithmetic and algebra equations in using functions from sympy.

```
from sympy import Symbol  
from sympy import simplify  
import math  
from sympy import solve_it  
# solve_it(equations, variable): solving the equations and return the variable value.
```

Description: (Aqua-rat) Solve the following arithmetic problems on ratios and fractions, writing out intermediate arithmetic calculations as python code. Store your result as a variable named 'ans'.

Input: In a flight of 600 km, an aircraft was slowed down due to bad weather. Its average speed for the trip was reduced by 200 km/hr and the time of flight increased by 30 minutes. The duration of the flight is: A)1 hour B)2 hours C)3 hours D)4 hours E)5 hours

Q1: [generate python code] write python code to solve the problem, using math and sympy.

```
#1:  
duration = Symbol('duration', positive=True)  
delay = 30 / 60  
total_disntace = 600  
original_speed = total_disntace / duration  
reduced_speed = total_disntace / (duration + delay)  
solution = solve_it(original_speed - reduced_speed - 200, duration)  
ans = solution[duration]  
print(ans)
```

Q2: [code execute] Execute the python code in #1 and get the value of "ans"

```
#2:
```

```
1.0
```

Q3: [compare] Which of the options among A)1 hour B)2 hours C)3 hours D)4 hours E)5 hours is most similar to the answer?

```
#3: A
```

```
Q4: [EOQ]
```

```
Ans: A
```

CoT Reasoning

In these examples, you are given a task description and an input. Break the input down into subtasks in order to solve the task. Thinking through the problem explicitly can be one of the substeps you use.

Description: (Sports Understanding) Determine whether an artificially constructed sentence relating to sports is plausible. The final answer should be "yes" or "no".

Input: Is the following sentence plausible? "Santi Cazorla scored a touchdown."

Q1: [think step-by-step]

#1: Let's think step-by-step. Santi Cazorla is a soccer player. Touchdown is part of American football and rugby. So the answer is no.

Q2: [EOQ]

Ans: no

Description: (Hyperbation) Identify correct adjective ordering from the two choices. This involves selecting what would be considered the more inexplicably "intuitive" sentence by a native English speaker.

Input: Which sentence has the correct adjective order:

Options:

- (A) repulsive small Brazilian exercise ship
- (B) Brazilian repulsive exercise small ship

Q1: [think step-by-step]

#1: Let's think step-by-step. When there is more than one adjective before a noun, the adjectives need to respect the following order before a noun: "[1. opinion] [2. size] [3. age] [4. shape] [5. color] [6. origin] [7. material] [8. purpose] noun".

Option (A): "repulsive small Brazilian exercise ship". (1) "repulsive" falls into the opinion category. (2) "small" falls into the size category. (3) "Brazilian" falls into the origin category. (4) "exercise" falls into the purpose category. Option (A) has the following adjective order: [1. opinion] [2. size] [6. origin] [8. purpose] (or, in numeric terms, 1 2 6 8). Because $1 < 2 < 6 < 8$ is correct, (A) has the correct ordering.

Option (B): "Brazilian repulsive exercise small ship". Option (B) has the following adjective order: [6. origin] [1. opinion] [8. purpose] [2. size] (or, in numeric terms, 6 1 8 2). Because $6 < 1 < 8 < 2$ is not correct, (B) does not have the correct ordering.

So the answer is (A).

Q2: [EOQ]

Ans: (A)

Task retrieval

Given a new task, ART retrieves N tasks from the task library to construct a dynamic multi-task prompt.

Strategy One: Performance based

- ▶ If a small number of labeled examples for the new task is available (≈ 50), we iterate over all five task clusters and select a few task programs from each cluster to compose the prompt.
- ▶ Ultimately, the task cluster with the highest performance on the held-out set of examples is chosen.

Task retrieval

Strategy Two: Use LLM to select similar tasks. For each pair, we provide a label of “Similar” or “Not similar”, and reasoning

Prompt to LLM for selecting similar tasks

Give two tasks with their descriptions and examples of inputs and outputs for the tasks, determine if they are similar. Two tasks are similar if require common subtasks like string operations, web search, translation, arithmetic, code execution, etc.

Task1: [Date understanding] Find the required date in MM/DD/YYYY using information about related events and dates in the input. Input: The deadline is Jun 1, 2021, which is 2 days away from now. What is the date 24 hours later in MM/DD/YYYY? The final answer is 05/01/2021.

Task2: [Language Games] Translate English into Pig Latin. Input: English sentence is "Sami made his way across the bar and hugged Layla". The final answer is "Amisay ademay ishay ayway acrossyay ethay arbay andyay uggedhay Aylalay." Are these similar? Yes. They both require answering in a specific string format.

Task1: [K'th letter concatenation] Take the letters at position 3 of the words in a list of words and concatenate them using a space. Input: What are the words in "Savita Saeed Ramos Sato Yadav"? The final answer is "v e m t d".

Task2: [Language Games] Translate English into Pig Latin. Input: English sentence is "Sami made his way across the bar and hugged Layla". The final answer is "Amisay ademay ishay ayway acrossyay ethay arbay andyay uggedhay Aylalay." Are these similar? Yes. They both require accessing and manipulating characters in strings.

Task 1: %s

Task 2: %s

Are these similar?

Task retrieval

	Simple Text Editing	CS Algorithms	Strategy QA	Physics Questions	Unit Interpretation	Reasoning about colored objects
Best task cluster	27.65	88.11	66.44	20.37	53.99	64.34
LLM-based task sim.	38.30	83.71	60.39	14.06	43.56	62.00

Table 12: Comparing ART results on GPT3 (175B) model with two similar task selection strategies. LLM-based similarity is worse on average compared to just choosing the best task cluster.

Tool library

Whenever a sub-task query name matches a tool name in the task library (e.g. “Qi : [search]”), generation is stopped and resumed after the tool is called and its output is incorporated into the partially completed program.

- ▶ **Search** We use SerpAPI, which provides an API for Google search. The input to search is the sequence generated by the LLM after “Qi: [search]”.
- ▶ **Code Generation** We use the Codex ([Chen et al., 2021](#)) model for code generation. Input to code generation is the sequence generated by the LM after the sub-task query symbol “Qi : [generate python code]”.
- ▶ **Code Execution** We run Python code in a virtual Python environment with arithmetic, symbolic, and scientific computing packages pre-installed.

Human feedback

- Users can incorporate feedback immediately into ART, by editing the task library and/or the tool library. Since ART generates multi-step reasoning programs that are interpretable, we explore feedback in the form of debugging, i.e. users edit existing programs rather than creating programs from scratch.

Human feedback

C

Q1: [search]...What is the formula for the horizontal component of the tension force?
#1: ... calculated as $F_{tens} \cdot \cos(\theta)$ where θ is ...
Q2: [generate code] Use formula $F_x = F_{tens} \cdot \cos(\theta)$ to solve: Hanks...
#2: $F_x = T \cdot \cos(radians)$... print(F_x)
Q3: [code execute] Execute snippet get the value of "Fx"
#3: 58.9789
Q4: [arithmetic] Round the answer to the nearest integer
#4: 59
Q5: [add unit] Add the appropriate unit of measurement to the answer.
#5: 59 N
Q4: [EOQ]
Ans: 59 N

(a) Correcting generated programs by adding additional reasoning steps

TASK LIBRARY

Q1: [string split] What are the letters in "nwist"
#1: %s
Q2: [string permutation] What are the possible permutations of 'nwisr'?
#2: ['nwist', 'nwits', 'nwsit', 'nwsti', 'nwtis', 'nwtsi', 'niwst', 'niwts', 'niswt', ...
Q3: [lookup] which word in the list is a common English word ?
#3: twins
Q4: [EOQ]
Ans: twins

```
def lookup(word_list):  
    import enchant  
    d = enchant.Dict("en_US")  
    valid_list = []  
    for word in word_list:  
        if d.check(word):  
            valid_list.append(word)
```

TOOL LIBRARY

(b) Adding additional tool use examples and new tool definitions

Figure 3: Human feedback to ART shown for (a) PQA where reasoning steps are added to the program and; (b) Word unscrambling where tool library is augmented with a new lookup tool.

Experiment setup

- ▶ We use **InstructGPT** (text-davinci-002) as the frozen LLM
- ▶ **Codex** as the code generation tool, with temperature set to 0.3.
- ▶ When constructing prompts, ART selects N tasks (where N is set to 3 in their experiments) and uses 2 demonstration programs from each selected task.
- ▶ We measure the preferred scoring metric for each task as in Srivastava et al. (2022, BigBench), and report performance averaged over 5 runs.
 - ▶ a unique preferred metric and high and low scores for this task. We use this metric when computing aggregate scores.
 - ▶ exact_string_match: Accuracy on exact.

Baselines

- ▶ Few-shot/Direct: Prompting LLMs with input-output pairs (but no intermediate reasoning).
- ▶ Auto-CoT: A baseline that automatically generates multi-step reasoning in natural language. A random subset of 5 examples is first used to elicit CoT-style reasoning (*Input + Let's think step-by-step.*).
- ▶ ART-tool: ART with tool-use turned off, i.e. the LLM generates the output of every sub-step, to verify the gains from tool use
- ▶ GPT-3 Best: Best published GPT-3/Codex (175B) result with multi-step decomposition and/or tool use. **These often include additional human supervision to decompose reasoning steps, and external tools to boost performance**

Results on tasks in the task library

Task Name (Cluster)	Few Shot	AutoCot	ART w/o Tool Use	ART	GPT-3 Best
Anachronisms (Search)	71.3 ⁵	51.48	70.87	75.66	-
Musique (Search)	2.03 ⁵	12.88	10.04	19.19	15.2 ³
Hindu Knowledge (Search)	85.02 ⁵	73.03	83.42	87.98	-
Known Unknown (Search)	68.90 ⁵	56.09	80.43	80.43	-
Δ with ART (Search)	+9.0	+17.44	+4.6	+4.0	
Elementary Math QA (Arithmetic)	56.40 ⁷	74.52	58.04	68.04	-
Aqua-rat (Arithmetic)	20.54 ⁷	34.41	36.29	54.20	54.1 ⁴
GSM8K (Arithmetic)	7.79 ⁷	21.99	53.4	71.00	71.6 ⁴
Navigate (Arithmetic)	60.7 ⁷	61.7	72.4	72.4	85.90 ¹
Δ with ART (Arithmetic)	+30.0	+18.25	+11.4	-4.7	
K'th letter concatenation (String)	3.2 ⁵	0.64	8.19	40.00	98.0 ²
Language games (String)	35.14 ⁵	18.58	11.19	23.08	-
Date Understanding (String)	37.53 ⁵	38.90	52.05	-	70.41 ¹
Auto Debugging (Code)	62.94 ⁵	38.24	55.29	62.94	-
Code Description (Code)	97.99 ⁷	88.67	84.67	88.00	-
Formal Fallacies (CoT)	44.84 ⁵	56.1 ¹ <small>Go to page 11</small>	64.76	-	58.4 ¹
Hyperbation (CoT)	62.72 ⁵	55.4	80.80	-	72.4 ¹
Δ with ART (Misc)	+9.6	+16.4	+13.7	-15.4	
Δ with ART (Overall)	+14.90	+17.17	+7.91	-9.0	

Table 2: ART performance on tasks in the task library. (¹Human-crafted CoT (Wei et al., 2022; Suzgun et al., 2022), ²Decomposed Prompting (Khot et al., 2022), ³Self-Ask (Press et al., 2022), ⁴PoT (Chen et al., 2022), ⁵InstructGPT (Ouyang et al., 2022), ⁷Code-davinci-002 (Chen et al., 2021)). (-) For tasks using CoT reasoning, no tool use is used.

Results on BigBench tasks (not in task library)

Task Name (Cluster)	Few Shot	AutoCot	ART w/o Tool Use	ART	GPT-3 Best
Test Tasks					
Sentence Ambiguity (Search)	70.67 ⁵	51.47	71.00	73.33	-
Strategy QA (Search)	55.49 ⁵	27.22	59.37	66.44	-
Physics (Search)	70.09 ⁵	61.83	59.13	67.55	-
Δ with ART (Search)	+3.7	+22.27	+ 5.9		
Physics Questions (Arithmetic)	7.02 ⁵	5.56	6.30	20.37	-
Operators (Arithmetic)	71.23 ⁷	75.52	71.80	92.00	-
Unit interpretation (Arithmetic)	58.2 ⁷	41.20	51.4	53.99	-
Repeat copy logic (Arithmetic)	50.01 ⁷	15.63	31.25	44.38	-
Object Counting (Arithmetic)	39.2 ⁷	26.80	42.2	87.00	81.20 ¹
Penguins in a table (Arithmetic)	58.23 ⁷	40.40	68.86	77.85	72.34 ¹
Reasoning about objects (Arithmetic)	71.00 ⁷	33.33	45.35	64.34	52.69 ¹
Tracking shuffled objects (Arithmetic)	22.39 ⁷	19.44	18.14	37.67	36.32 ¹
Δ with ART (Arithmetic)	+19.0	+36.7	+ 23.1		+6.1
Word Unscramble (String)	40.72 ⁷	32.44	23.03	42.7	-
Simple Text Editing (Code)	35.31 ⁵	30.21	20.74	27.65	-
CS Algorithms (Code)	73.48 ⁷	0.0	41.59	88.11	-
Sports Understanding (CoT)	69.74 ⁵	51.47	92.89	-	86.59 ¹
Snarks (CoT)	54.58 ⁵	57.24	57.13	-	65.2 ¹
Disambiguation QA (Free-form)	55.03 ⁵	48.45	55.89	-	60.62 ¹
Temporal sequences (CoT)	55.80 ⁷	19.70	49.5	-	81.8 ¹
Ruin names (CoT)	71.01 ⁵	55.28	60.22	-	-
Δ with ART (Misc)	2.4	22.5	24.37		-9.4
Δ with ART (Overall)	+6.9	+24.6	+16.7		-1.7

Results on MMLU tasks

Task Name (Cluster)	Few Shot	AutoCot	ART w/o Tool Use	ART	GPT-3 Best
Test Tasks					
MMLU					
College Computer Science (Search)	41.00	43.99	63.40	67.80	63.6 ⁶
Astronomy (Search)	62.10	41.48	76.71	79.1	62.5 ⁶
Business Ethics (Search)	61.60	48.8	77.17	81.16	72.7 ⁶
Virology (Search)	50.03	49.52	71.60	71.49	50.72 ⁶
Geography (Search)	77.67	57.07	70.30	71.71	81.8 ⁶
Mathematics (Arithmetic)	36.67	33.77	39.50	45.66	34.5 ⁶
Δ with ART (MMLU)	+14.6	+23.7	+3.0		+8.5

Improving ART: Self-consistency

	Simple Text Editing	CS Algorithms	Strategy QA	Physics Questions	Unit Interpretation	Reasoning about colored objects
ART	27.65	88.11	66.44	20.37	53.99	64.34
+ Self Consistency	30.67(+3.0)	90.99(+2.9)	70.76(+4.3)	24.07(+3.7)	57.20(+3.2)	69.11(+4.8)

Table 5: Improving ART via self-consistency (Wang et al., 2022). Ensembling model generations over 15 runs further boosts performance.

Improving ART: Human Feedback

Task	CoT +Human		ART + Human	GPT-3 Best	Human Feedback
CS Algorithms	0.0	23.0	88.11	92.73	73.48
Reasoning about objs.	33.33	67.75	64.34	98.90	71.00
Repeat Copy Logic*	15.63	45.22	44.38	80.31	50.01
Sentence Ambiguity	51.47	72.33	73.33	83.67	70.67
Simple Text editing*	30.21	35.31	27.65	36.11	35.31
Strategy QA*	27.22	29.19	66.44	69.15	55.49
Physics*	61.83	68.21	67.55	72.55	70.09
Temporal Sequences	19.70	30.22	49.5	88.00	81.8
Track Shuffled objs.	19.44	36.48	37.67	99.86	36.32
Unit Interpretation*	41.2	41.2	53.99	95.0	58.2
Word Unscrambling*	32.44	33.40	42.70	62.11	40.72
Average	30.2	43.8	56.0	79.85	58.5

Table 6: Improving ART and free-form CoT via self-consistency and human-in-the-loop feedback. (*) indicates that human-in-the-loop improvement was done over automatically generated CoT reasoning for these tasks. Feedback for ART includes correcting sub-steps in programs (“C:"), adding additional sub-steps(“A:"), and defining new tools(“T:"). Note that only five examples were edited for each task.

Conclusion

- ▶ Our main contributions include **a lightweight grammar to represent multi-step reasoning as a program** (with tool calls and arguments)
- ▶ an extensible library of seed tasks for which **programs** are authored,
- ▶ a **tool library** that consists of useful external utilities like search, code generation, and execution.
- ▶ The **interpretable reasoning framework** also allows humans to improve task decomposition and tool use to boost performance.