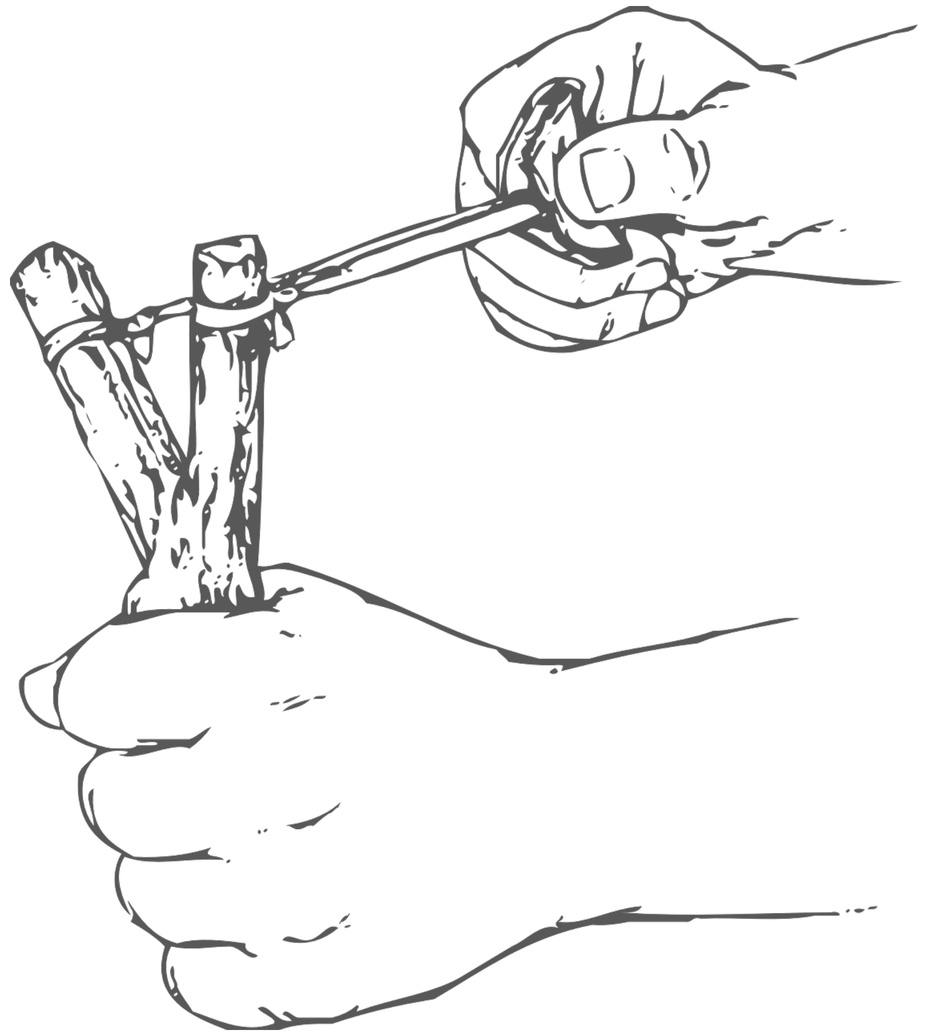
Understanding Large Language Models

Carsten Eickhoff, Michael Franke and Polina Tsvilodub

Session 03: State-of-the-art LLMs, in-context learning & prompt-engineering

Main learning goals



1. taking stock
    - recap RNN, LSTM, Transformer
2. Large Language Models (2020-2022)
    - close-up LLaMA
3. in-context learning
    - few-shot learning
    - prompt understanding
4. sophisticated prompting
    - chain-of-thought
    - self-consistency
    - knowledge generation
    - tree-of-thought

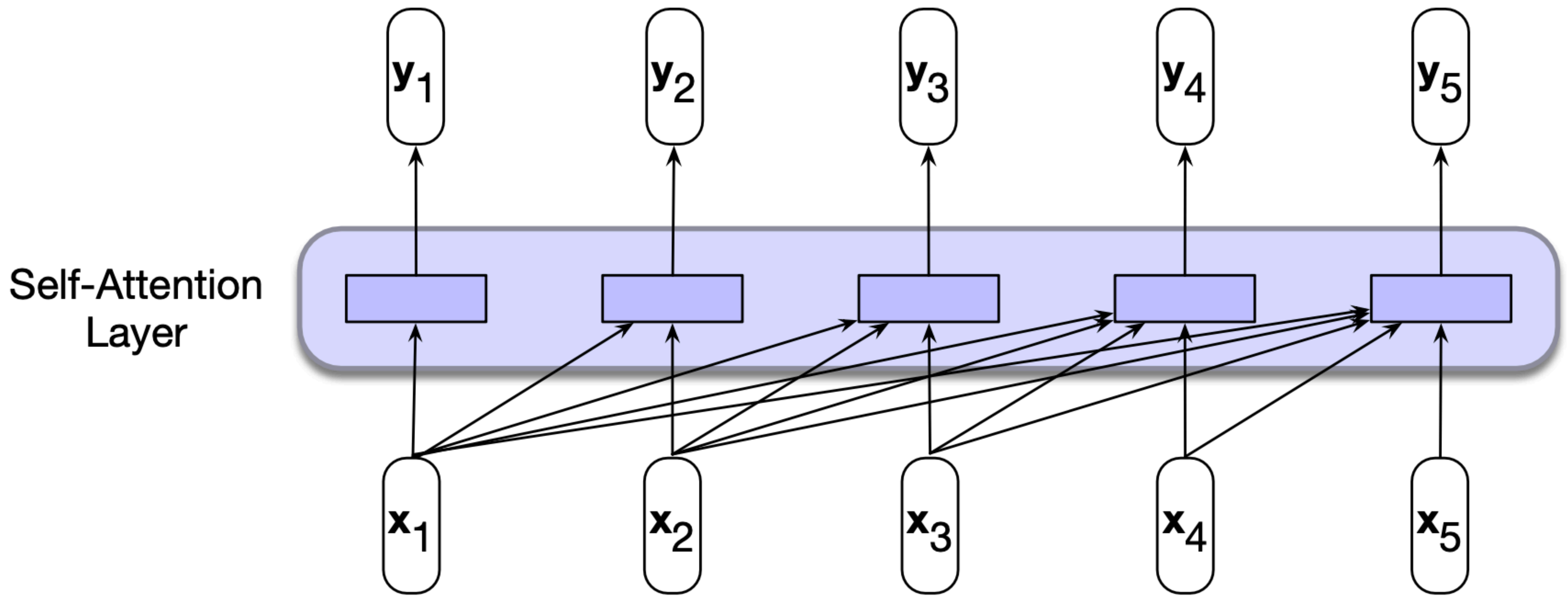
Taking stock & looking ahead

Language & sequence models

welcome to the (terminological) jungle

- **language model** (wide definition)
  - any kind of model used to perform tasks with natural language
- **language model** (narrow definition)
  - probabilistic model of natural language
- **neural language model** (super narrow definition)
  - neural network predicting natural language, given some input
    - autoregressive, masked, image captioning … whatever
- **embedding**
  - vector representation of a chunk of natural language
    - used for many different tasks, including language modeling and more
    - can arise as a 'by-product' of language modeling
- **sequence model**
  - model that processes and/or generates a token sequence

Autoregressive / left-to-right

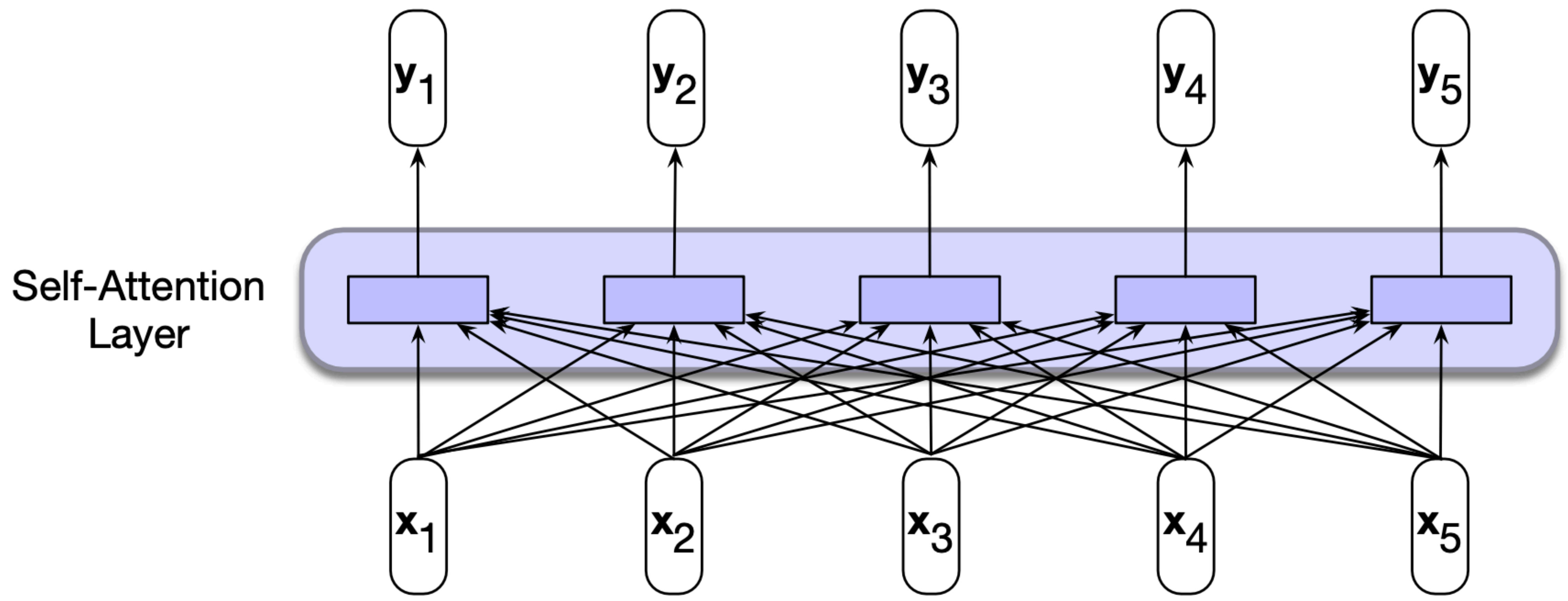


computation for input $\mathbf{x}_1, \ldots, \mathbf{x}_3$ blind to $\mathbf{x}_4$ and $\mathbf{x}_5$

$\mathbf{y}_5$ is embedding for input $\mathbf{x}_1, \ldots, \mathbf{x}_5$

$\mathbf{y}_5$ is a "left-contextual embedding"

Bidirectional



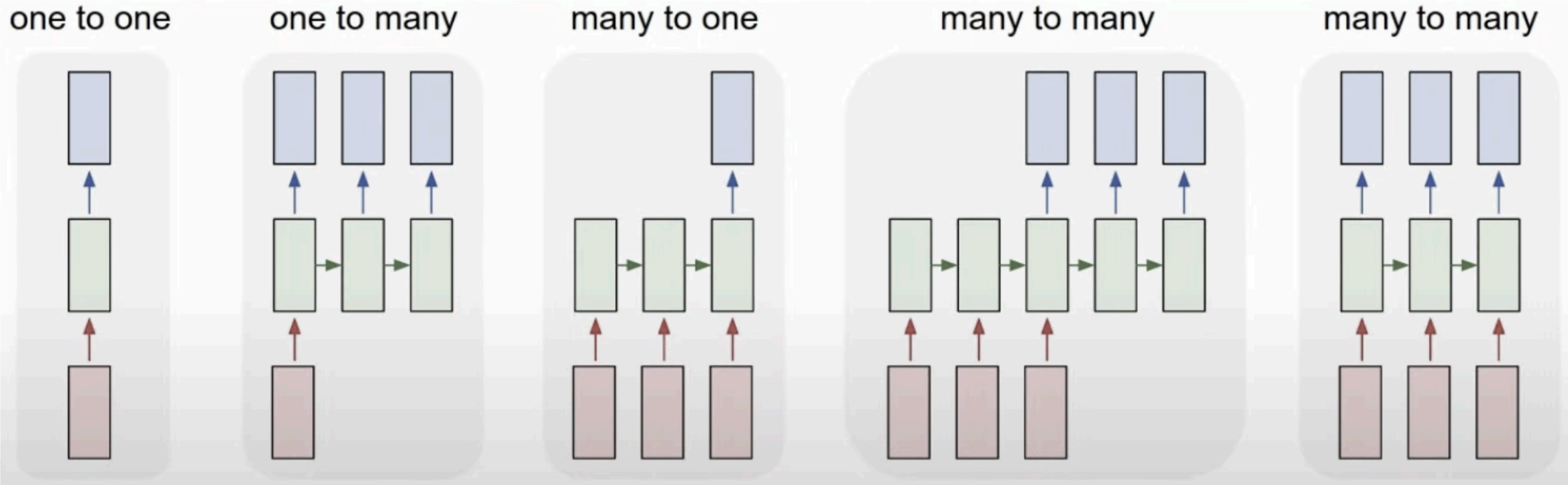
computation for input $\mathbf{x}_1, \ldots, \mathbf{x}_3$ sees $\mathbf{x}_4$ and $\mathbf{x}_5$

$\mathbf{y}_1, \ldots, \mathbf{y}_5$ is embedding for input $\mathbf{x}_1, \ldots, \mathbf{x}_5$

$\mathbf{y}_i$ are bidirectional "contextual embeddings"

Different kinds of sequence processing models

sequence as input and/or (simultaneous) output



Think break:

Which of these architectures go with which kinds of applications?

Where, if at all, would you locate autoregressive / masked language modeling?

Memory, attention & relevance

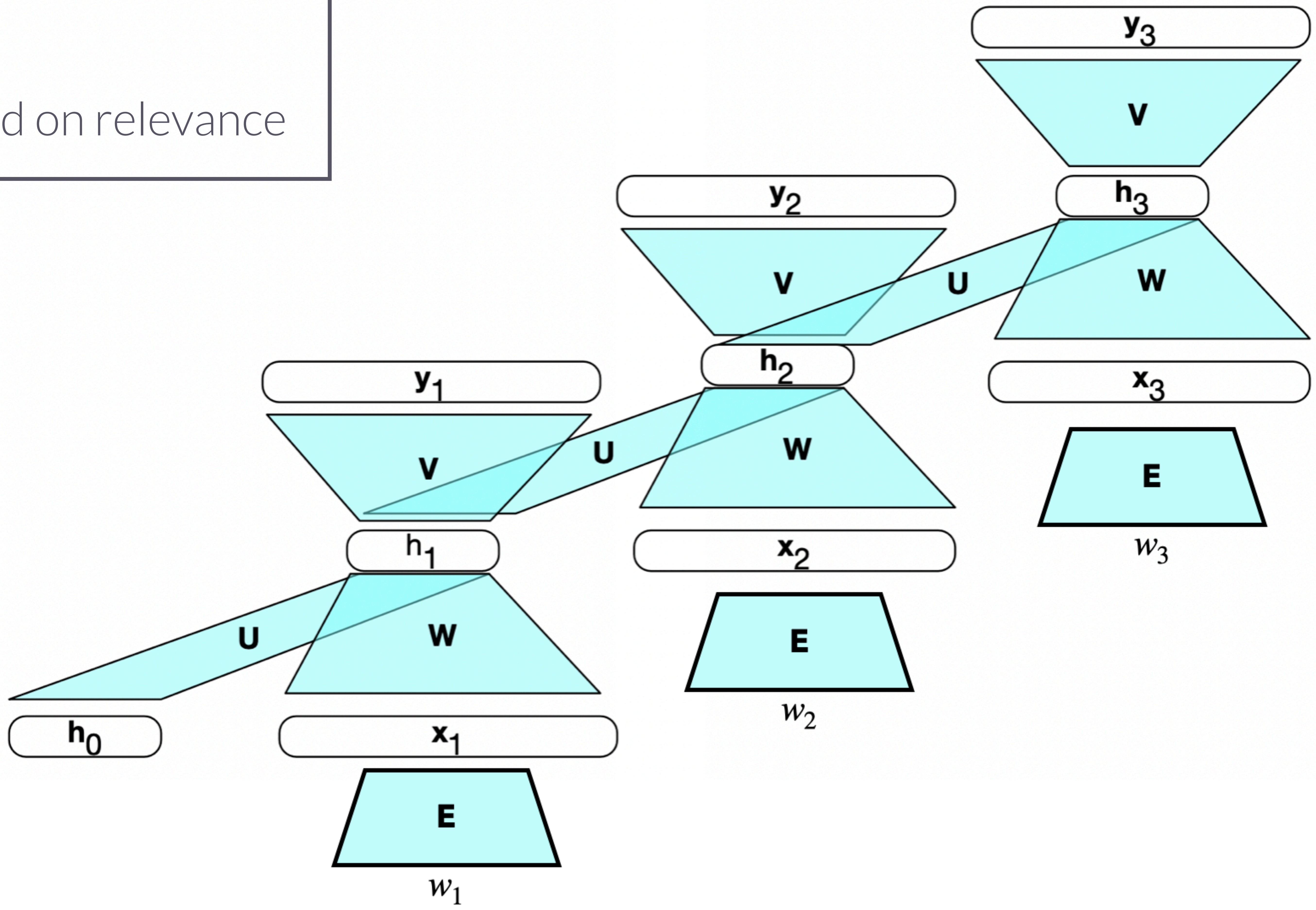
in humans & machines

- human memory
  - optimized to retrieve context-relevant info when cued
  - optimized to store information deemed relevant
- human attention
  - filters external and internal stimuli based on relevance

Memory, attention & relevance
in humans & machines

- human memory
  - optimized to retrieve context-relevant info when cued
  - optimized to store information deemed relevant
- human attention
  - filters external and internal stimuli based on relevance

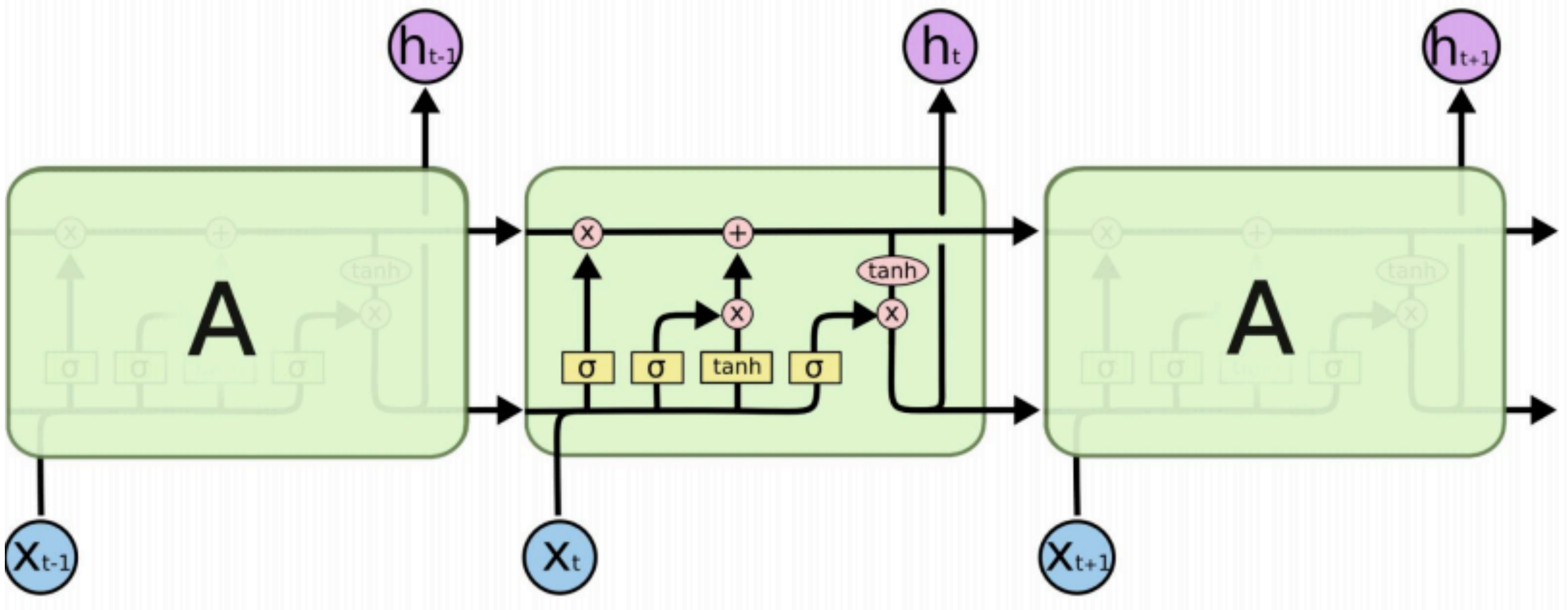
RNNs



Memory, attention & relevance
in humans & machines

- human memory
  - optimized to retrieve context-relevant info when cued
  - optimized to store information deemed relevant
- human attention
  - filters external and internal stimuli based on relevance

LSTMs

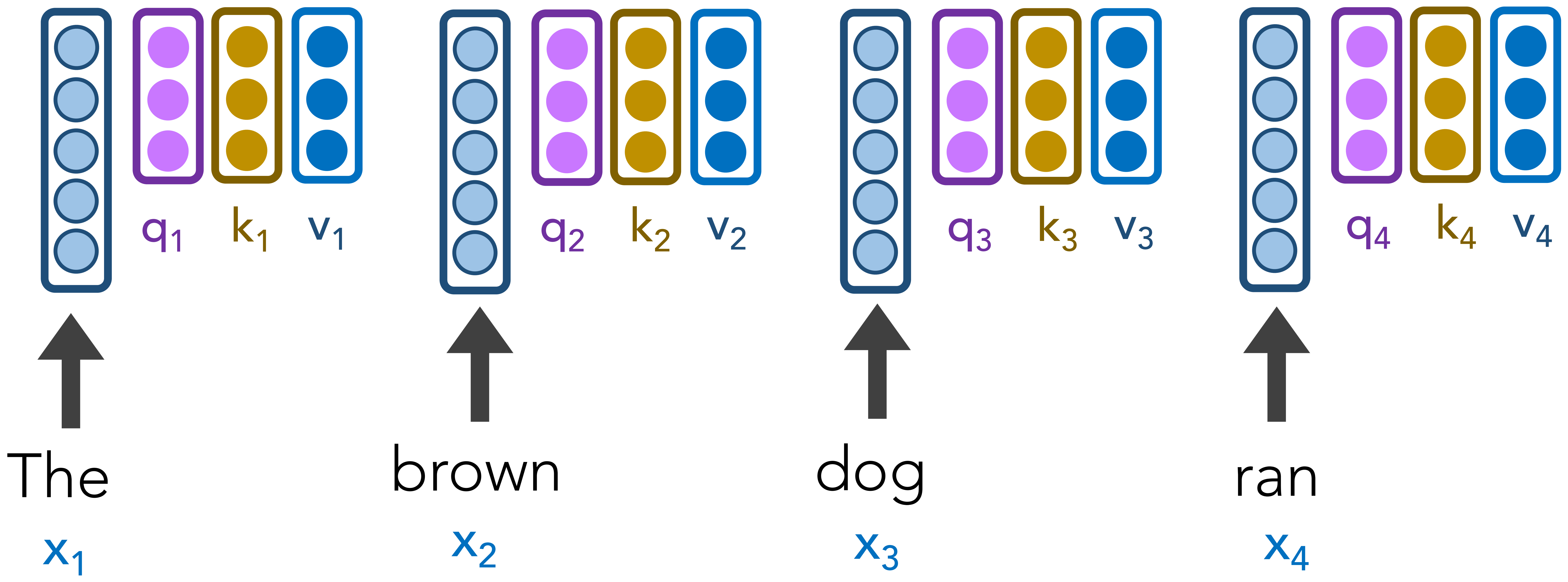


Memory, attention & relevance
in humans & machines

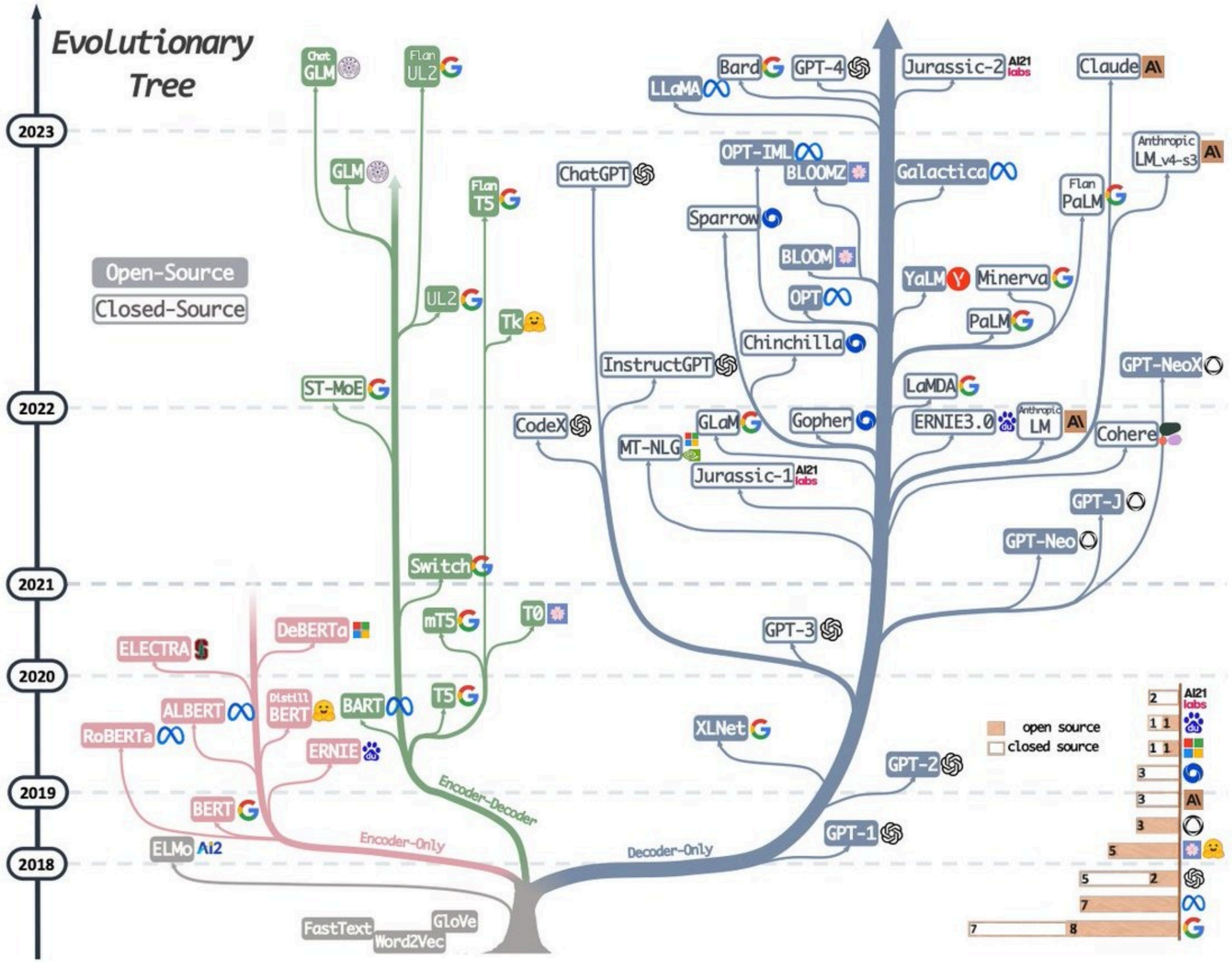
- human memory
  - optimized to retrieve context-relevant info when cued
  - optimized to store information deemed relevant
- human attention
  - filters external and internal stimuli based on relevance

query: what's relevant for me now?
key: what **kind of info** do I have?
value: what **concrete info** do I have?

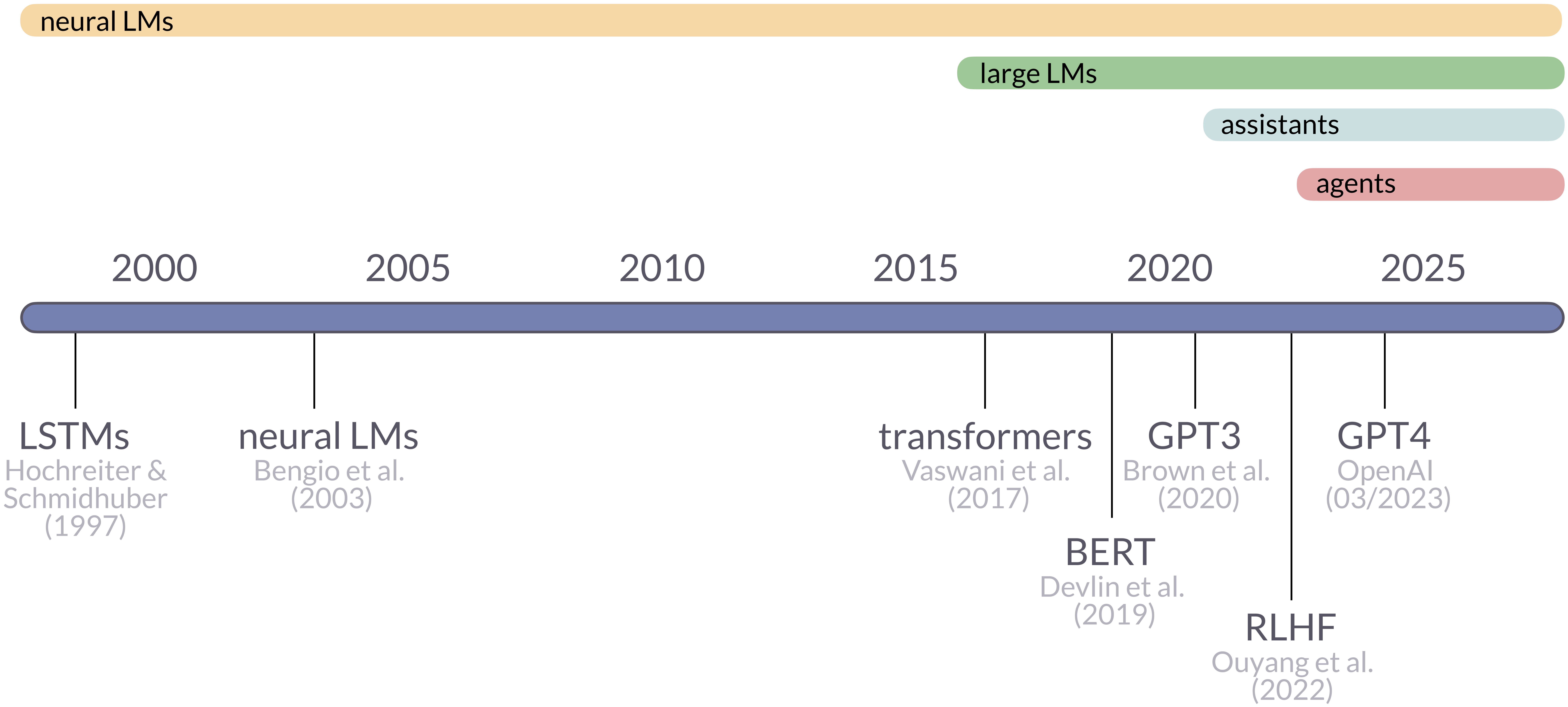
Transformers



State-of-the-art LLMs


Evolutionary Tree
2023
2022
2021
2020
2019
2018
Open-Source
Closed-Source
Chat GLM
Flan UL2
GLM
Flan T5
UL2
Tk
ST-MoE
Switch
mT5
T0
T5
BART
DeBERTa
ELECTRA
ALBERT
Distill BERT
RoBERTa
ERNIE
BERT
ELMo
Ai2
FastText
GloVe
Word2Vec
Encoder-Only
Encoder-Decoder
Decoder-Only
Bard
GPT-4
Jurassic-2
Claude
LLaMA
OPT-IML
BLOOMZ
Galactica
Anthropic LM v4-s3
ChatGPT
Sparrow
Flan PaLM
BLOOM
YaLM
Minerva
OPT
PaLM
Chinchilla
InstructGPT
GPT-NeoX
LaMDA
CodeX
GLaM
Gopher
ERNIE3.0
Anthropic LM
Cohere
MT-NLG
Jurassic-1
GPT-J
GPT-Neo
GPT-3
XLNet
GPT-2
GPT-1
open source
closed source

source



neural LMs
large LMs
assistants
agents
2000
2005
2010
2015
2020
2025
LSTMs
Hochreiter & Schmidhuber (1997)
neural LMs
Bengio et al. (2003)
transformers
Vaswani et al. (2017)
BERT
Devlin et al. (2019)
GPT3
Brown et al. (2020)
RLHF
Ouyang et al. (2022)
GPT4
OpenAI (03/2023)

Kinds of Large Language Models

core LLMs
(foundation models)

- predict statistically likely next token
- e.g.,
  - GPT-2
  - LLaMA2 / LLaMA3
  - ...

today

prepped LLMs
(assistants)

- fine-tuned (e.g. RLHF)
- predict token likely to please the user
- e.g.,
  - GPT-3.5
  - LLaMA3 Instruct
  - ...

next session

LLM-based applications
(agents)

- algorithm using LLMs
- e.g.:
  - sophisticated prompting
  - Chat-GPT w/ tools
  - AutoGPT
  - neuro-symbolic models
  - ...

next to next session

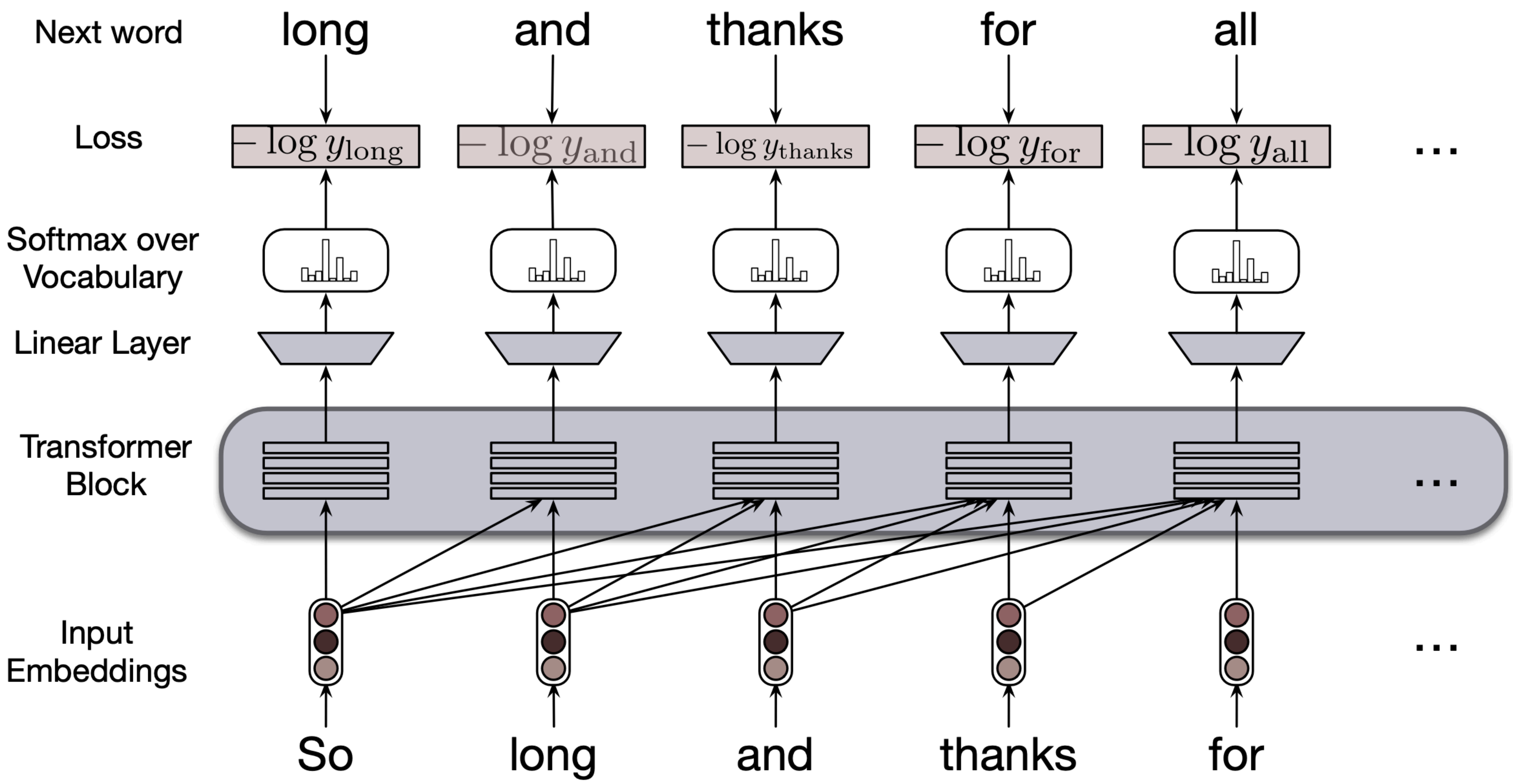
WEIRD WYOMING

- just as experimental psychology is **WEIRD**
  - Western
  - Eductated
  - Industrialized
  - Rich
  - Democratic
- usual LLM training data is from **WYOMING**
  - Western
  - Young
  - Opinionated
  - Males with
  - Internet from
  - Non-marginalized
  - Groups

Bender et al. (2021)

Language modeling objective

maximize token-in-context probability



The LLaMA model family

Large Language Model Meta AI

Name	Release date	Parameters	Context length	Corpus size	Commercial viability?
LLaMA	Feburary 24, 2023	• 6.7B • 13.0B • 32.5B • 65.2B	2048	1.0-1.4T	No
Llama 2	July 18, 2023	• 6.7B • 13.0B • ~34B (unreleased) • 69B	4096	2T	Yes
Llama 3	April 18, 2024	• 8B • 70.6B • 400B+ (unreleased)	8912	15T	Yes

https://en.wikipedia.org/wiki/LLaMA

base model only

base model,
fine-tuned & chat

base & instruction

research reports:

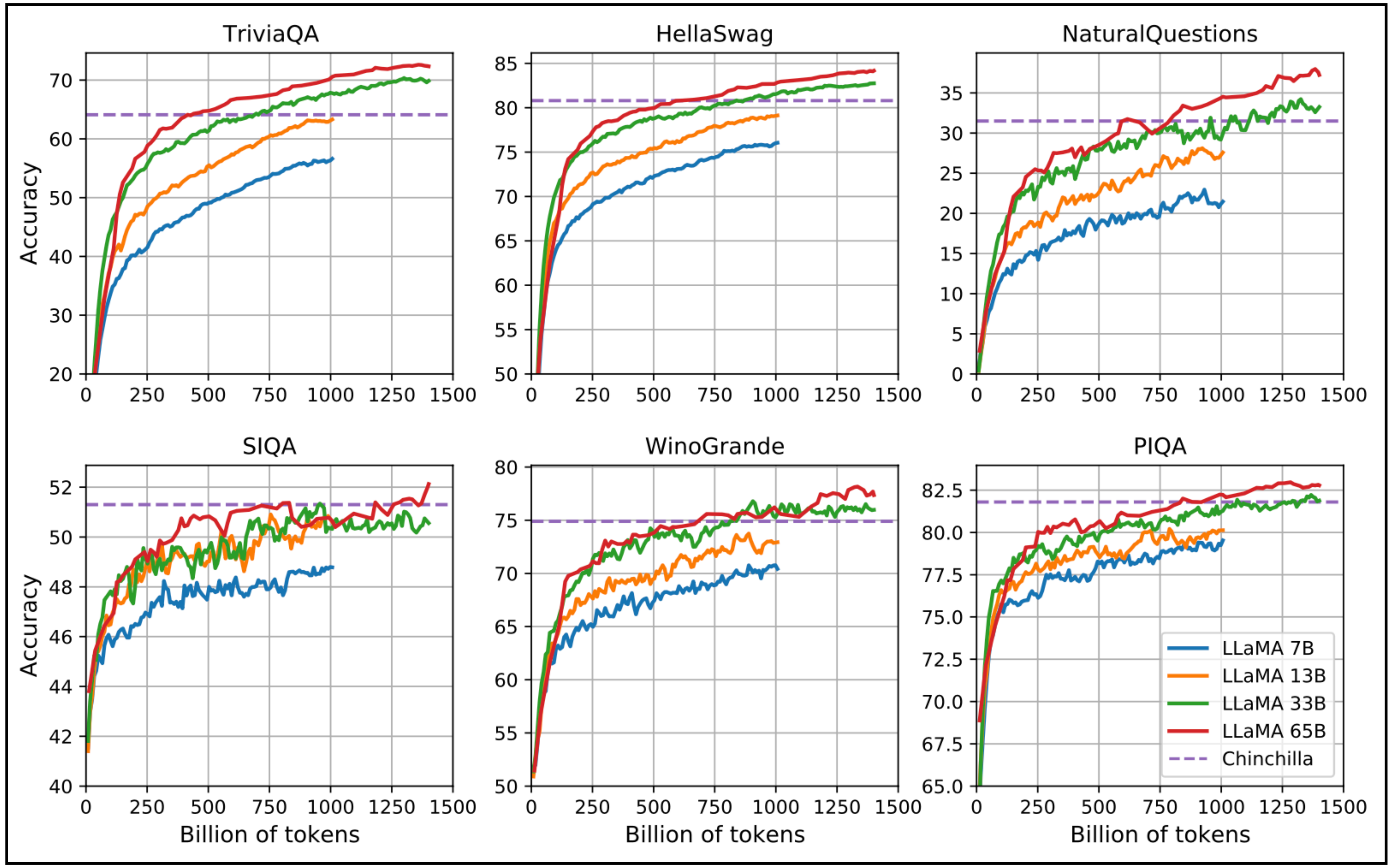
LLaMA ⇒ Touvron, Lavril et al. (2023)

Llama-2 ⇒ Touvron, Martin et al. (2023)

Llama-3 ⇒ ... pending ...

Building LLaMA
motivation

- trained on publicly available data only
- weights released on request
  - restricted commercial use
- two ways of spending compute
  - (1) larger model w/ same training volume
  - (2) smaller model w/ more training [← LLaMA]
    - (2) is cheaper at inference time
- 7B, 13B, 70B parameter models
  - 13B LLaMA matches best GPT-3 model of the time
    - the latter at 175B parameters
  - 70B LLaMA matches leading models of the time Chinchilla (70B) and PaLM (540B)



Touvron, Lavril et al. (2023)

Building LLaMA
training details

- training data set:
  - various sources, all preprocessed, see →
- tokenization
  - byte-pair encoding ⇒ 1.4T tokens (!)
- model architecture
  - autoregressive transformer w/ tweaks
    - normalize input, not output of transformer sublayers
    - replace ReLU activation w/ SwiGLU (Shazeer 2020)
    - rotary positional embeddings
- optimizer
  - AdamW w/ carefully chosen parameters
    - the latter at 175B parameters
  - 70B LLaMA matches leading models of the time Chinchilla (70B) and PaLM (540B)
- compute optimization (see paper)

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

Table 1: **Pre-training data.** Data mixtures used for pre-training, for each subset we list the sampling proportion, number of epochs performed on the subset when training on 1.4T tokens, and disk size. The pre-training runs on 1T tokens have the same sampling proportion.

params	dimension	n heads	n layers	learning rate	batch size	n tokens
6.7B	4096	32	32	$3.0e^{-4}$	4M	1.0T
13.0B	5120	40	40	$3.0e^{-4}$	4M	1.0T
32.5B	6656	52	60	$1.5e^{-4}$	4M	1.4T
65.2B	8192	64	80	$1.5e^{-4}$	4M	1.4T

Touvron, Lavril et al. (2023)

Building LLaMA
example results

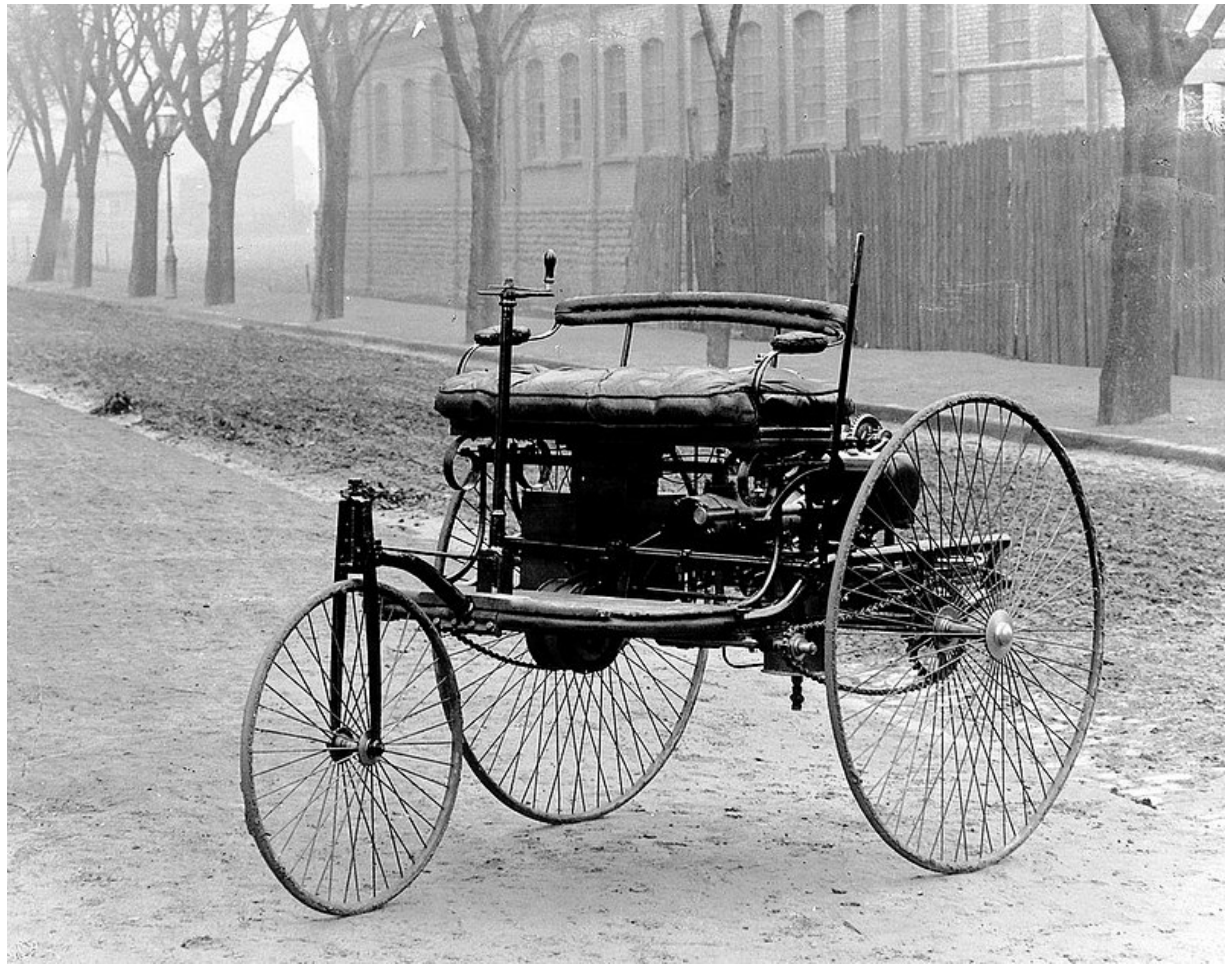
		BoolQ	PIQA	SIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA
GPT-3	175B	60.5	81.0	-	78.9	70.2	68.8	51.4	57.6
Gopher	280B	79.3	81.8	50.6	79.2	70.1	-	-	-
Chinchilla	70B	83.7	81.8	51.3	80.8	74.9	-	-	-
PaLM	62B	84.8	80.5	-	79.7	77.0	75.2	52.5	50.4
PaLM-cont	62B	83.9	81.4	-	80.6	77.0	-	-	-
PaLM	540B	**88.0**	82.3	-	83.4	**81.1**	76.6	53.0	53.4
LLaMA	7B	76.5	79.8	48.9	76.1	70.1	72.8	47.6	57.2
	13B	78.1	80.1	50.4	79.2	73.0	74.8	52.7	56.4
	33B	83.1	82.3	50.4	82.8	76.0	**80.0**	**57.8**	58.6
	65B	85.3	**82.8**	**52.3**	**84.2**	77.0	78.9	56.0	**60.2**

Table 3: **Zero-shot performance on Common Sense Reasoning tasks.**

Touvron, Lavril et al. (2023)

On engineering, or how to overpower a simple idea

you’ve come a long way, baby



Benz Patent-Motorwagen (1885)

Red Bull Racing RB18 (2022)

In-Context Learning

demo

Zero-shot prompting

- may include task instruction, but has no example or illustration
  - works well only for models fine-tuned on instruction-following data
  - works well only for frequent (simple) tasks

INPUT

```
Classify the sentence as positive, neutral or negative.
Sentence: This class is super exciting!
Sentiment:
```

OUTPUT

```
positive
```

In-context learning

a.k.a. k-shot learning

- prompt with k pairs of **demonstrations** (x_i, y_i)
  - x_i is the **input**
  - y_i is the **output** or **label**
- add only the target input x_t of test item (x_t, y_t)
- ICL is an emerging ability
  - boosts performance on common tasks, even though models are NOT trained do in-context learning
  - shown first for GPT-3 by Brown et al. (2020)
- works with or without (!) task instructions
- common observation:
  - performance increases with size k of demonstrations

INPUT

```
This class in interesting. neutral
This class is my favorite. positive
This class is a drag. negative.
This class is super exciting.
```

OUTPUT

```
positive
```

In-context learning

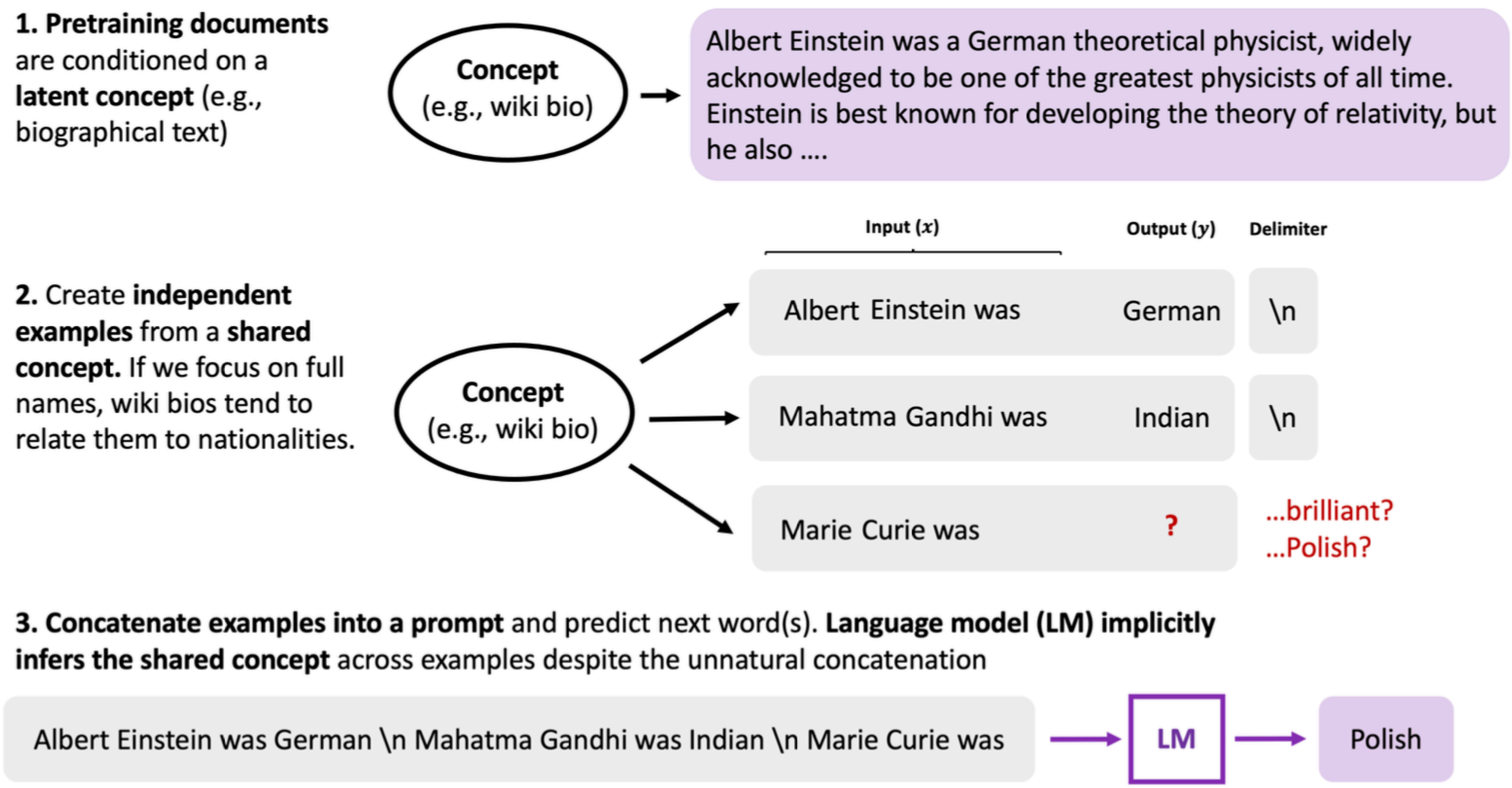
to be covered next

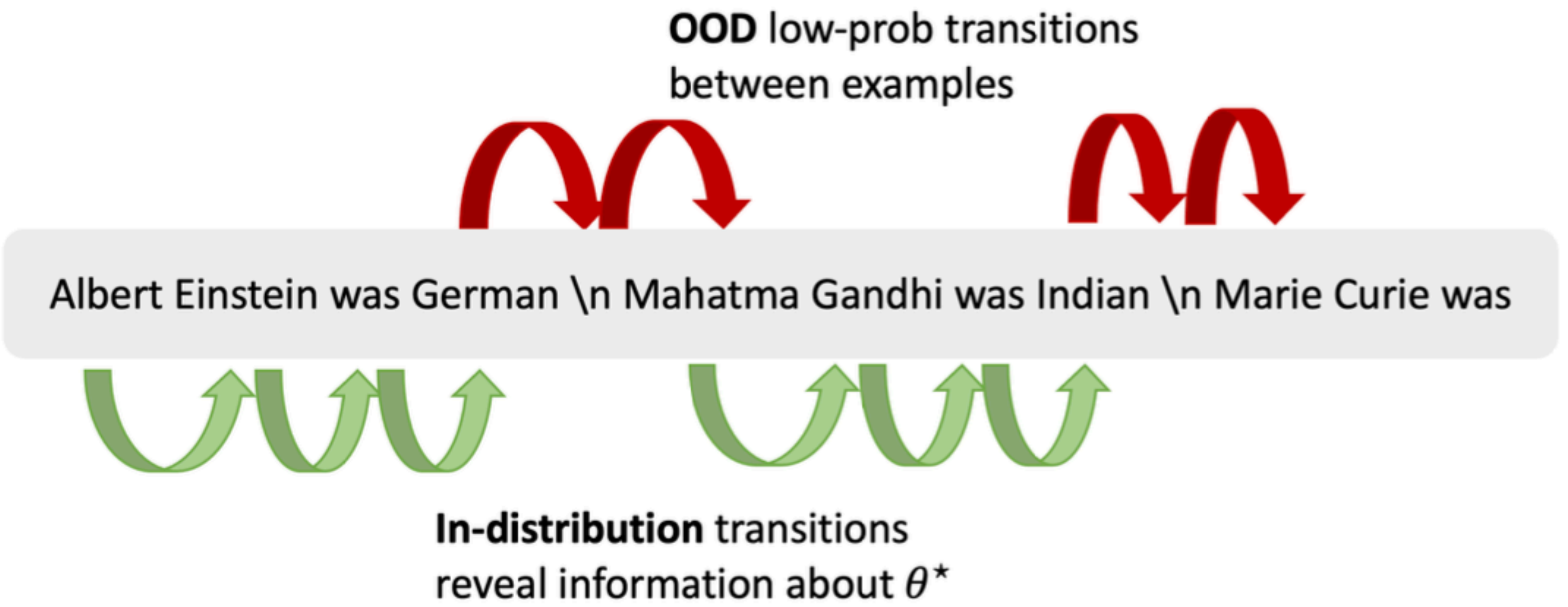
- ICL as implicit Bayesian inference
  - Xie et al. (2022)
- ICL may not be “learning” proper
  - Reynolds & McDonell (2021)
- conditions influencing ICL performance
  - Min et al. (2022)
- ICL without “prompt understanding”
  - Webson & Pavlick (2022)
- ICL improves with additional explanations
  - Lampinen et al. (2022)

In-context learning as implicit Bayesian inference

Xie et al. (2022)

- starting point:
  - training data is generated by a **hierarchical** stochastic process
    - this is a more commonly expressed idea
- observation:
  - k-shot input is out-of-distribution
  - high within-example probability
  - low between-example probability
- conjecture:
  - in-context learning is inference of the **latent type** shared by demonstrations



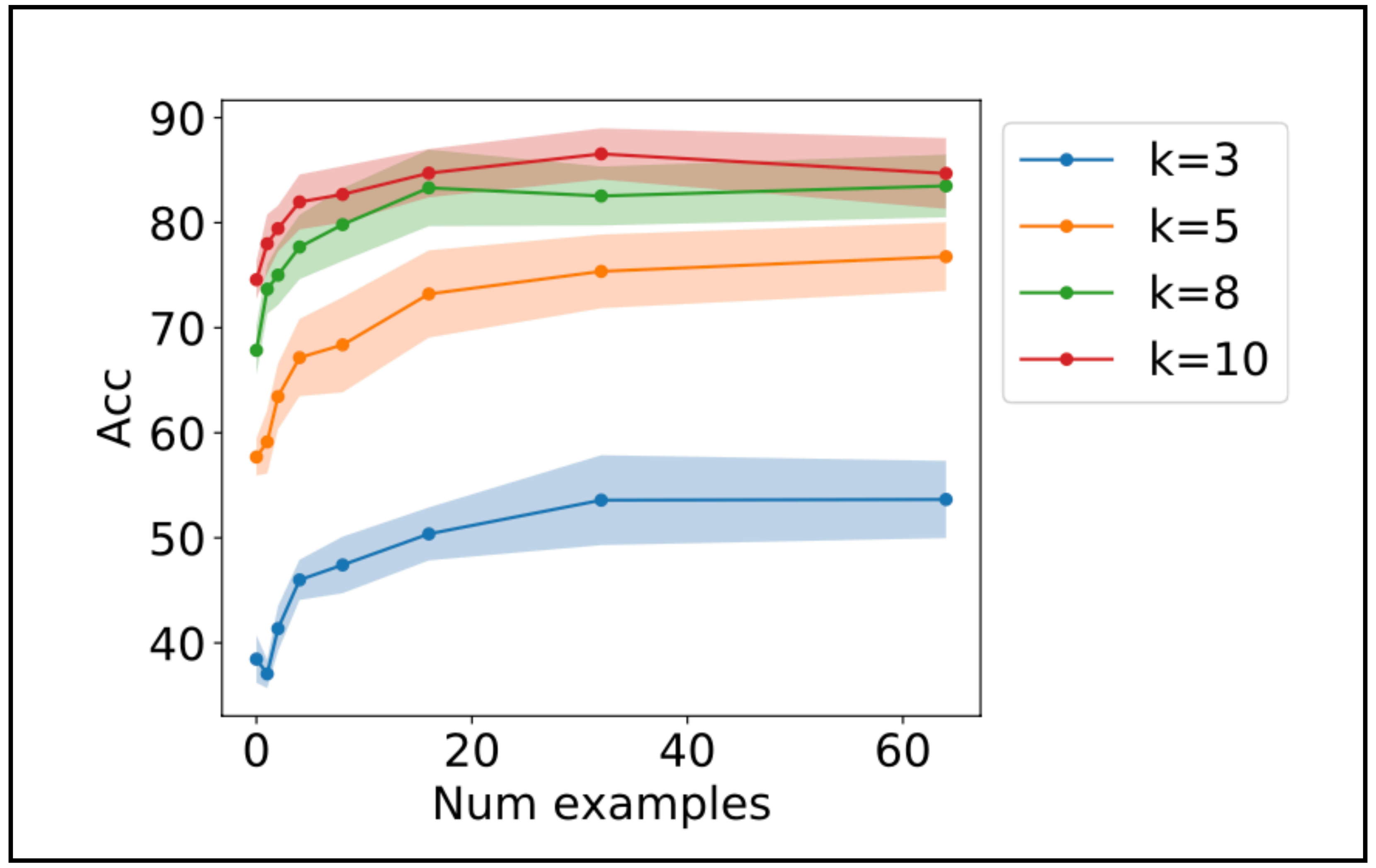


In-context learning as implicit Bayesian inference

Xie et al. (2022)

- theoretical argument:
  - proof that recovering the latent document type from k-shot examples is theoretically possible
- simulation results:
  - generate synthetic training data from a **hierarchical HMM process**
  - train autoregressive LSTM and transformer models (small-ish)
- results:
  - both LSTMs and transformers show **emerging in-context learning ability**
  - reproduces known effects:
    - more examples ⇒ better performance
    - example ordering matters
  - larger models ⇒ better in-context learning performance at equal predictive accuracy on train/test data

effect of demonstration size



Towards "prompt engineering"

Reynolds & McDonell (2021)

- contra claim "k-shot learning = meta-learning" *sensu strict*
- result: effect of examples can be retrieved by proper prompt
- interpretation: task ability comes from pre-training; prompt guides the system towards the "task location"

simple colon prompt

```
French: example_source_phrase
English: example_target_phrase

French: example_source_phrase
English: example_target_phrase

[...]

French: source_phrase
English:
```

master translator prompt

```
A French phrase is provided: source_phrase
The masterful French translator flawlessly
translates the phrase into English:
```

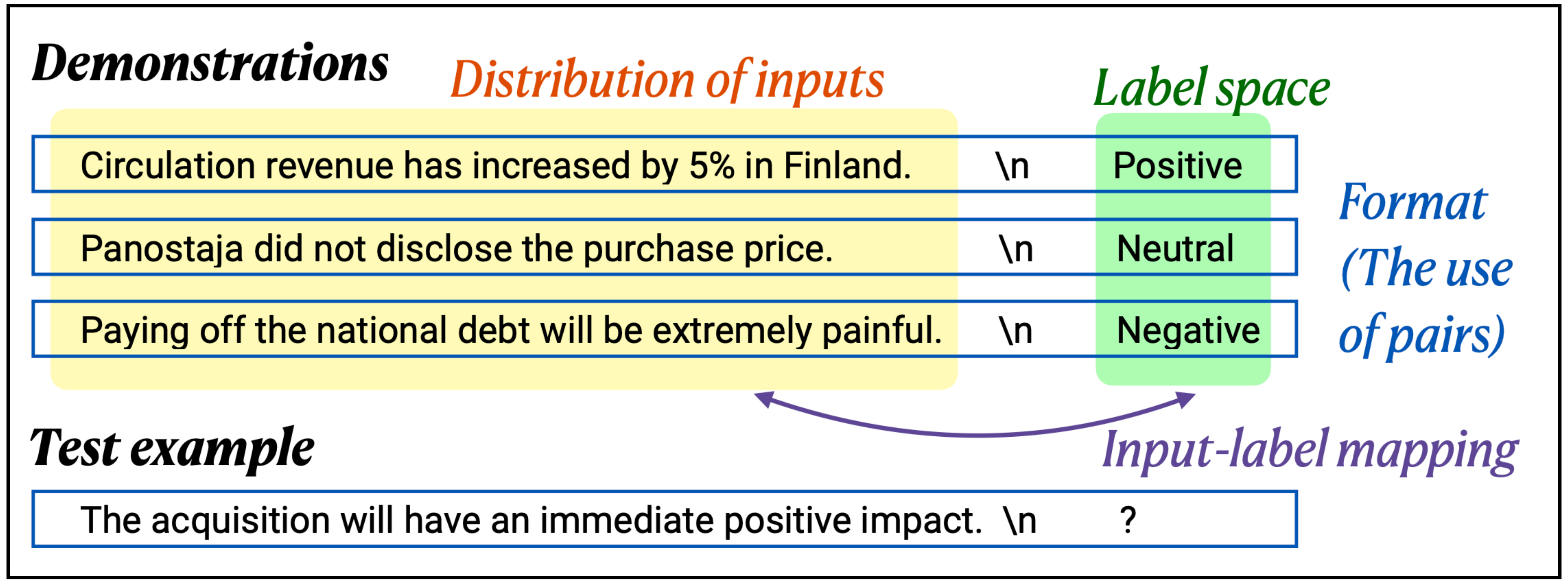
results

Prompt	Babbage / 6.7B	Curie / 13B
OpenAI 0-shot	15.5	22.4
OpenAI 1-shot	31.6	31.4
OpenAI 64-shot	36.4	38.3
Reproduced OpenAI 0-shot	15.9	18.7
Reproduced OpenAI 1-shot	21.8	24.1
Reproduced OpenAI 10-shot	25.1	27.9
Simple colon 0-shot	23.5	33.3
Simple colon 1-shot	18.0	27.6
Simple colon 10-shot	24.1	33.4
Master translator 0-shot	26.5	32.9

What makes in-context learning work?

Min et al. (2022)

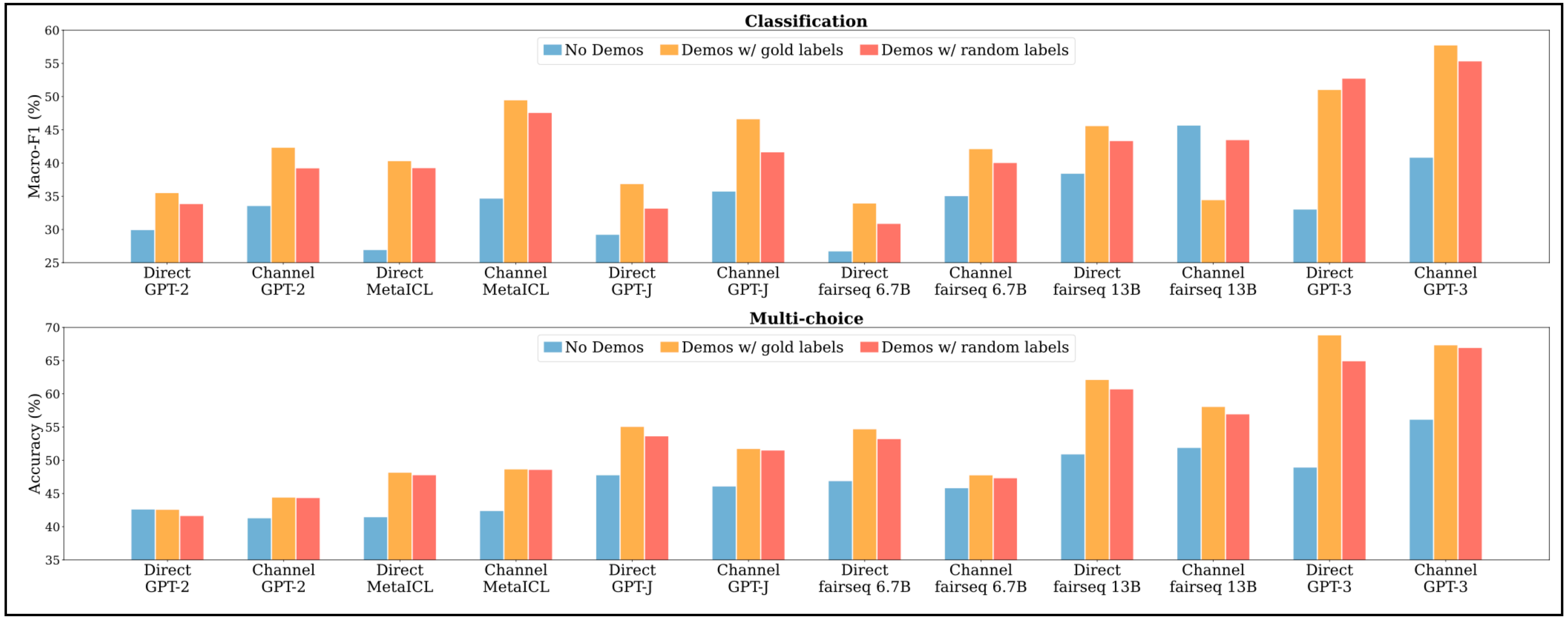
- empirically test effect of four factors:
  - **correctness** of input-label mapping
  - **similarity** of examples and target input
  - **coverage** of label space in examples
  - **format** of input-label examples
- main results:
  - input-label correctness matters little
  - other factors matter more



What makes in-context learning work?

Min et al. (2022)

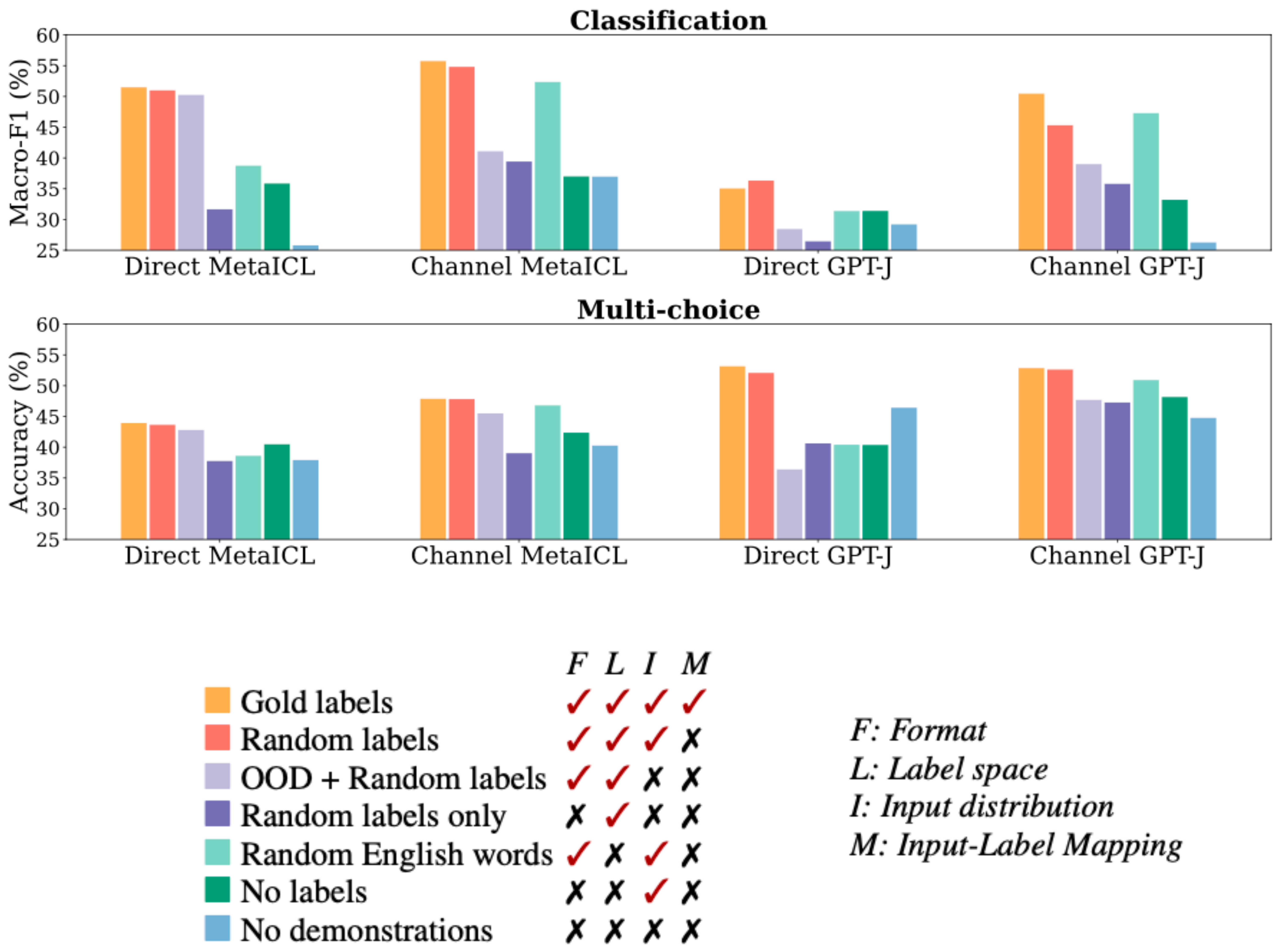
- methods:
  - 6 LLMs * 2 inference methods
    - normal & noisy channel
  - 26 common tasks
    - classification & multiple choice
  - different prompting methods
    - zero-shot
    - k-shot w/ correct labels
    - k-shot w/ scrambled labels



What makes in-context learning work?

Min et al. (2022)

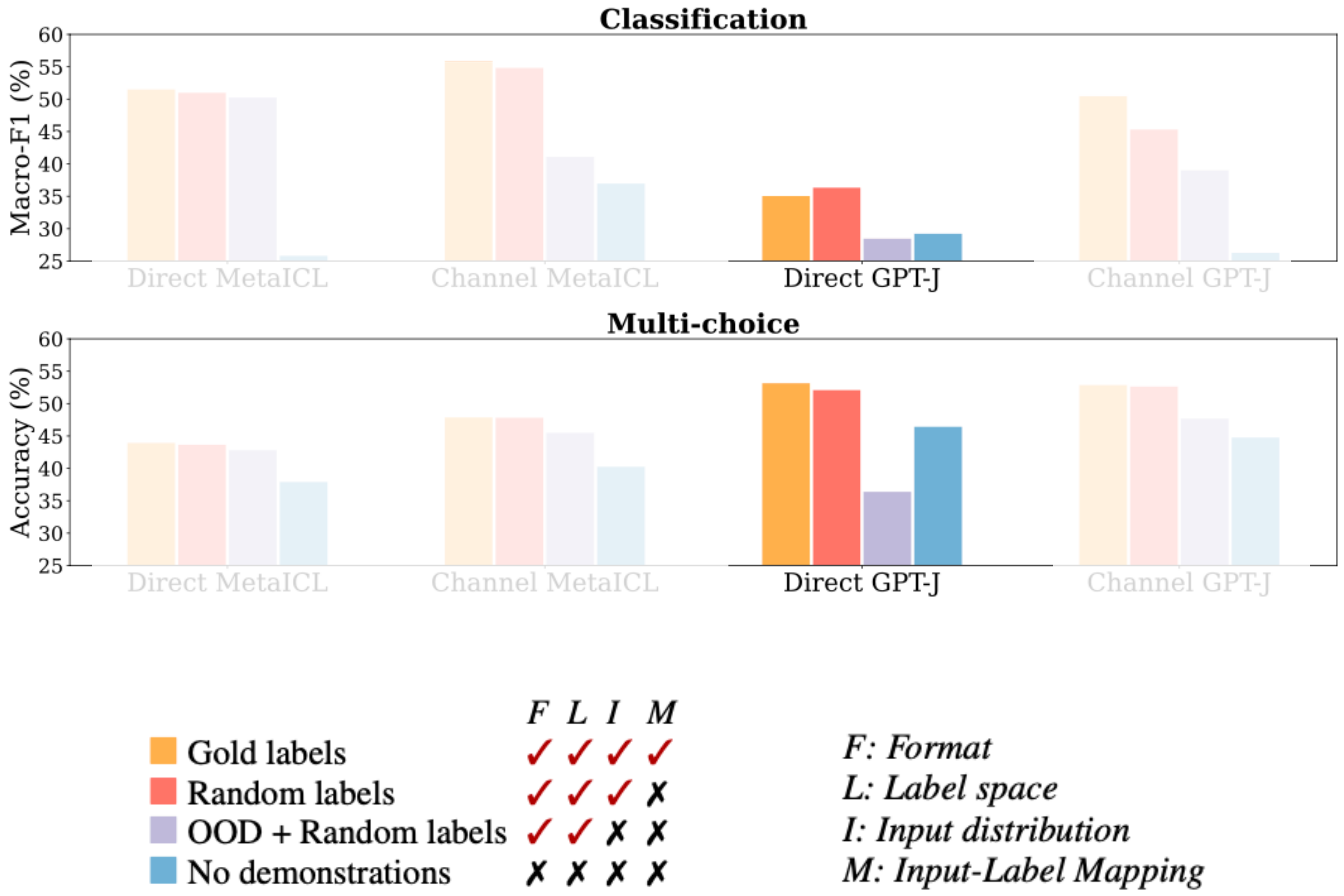
- methods:
  - different prompting methods
    - zero-shot
    - k-shot w/ correct labels
    - k-shot w/ scrambled labels
    - k-shot w/ nonsense labels
    - k-shot w/ out-of-distribution inputs
    - k-shot w/ only input but no labels
    - k-shot w/ only labels but no input



What makes in-context learning work?

Min et al. (2022)

- in-distribution inputs matter
  - to standard models
- possible explanation:
  - the more similar the demonstrations are to the target input, the more the task resembles autoregressive text-generation; the task becomes "more similar" to what is seen in the demonstrations
  - unlike "meta-learning" *sensu stricto*



This suggests that in-distribution inputs in the demonstrations substantially contribute to performance gains. This is likely because conditioning on the in-distribution text makes the task closer to language modeling, since the LM always conditioned on the in-distribution text during training.

Do models understand the meaning of the prompts?

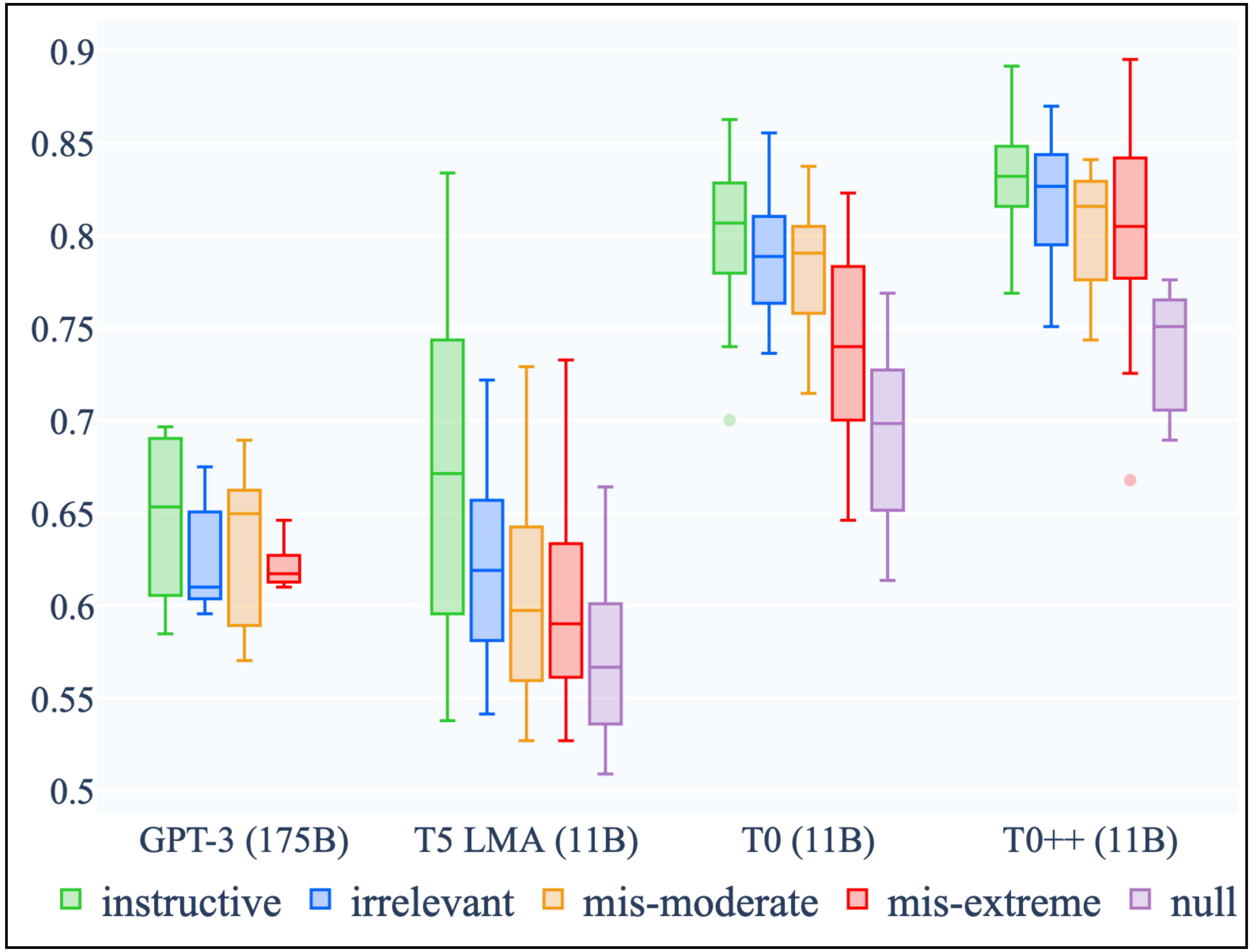
Webson & Pavlick (2022)

- zero-shot, and k-shot in-context learning
  - $k \in \{0,4,8,16,32,64,128,256\}$
- different example templates
  - see →
- different target words
  - yes/no
  - "yes/no"-like (true/false, positive/negative)
  - arbitrary (cat/dog, Jane/Jake)
  - reversed (no/yes)

Category	Examples
instructive	{prem} Are we justified in saying that "{hypo}"? Suppose {prem} Can we infer that "{hypo}"?
misleading-moderate	{prem} Can that be paraphrased as: "{hypo}"? {prem} Are there lots of similar words in "{hypo}"?
misleading-extreme	{prem} is the sentiment positive? {hypo} {prem} is this a sports news? {hypo}
irrelevant	{prem} If bonito flakes boil more than a few seconds the stock becomes too strong. "{hypo}"?
null	{premise} {hypothesis} {hypothesis} {premise}

Do models understand the meaning of the prompts?

Webson & Pavlick (2022)

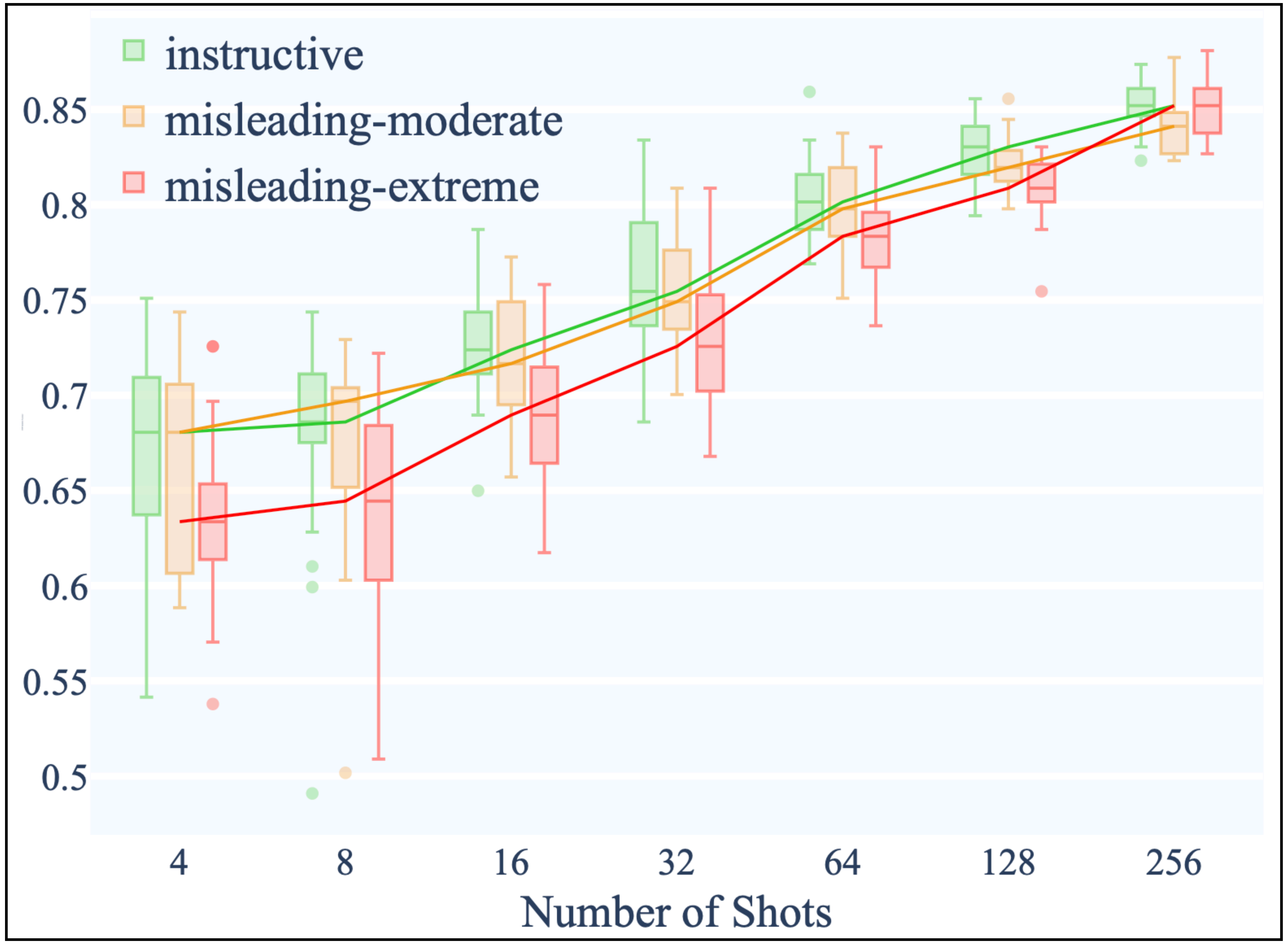


16-shot accuracy: no differences btw. categories except for comparisons against "null"

Category	Examples
instructive	{prem} Are we justified in saying that "{hypo}"? Suppose {prem} Can we infer that "{hypo}"?
misleading-moderate	{prem} Can that be paraphrased as: "{hypo}"? {prem} Are there lots of similar words in "{hypo}"?
misleading-extreme	{prem} is the sentiment positive? {hypo} {prem} is this a sports news? {hypo}
irrelevant	{prem} If bonito flakes boil more than a few seconds the stock becomes too strong. "{hypo}"?
null	{premise} {hypothesis} {hypothesis} {premise}

Do models understand the meaning of the prompts?

Webson & Pavlick (2022)



performance of T0 (3B): only difference btw. "misleading extreme" for 8 to 128 shots

Category	Examples
instructive	{prem} Are we justified in saying that "{hypo}"? Suppose {prem} Can we infer that "{hypo}"?
misleading-moderate	{prem} Can that be paraphrased as: "{hypo}"? {prem} Are there lots of similar words in "{hypo}"?
misleading-extreme	{prem} is the sentiment positive? {hypo} {prem} is this a sports news? {hypo}
irrelevant	{prem} If bonito flakes boil more than a few seconds the stock becomes too strong. "{hypo}"?
null	{premise} {hypothesis} {hypothesis} {premise}

Do models understand the meaning of the prompts?

Webson & Pavlick (2022)

- different target words
  - yes/no
  - “yes/no”-like (true/false, positive/negative)
  - arbitrary (cat/dog, Jane/Jake)
  - reversed (no/yes)
- target word matters for average success
- but unintuitive interactions with:
  - the former can outperform the latter
    - {premise} Does the paragraph start with “the"? {hypothesis} [yes/no]
    - {premise} Based on the previous passage, is it true that "{hypothesis}"? [cat/dog]

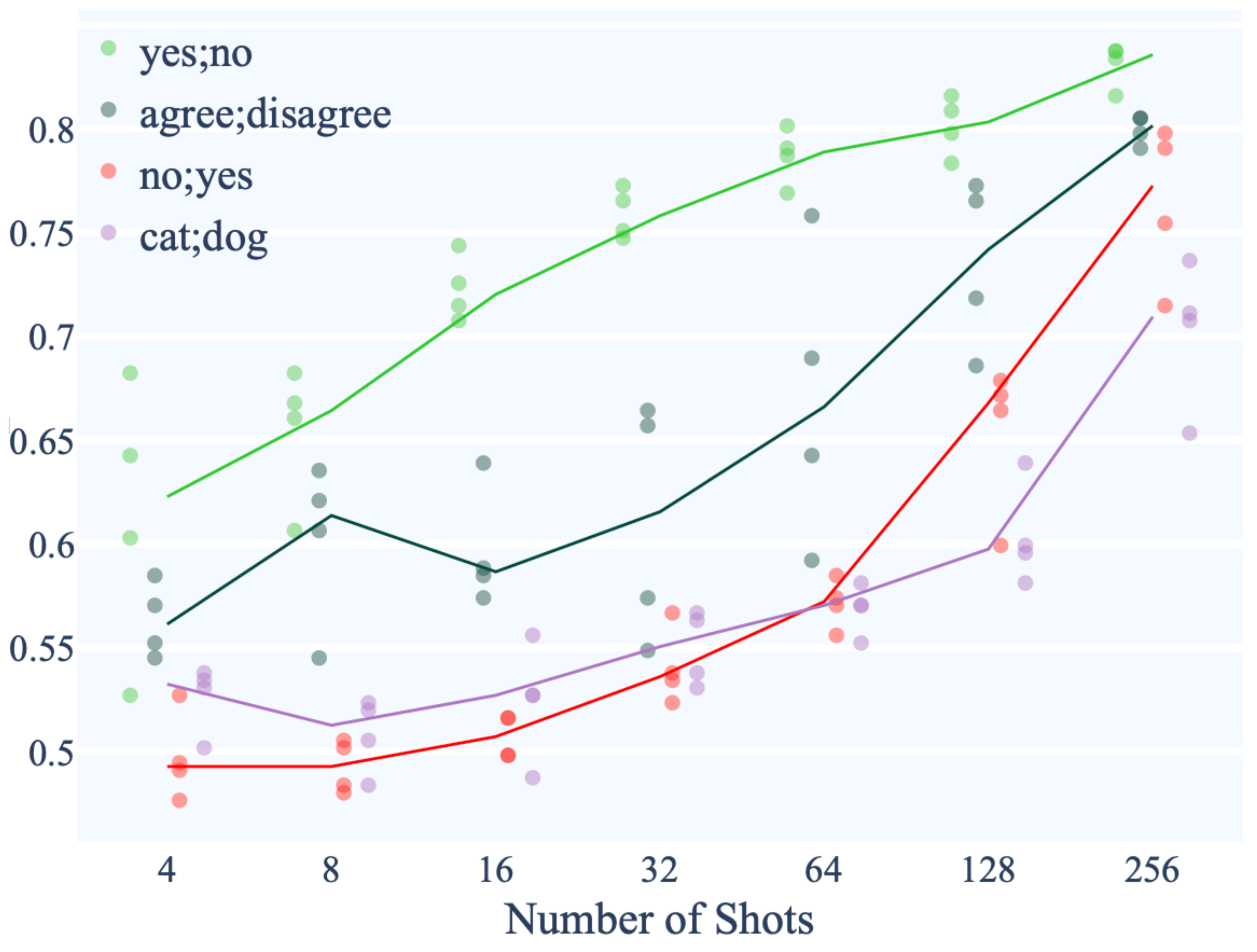
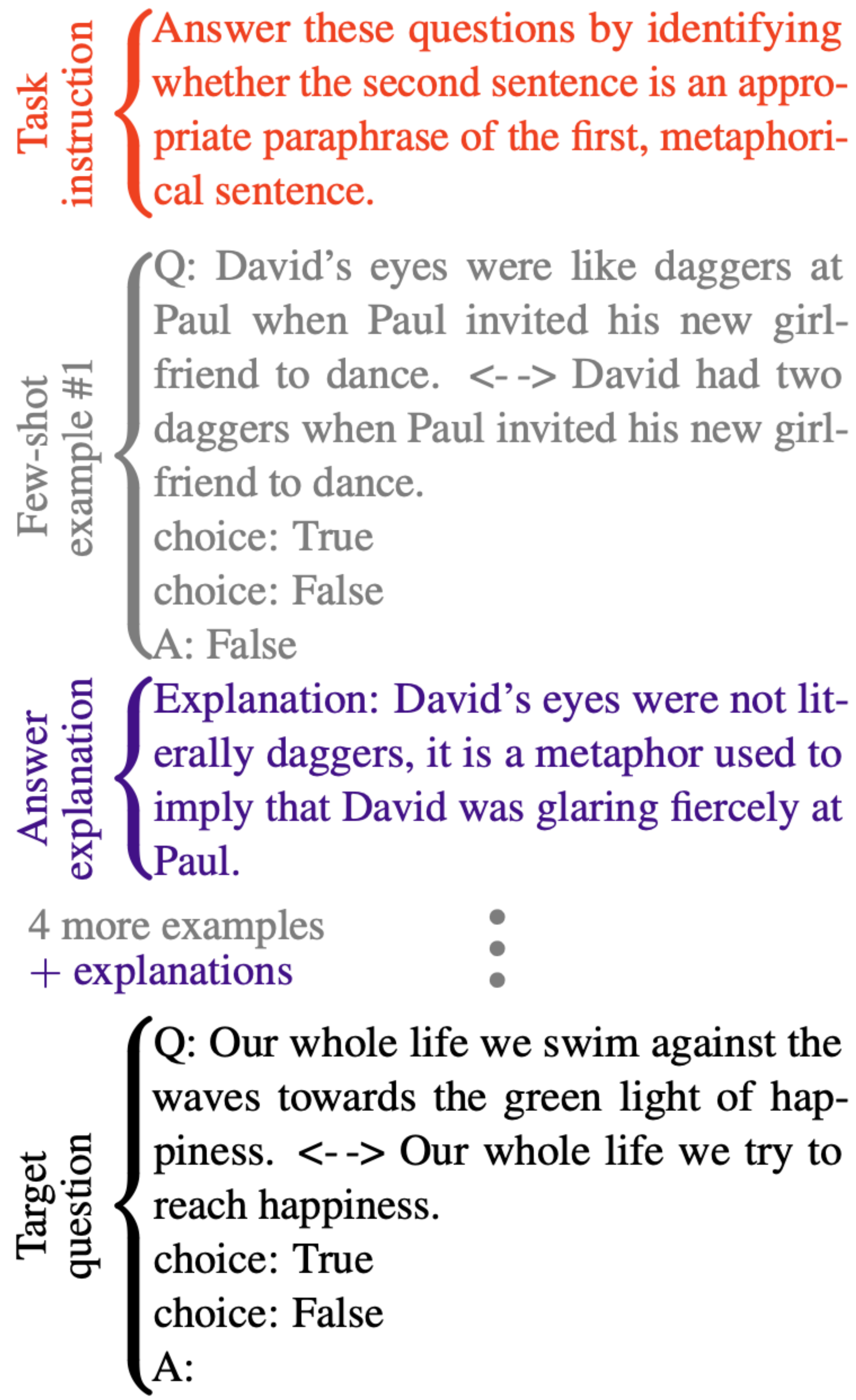


Figure 5: The best-performing instructive template for ALBERT on RTE, `{prem} Are we justified in saying that "{hypo}"?` with select LM targets from each category.

Can LMs learn from explanations in context?

Lampinen et al. (2022)

- method:
  - supplement ICL with additional examples
- main findings:
  - explanations **can** further improve ICL performance
  - hand-tuned explanations work better
  - benefit only for larger LMs



Summary

In-context learning

- structure of simple prompt also matters; so ICL need not be genuine "learning"
  - Reynolds & McDonell (2021)
- k-shot examples may provide statistical cues which the LM can exploit
  - Xie et al. (2022)
- similarity of demonstrations and target matters
  - Min et al. (2022)
- instructions and label correctness do not matter
  - Min et al. (2022), Webson & Pavlick (2022)
- still, good explanations can help (for large LMs)
  - Lampinen et al. (2022)

From simple to engineered prompting

Counting letters

Exploring step-by-step reasoning

INPUT

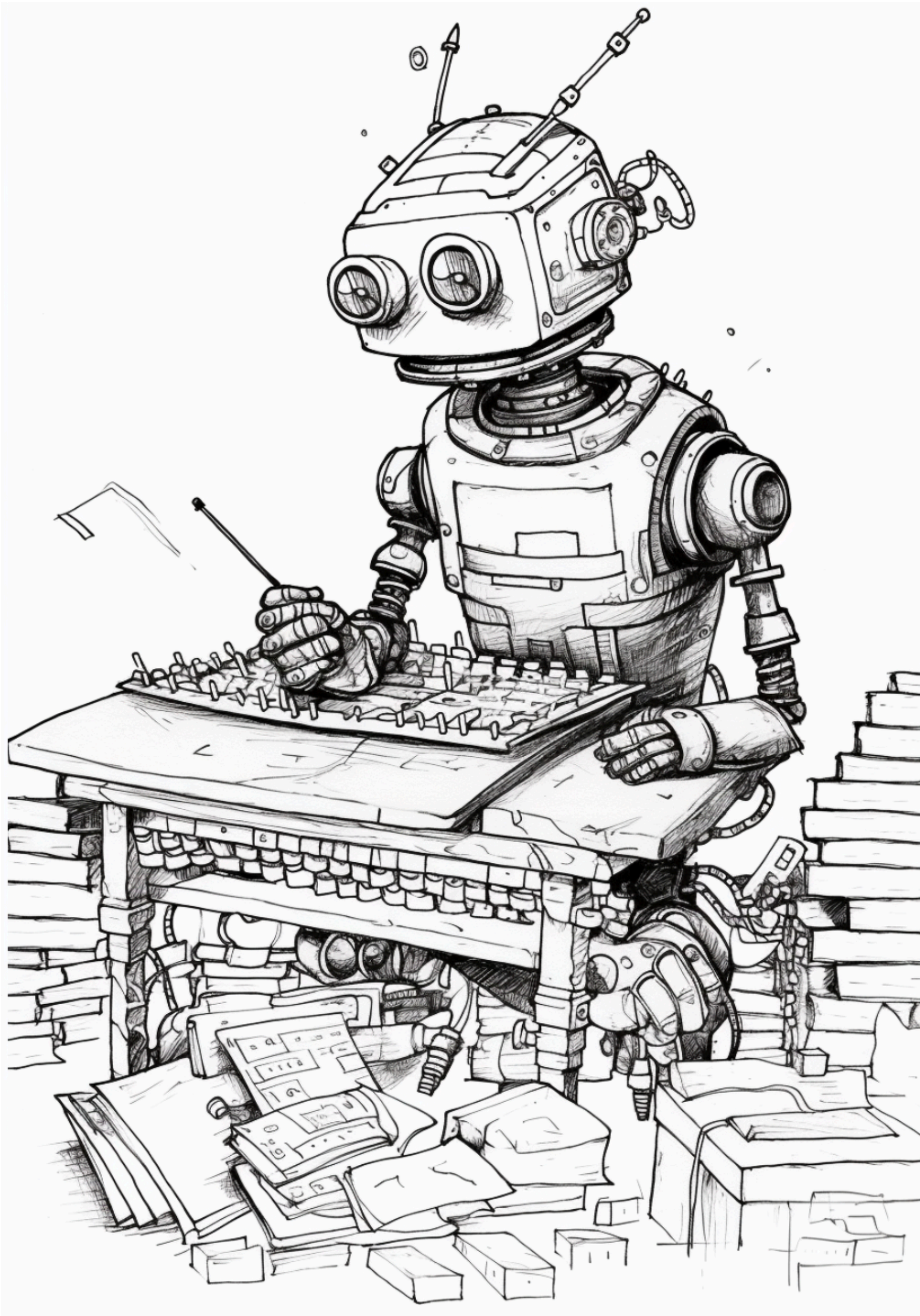
```
Do the numbers of letters in all words starting with a
vowel from the following list sum up to 42?

Polina, Michael, eggplant, cheese, oyster, imagination,
elucidation, induce

Please answer just 'yes' or 'no'
```

OUTPUT

```
>>> ???
```



demo

Structured reasoning

GPT-3.5 turbo (Dec 2023)

Do the numbers of letters in all words starting with a vowel from the following list sum up to 42?

Polina, Michael, eggplant, cheese, oyster, imagination, elucidation, induce

Please answer just 'yes' or 'no'

No

No = 98.42%

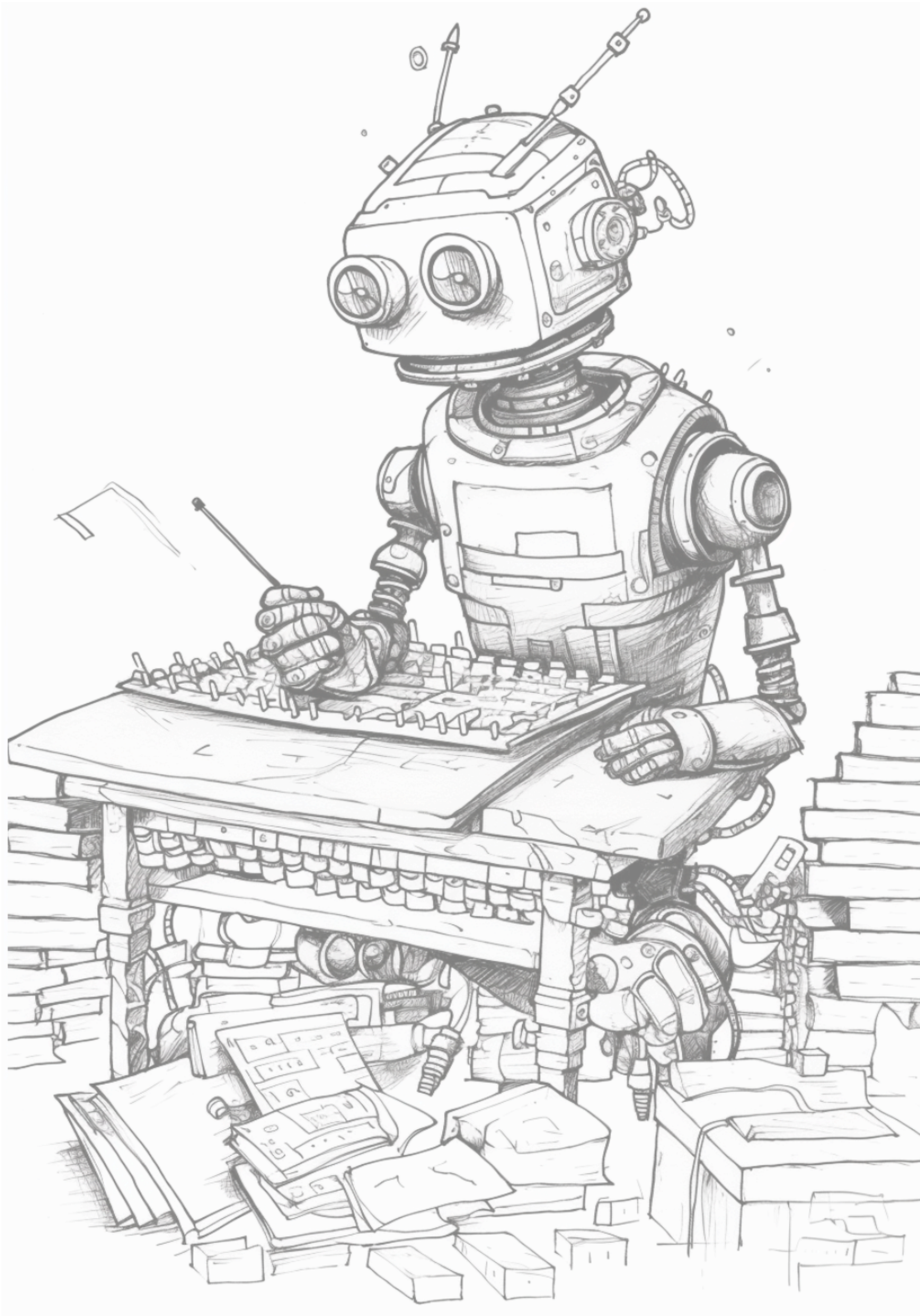
Yes = 1.57%

NO = 0.01%

no = 0.01%

No = 0.00%

Total: -0.02 logprob on 1 tokens
(100.00% probability covered in top 5 logits)



Structured reasoning

GPT-3.5 turbo (Dec 2023)

Do the numbers of letters in all words starting with a vowel from the following list sum up to 42?

Polina, Michael, eggplant, cheese, oyster, imagination, elucidation, induce

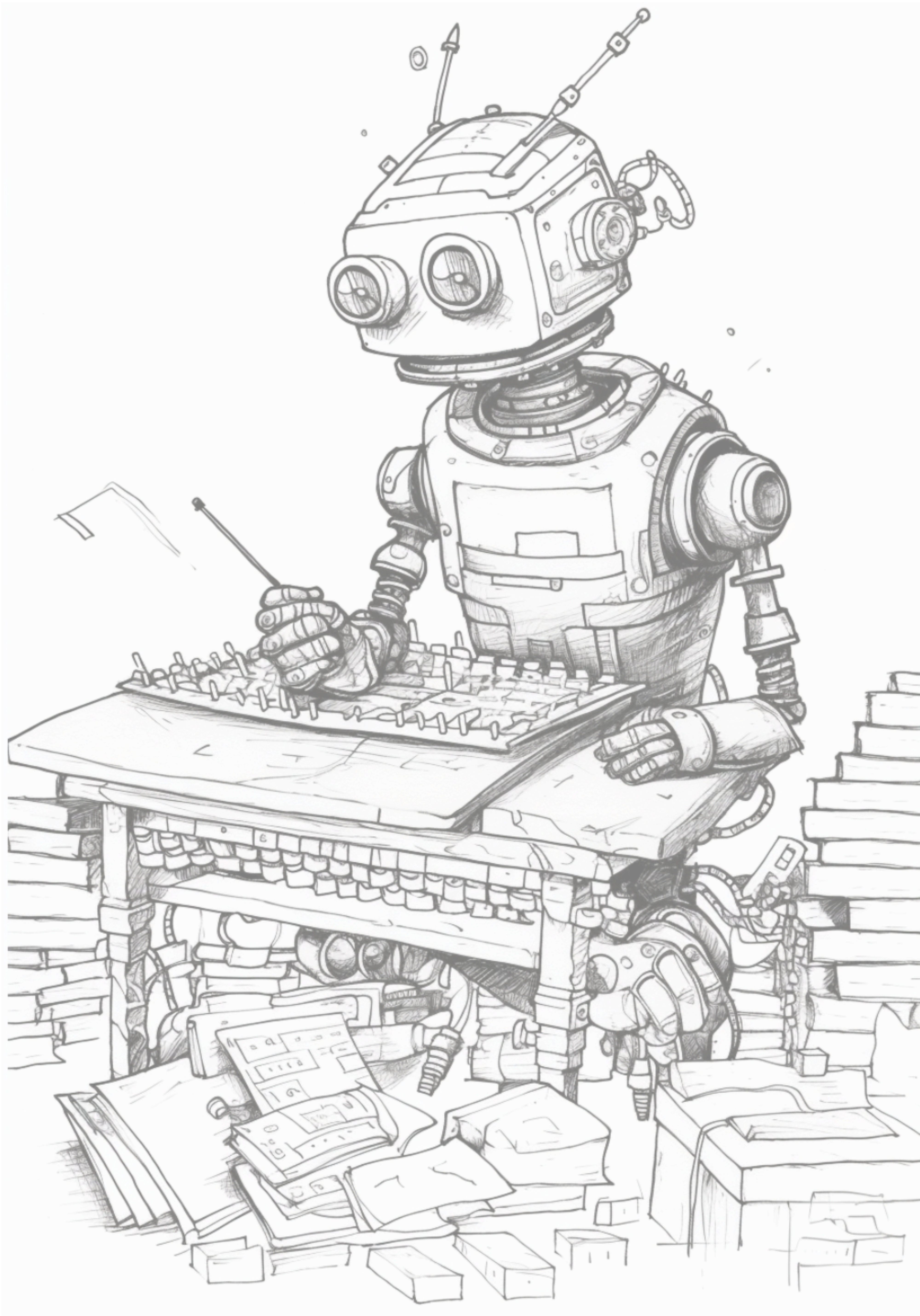
Let's think step by step. Correctness is very important here.

We can quickly observe that Polina, eggplant, oyster, and imagination all start with a vowel.

Letters in Polina: 6
Letters in eggplant: 8
Letters in oyster: 6
Letters in imagination: 11

6 + 8 + 6 + 11 = 31

The answer is no, the numbers of letters in all words starting with a vowel from the list do not sum up to 42.



Structured reasoning

ChatGPT (Dec 2023)

MC **You**

Do the numbers of letters in all words starting with a vowel from the following list sum up to 42?

Polina, Michael, eggplant, cheese, oyster, imagination, elucidation, induce

Please answer just 'yes' or 'no'

ChatGPT

No.

MC **You**

Are you sure?

ChatGPT

Yes, the numbers of letters in all words starting with a vowel from the list do sum up to 42. [>-]

Analysis

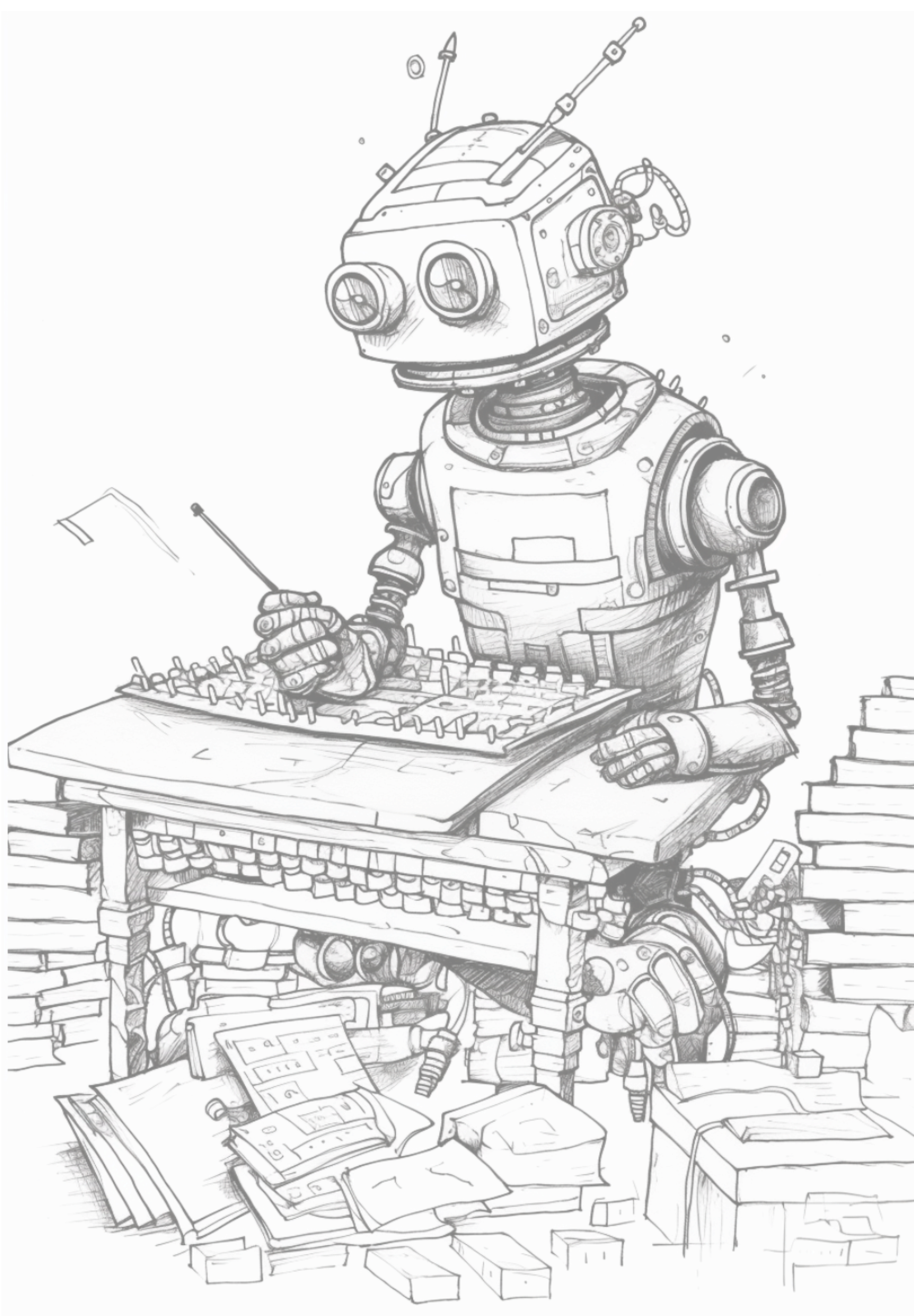
python Copy code

```python
# List of words starting with a vowel
words = ["eggplant", "oyster", "imagination", "elucidation'

# Calculating the sum of the number of letters in each worc
total_letters = sum(len(word) for word in words)
total_letters == 42
```

Result

True



Chain-of-Thought prompting

- give task instruction
- give one or more examples with explicit chain-of-thought reasoning leading to the correct answer
- works for example to complex for few-shot prompting
- requires “right” task analysis in CoT steps

INPUT

```
The odd numbers in this group add up to an even number:
4, 8, 9, 15, 12, 2, 1.

A: Adding all the odd numbers (9, 15, 1) gives 25. The
answer is False.

The odd numbers in this group add up to an even number:
15, 32, 5, 13, 82, 7, 1.

A:
```

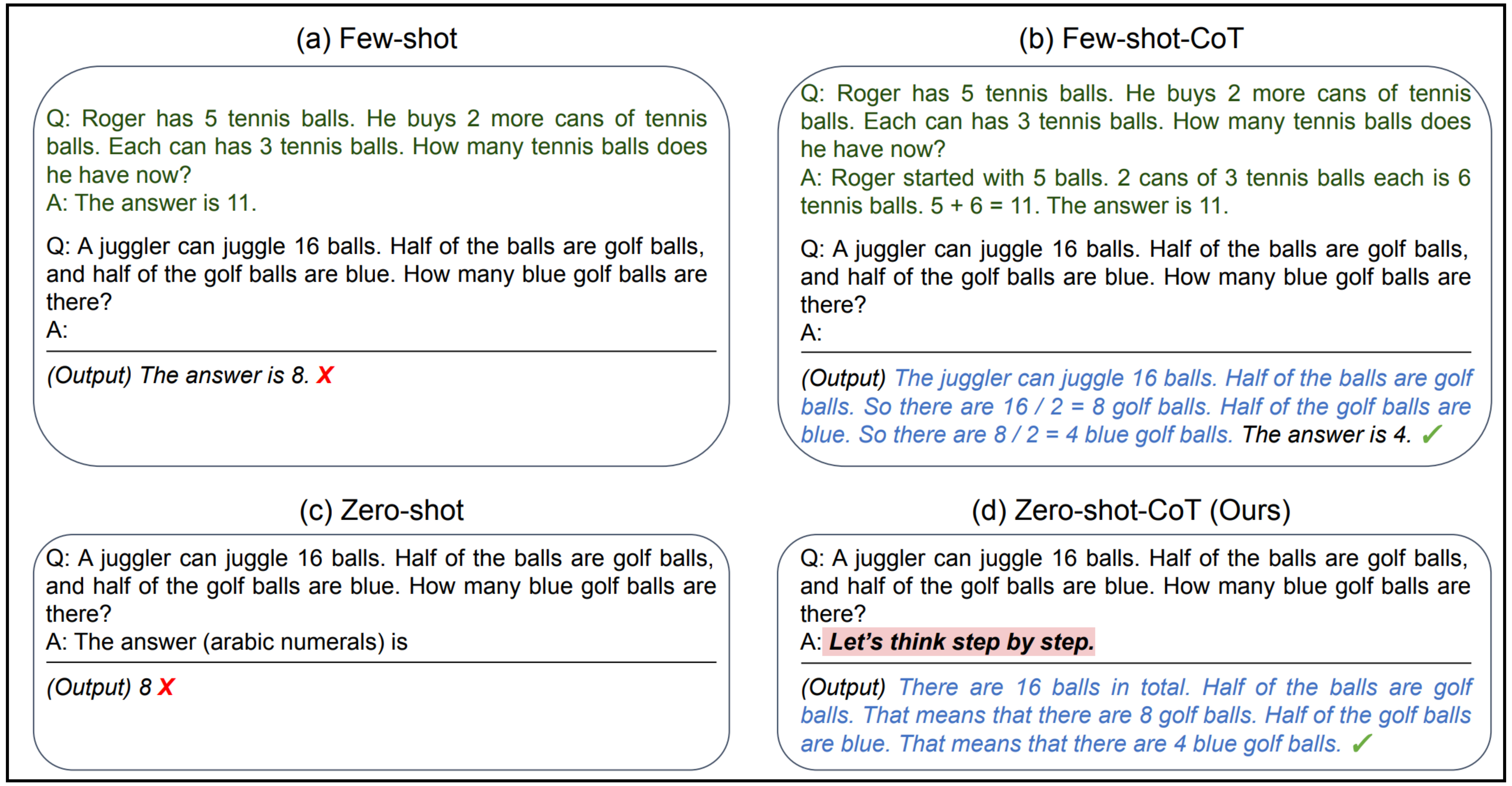
OUTPUT

```
Adding all the odd numbers (15, 5, 13, 7, 1) gives 41.
The answer is False.
```


Wei et al. (2022)

Zero-shot CoT prompting

- just add “Let’s think step by step”
  - even better (Zhou et al. 2022): "Let's work this out in a step by step way to be sure we have the right answer.”



INPUT

```
The odd numbers in this group add up to an even number:
15, 32, 5, 13, 82, 7, 1.

A: Let’s think step by step.
```

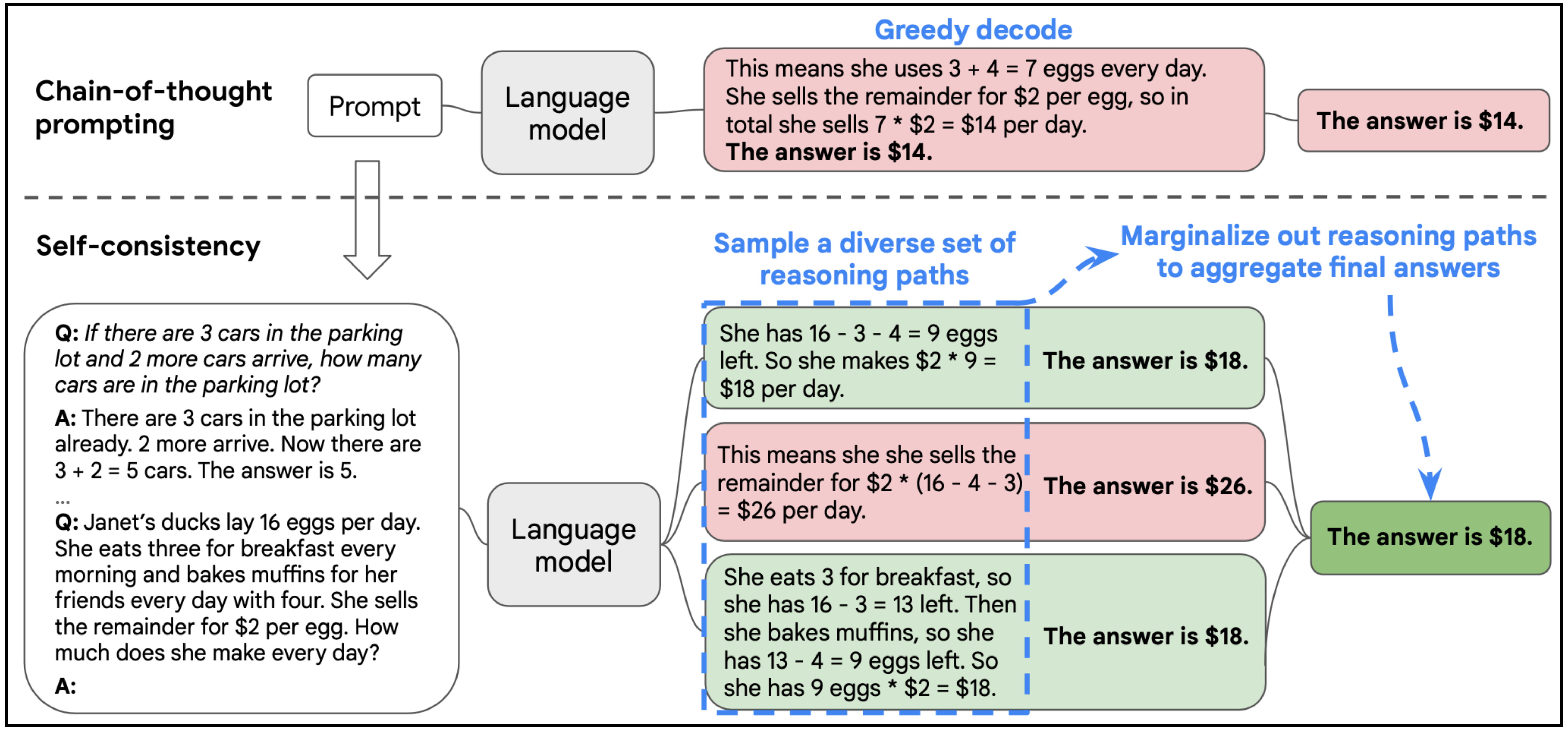
OUTPUT

```
Adding all the odd numbers (15, 5, 13, 7, 1) gives 41.
The answer is False.
```


Kojima et al. (2022)

Self-consistency prompting

- few-shot CoT prompting with self-generated CoT sequences (greedily)
- aggregation over stochastic answer generation

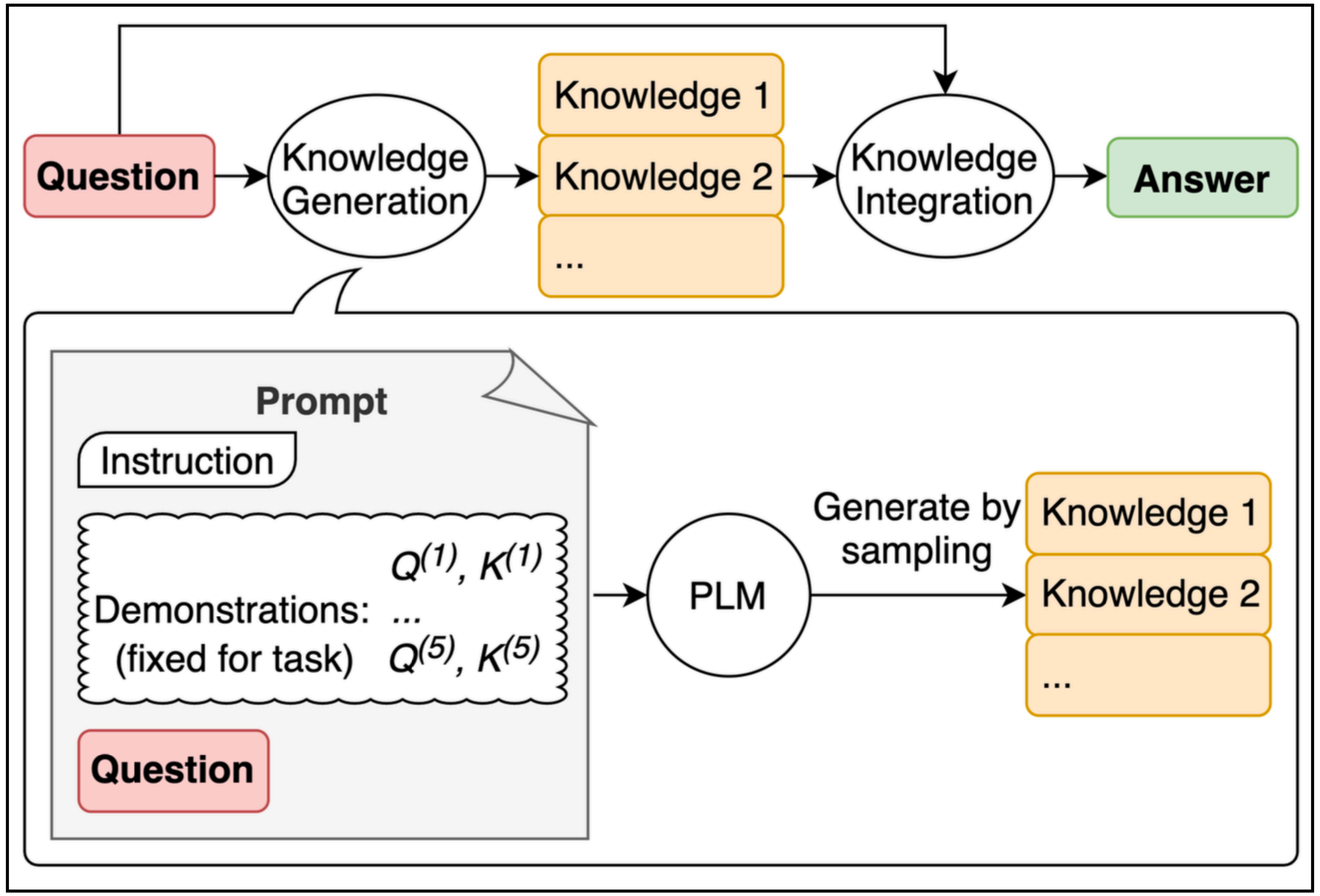


Wang et al. (2022)

Generated knowledge prompting

for common sense QA

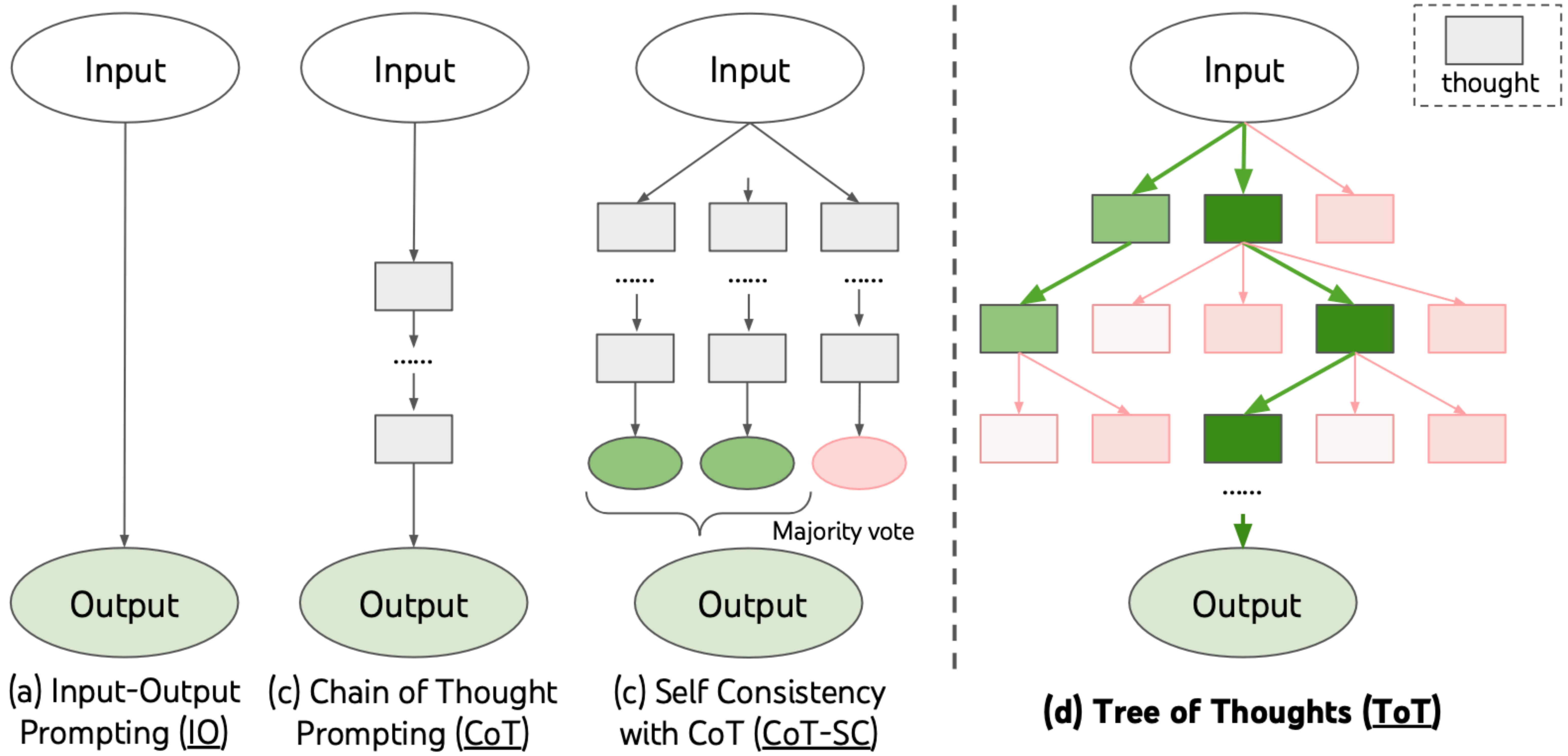
- generate knowledge statements K for question Q
- generate several answers A's to Q for each K
- final answer to Q is max of weighted A's



Liu et al. (2022)

Tree of Thought

deliberate problem solving with LLMs



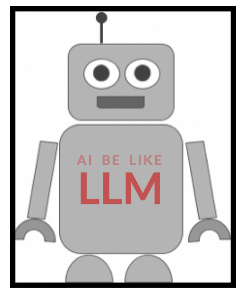
Yao et al. (2023)

Tree of Thought

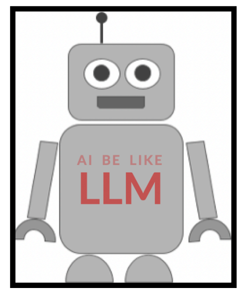
modules of reasoning

- thought decomposition
  - what counts as a next step?

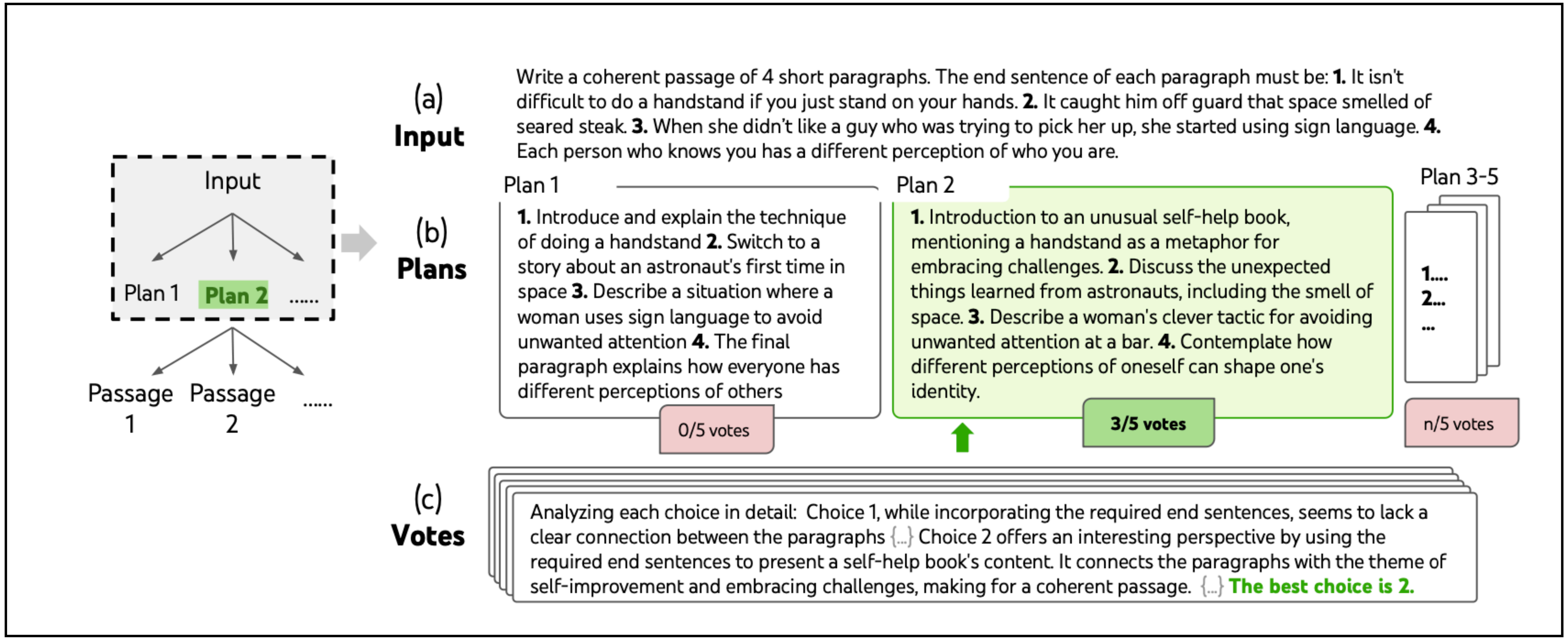
- thought generation
  - sample / generate next steps



- state evaluation
  - evaluate / vote states based on closeness to goal



- search algorithm
  - how to build / traverse the tree



Creative writing task

Yao et al. (2023)

Summary

prompt engineering

- different kinds of prompting techniques:
  - zero-shot w/ or w/o CoT
  - few-shot w/ or w/o CoT
  - ensemble methods
    - self-consistency
    - generated knowledge prompting
  - task-composition / neuro-symbolic generation:
    - tree of thoughts