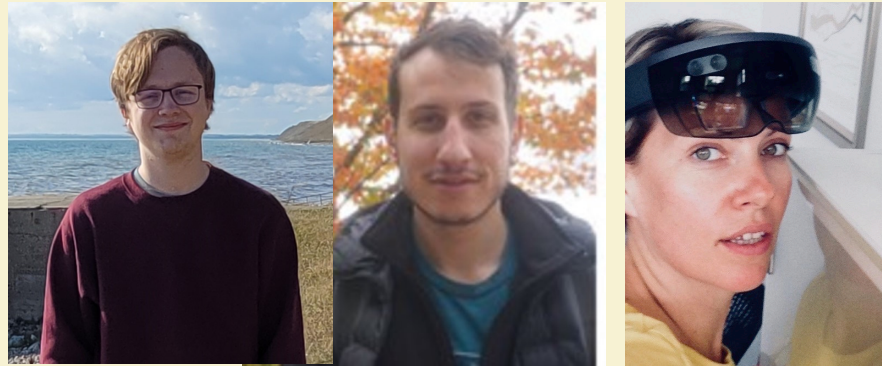


# CS 229br: Foundations of Deep Learning

## Lecture 8: Test time compute

Boaz Barak

Gustaf Ahdritz Gal Kaplun Zona Kostic



# Announcements

Project progress report – April 17

Progress final report – May 3 (end of reading period)

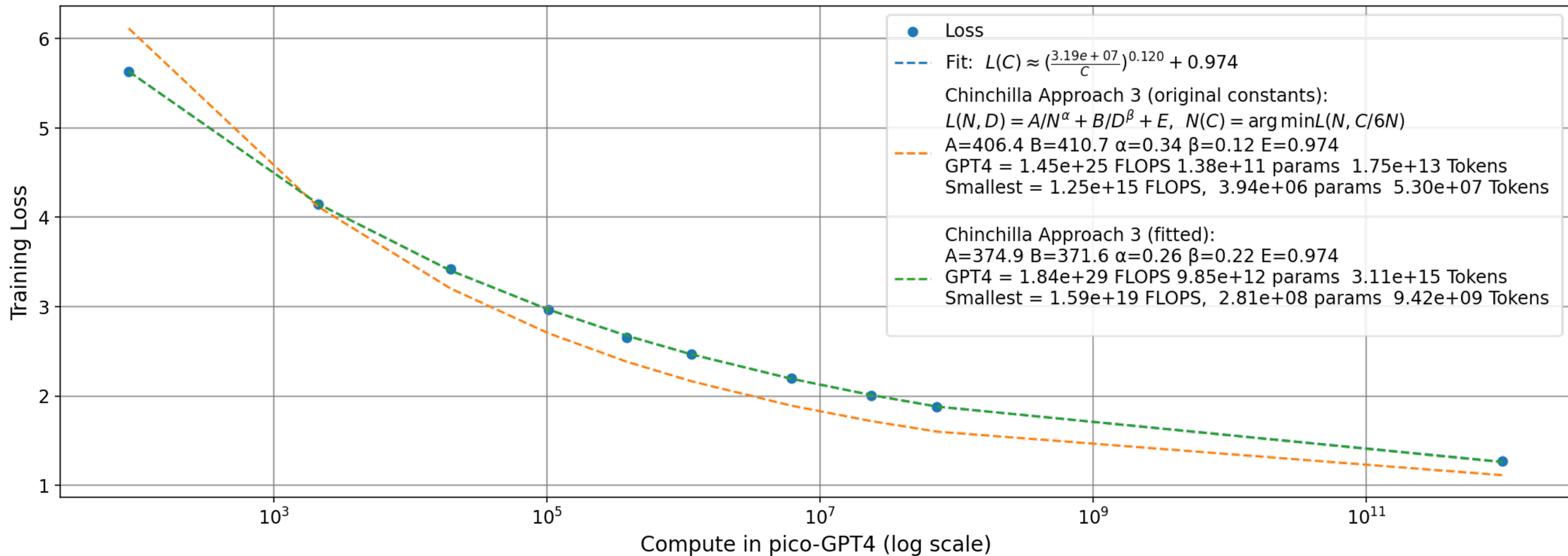
Next lecture (April 6):

AI Safety (including fairness, accountability, transparency)

April 13

Horace He on performance in deep learning training

# Digression: GPT4



# GPTs are GPTs: An Early Look at the Labor Market Impact

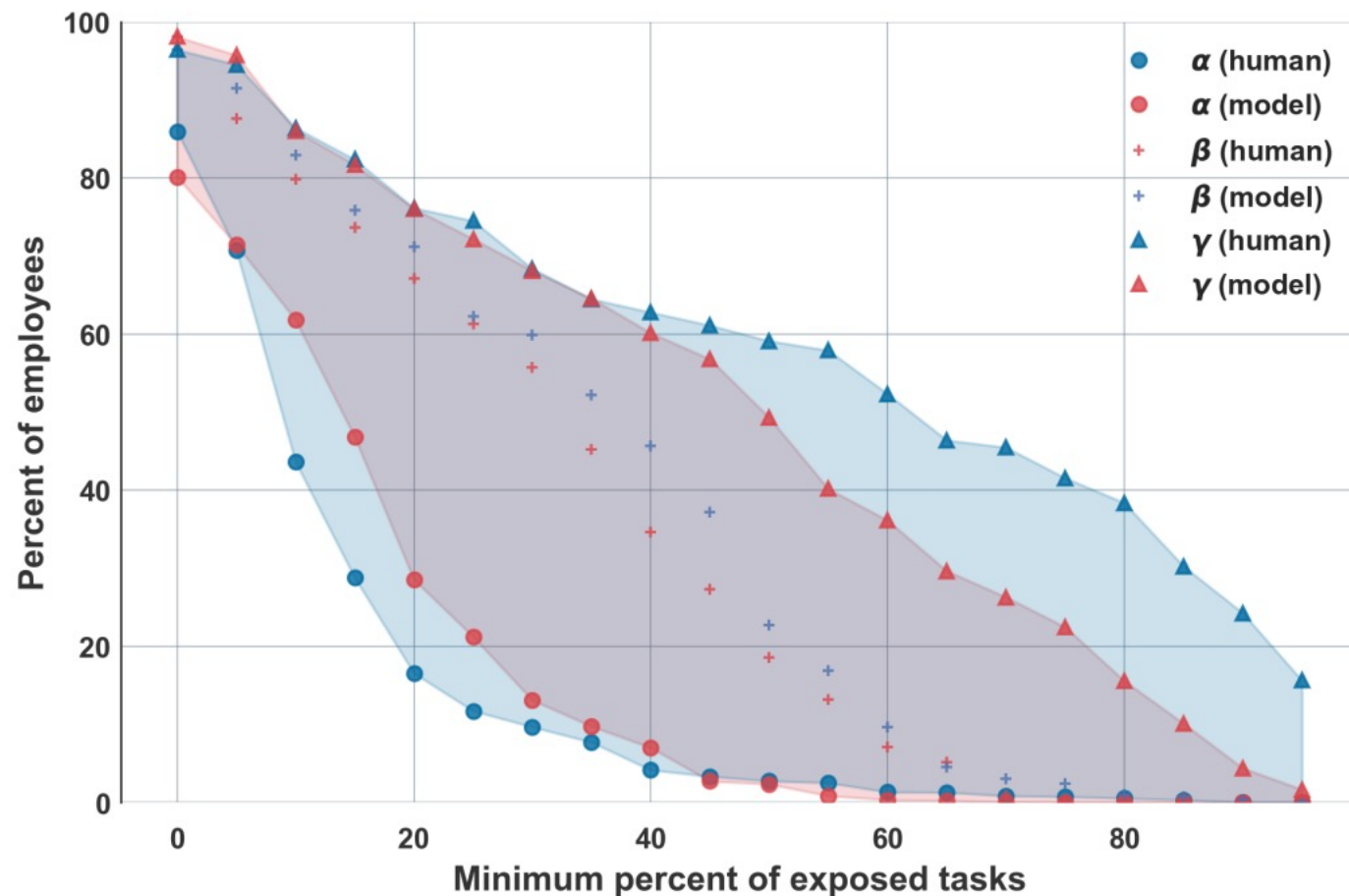
## Potential of Large Language Models

Tyna Eloundou<sup>1</sup>, Sam Manning<sup>1,2</sup>, Pamela Mishkin<sup>\*1</sup>, and Daniel Rock<sup>3</sup>

<sup>1</sup>OpenAI

<sup>2</sup>OpenResearch

<sup>3</sup>University of Pennsylvania



Group	Occupations with highest exposure	% Exposure
<b>Human <math>\alpha</math></b>	Interpreters and Translators	76.5
	Survey Researchers	75.0
	Poets, Lyricists and Creative Writers	68.8
	Animal Scientists	66.7
	Public Relations Specialists	66.7
<b>Human <math>\beta</math></b>	Survey Researchers	84.4
	Writers and Authors	82.5
	Interpreters and Translators	82.4
	Public Relations Specialists	80.6
	Animal Scientists	77.8
<b>Human <math>\zeta</math></b>	Mathematicians	100.0
	Tax Preparers	100.0
	Financial Quantitative Analysts	100.0
	Writers and Authors	100.0
	Web and Digital Interface Designers	100.0
<i>Humans labeled 15 occupations as "fully exposed."</i>		
<b>Model <math>\alpha</math></b>	Mathematicians	100.0
	Correspondence Clerks	95.2
	Blockchain Engineers	94.1
	Court Reporters and Simultaneous Captioners	92.9
	Proofreaders and Copy Markers	90.9
<b>Model <math>\beta</math></b>	Mathematicians	100.0
	Blockchain Engineers	97.1
	Court Reporters and Simultaneous Captioners	96.4
	Proofreaders and Copy Markers	95.5
	Correspondence Clerks	95.2
<b>Model <math>\zeta</math></b>	Accountants and Auditors	100.0
	News Analysts, Reporters, and Journalists	100.0
	Legal Secretaries and Administrative Assistants	100.0
	Clinical Data Managers	100.0
	Climate Change Policy Analysts	100.0
<i>The model labeled 86 occupations as "fully exposed."</i>		
<b>Highest variance</b>	Search Marketing Strategists	14.5
	Graphic Designers	13.4
	Investment Fund Managers	13.0
	Financial Managers	13.0
	Insurance Appraisers, Auto Damage	12.6

Table 4: Occupations with the highest exposure according to each measurement. The final row lists the occupations with the highest  $\sigma^2$  value, indicating that they had the most variability in exposure scores. Exposure percentages indicate the share of an occupation’s task that are exposed to GPTs ( $\alpha$ ) or GPT-powered software ( $\beta$  and  $\zeta$ ), where exposure is defined as driving a reduction in time it takes to complete the task by at least 50% (see exposure rubric [A.1](#)). As such, occupations listed in this table are those where we estimate that GPTs and GPT-powered software are able to save workers a significant amount of time completing a large share of their tasks, but it does not necessarily suggest that their tasks can be fully automated by these



# GPT4

## Sparks of Artificial General Intelligence: Early experiments with GPT-4

Sébastien Bubeck      Varun Chandrasekaran      Ronen Eldan      Johannes Gehrke  
Eric Horvitz      Ece Kamar      Peter Lee      Yin Tat Lee      Yuanzhi Li      Scott Lundberg  
Harsha Nori      Hamid Palangi      Marco Tulio Ribeiro      Yi Zhang

Microsoft Research

**Prompt:** Draw a unicorn in TikZ.

**GPT-4:** [Produces  $\text{\LaTeX}$  compiling to following picture.]

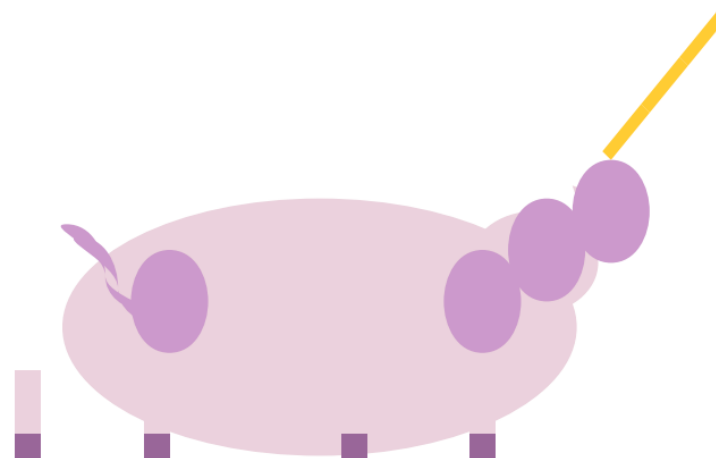
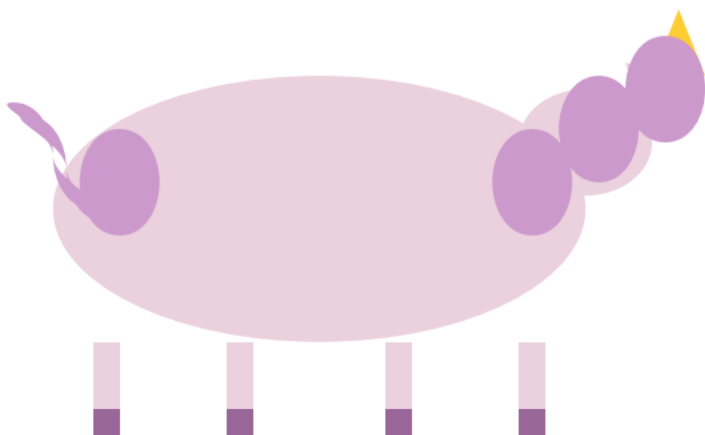


Figure 1.4: We gave to GPT-4 a transformed version of the TikZ code it produced for Figure 1.1, with the part drawing the horn removed. We asked for code to add back the horn, and display the result. This demonstrates that GPT-4 can “see” despite being a pure language model (we emphasize again that the version we test with is **not** multimodal).

## GPT-4

Draw a photo using SVG inspired by the sentence

"A frog hops into a bank and asks the teller, 'Do you have any free lily pads?' The teller responds, 'No, but we do offer low interest loans for pond upgrades.'"

You are not allowed to use external images, textures but only basic shapes.

Can you add more details?

Add a roof to the bank

Make sure all the text on the top layer, it is covered by some object now.

Add 2 wheels to the car

Make the grass red

Add a woman on the right of the teller

Add a big red frog on the top left of the green frog.

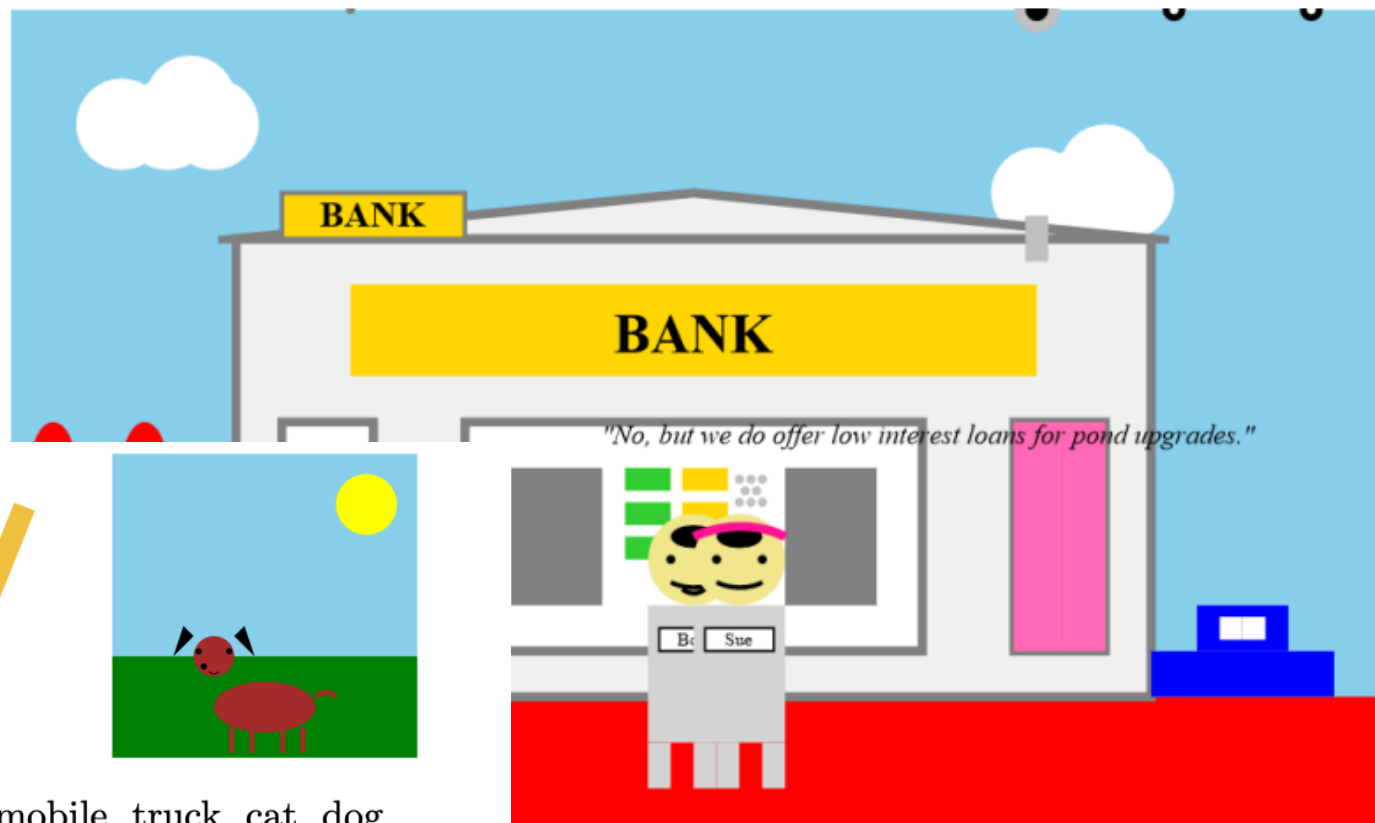
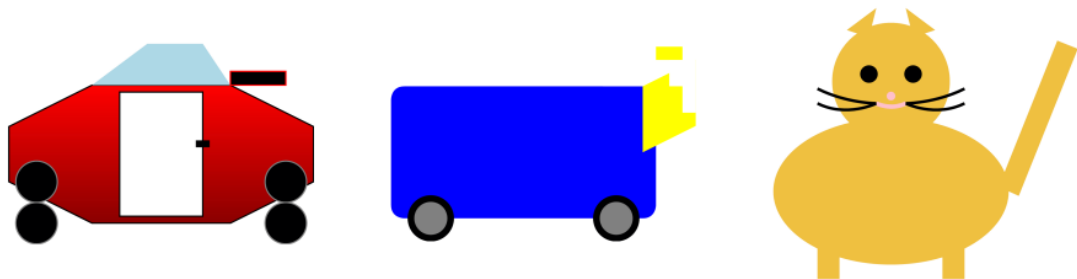
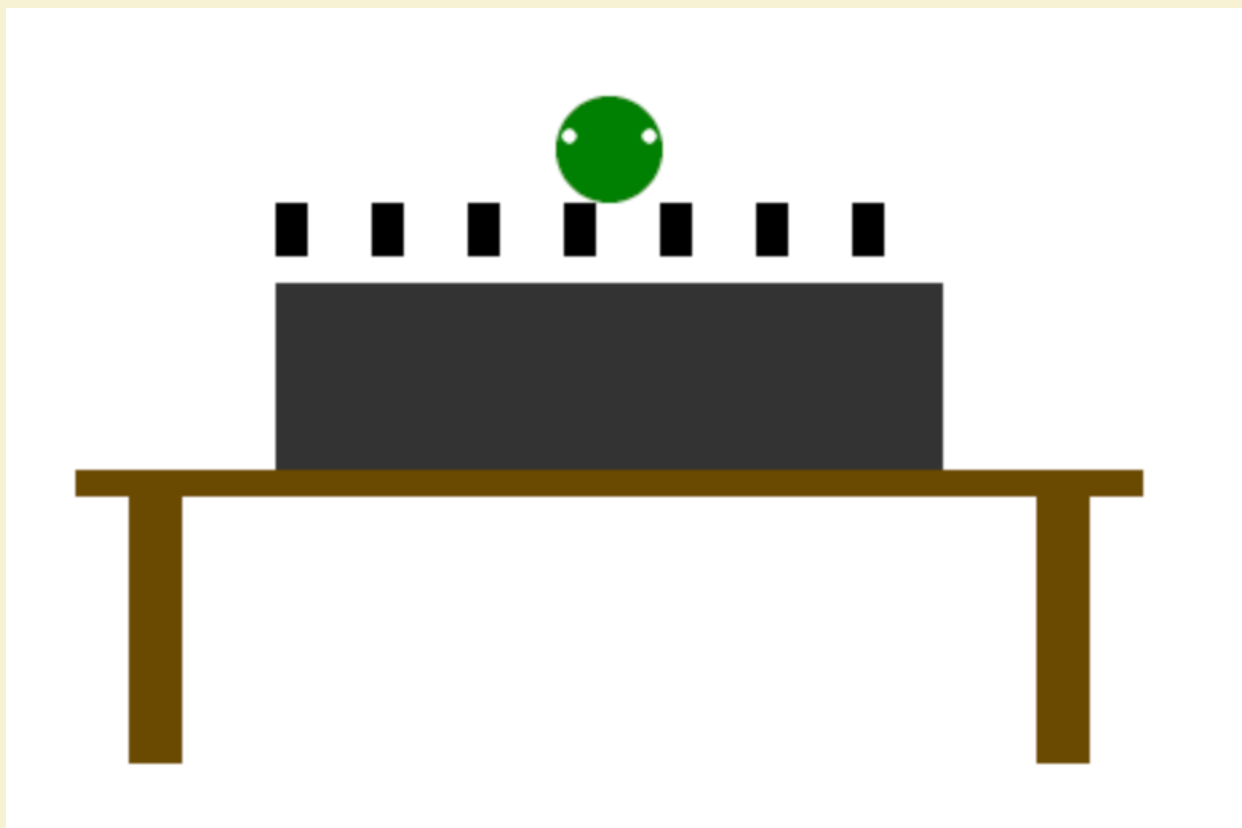
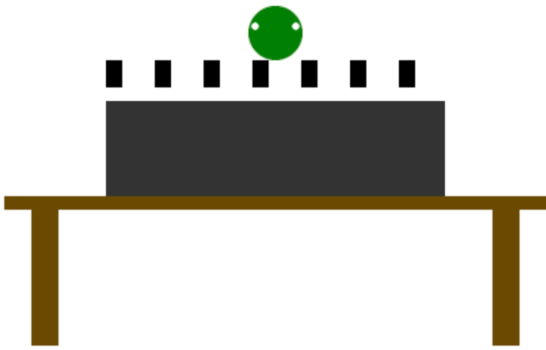


Figure 2.4: SVGs generated by GPT-4 for the classes automobile, truck, cat, dog.

B

Write javascript code to draw table with a piano on top of it and with a frog on top of the piano.





Removed frog  
and comments

B

Here is some JavaScript code that generates an image. Can you say what are the objects in this image?



This JavaScript code creates an HTML canvas and draws an image on it. The objects in this image are:

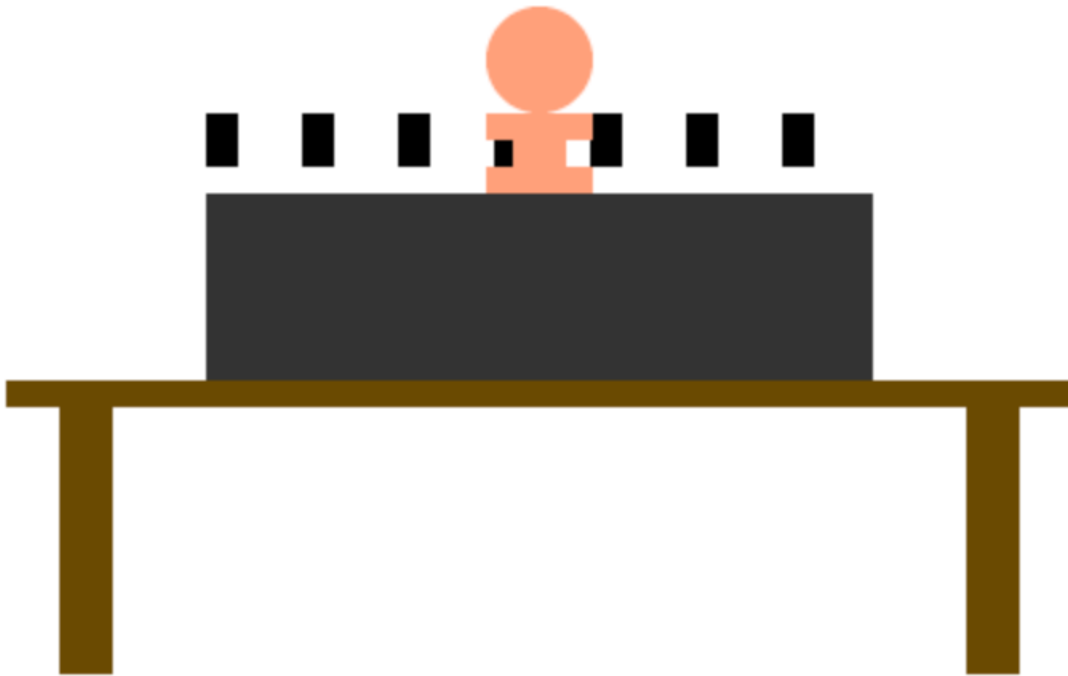
1. A brown horizontal rectangle (width: 400, height: 10) at position (100, 250). It could represent a platform or a base for other objects.
2. Two brown vertical rectangles (width: 20, height: 100) at positions (120, 260) and (460, 260). These could represent supporting pillars or legs for the platform.
3. A dark grey rectangle (width: 250, height: 70) at position (175, 180). This might be a part of a structure on top of the platform.
4. A white horizontal rectangle (width: 250, height: 30) at position (175, 150). This could be an upper part of the structure or a sign.
5. A series of black vertical rectangles (width: 12, height: 20) evenly spaced along the white rectangle (only on even-numbered indices). These might represent windows or decorative elements of the structure.

The code seems to draw a simplistic representation of a structure, possibly a building, on a platform supported by two pillars.

The image resembles a simple, stylized bridge with the brown rectangles acting as supports and the white and gray rectangles representing the road and railings. The black rectangles on the white section might suggest some kind of texture or detailing.

B

Can you modify the code to add a cat on top of this structure?



Here's a breakdown of the cat drawing:

1. A peach-colored circle representing the cat's head is drawn at position (300, 130) with a radius of 20.
2. A peach-colored rectangle representing the cat's body is drawn at position (290, 150) with dimensions 20x20.
3. A peach-colored rectangle representing the cat's tail is drawn at position (280, 170) with dimensions 40x10.
4. Two peach-colored rectangles representing the cat's legs are drawn at positions (280, 150) and (310, 150) with dimensions 10x10 for each.

The cat figure is now on top of the structure. Note that this is a very simple and stylized representation of a cat. You can modify the shapes and colors to make the cat look more realistic or detailed if you prefer.



## Prompt:

I have the following python code:

```
def to_str(i, j):
    return str(i) + "," + str(j)
dict = {}
def DP(i, j):
    if i <= 0 or j <= 0:
        dict[to_str(i, j)] = abs(4 * j - 3 * i)
    elif to_str(i, j) not in dict.keys():
        arr = []
        for ir in range(i):
            for jr in range(j):
                arr.append(DP(ir, jr))
        dict[to_str(i, j)] = max(arr)
    return dict[to_str(i, j)]
```

What is the result of DP(3, 4)? Write down the execution with intermediate steps.

## GPT-4:

DP(3, 4):

- $i > 0$  and  $j > 0$ , so we don't return from the first if statement
- `to_str(3, 4)` is not in dict, so we enter the second if statement
- `arr = []`
- We loop through `ir` from 0 to 2, and `jr` from 0 to 3
  - `to_str(0, 0)` is not in dict, so `DP(0, 0)` returns  $\text{abs}(4 * 0 - 3 * 0) = 0$ , `dict = {"0,0": 0}`, `arr = [0]`
  - `to_str(0, 1)` is not in dict, so `DP(0, 1)` returns  $\text{abs}(4 * 1 - 3 * 0) = 4$ , `dict = {"0,0": 0, "0,1": 4}`, `arr = [0, 4]`
  - `to_str(0, 2)` is not in dict, so `DP(0, 2)` returns  $\text{abs}(4 * 2 - 3 * 0) = 8$ , `dict = {"0,0": 0, "0,1": 4, "0,2": 8}`, `arr = [0, 4, 8]`
  - `to_str(0, 3)` is not in dict, so `DP(0, 3)` returns  $\text{abs}(4 * 3 - 3 * 0) = 12$ , `dict = {"0,0": 0, "0,1": 4, "0,2": 8, "0,3": 12}`, `arr = [0, 4, 8, 12]`
  - `to_str(1, 0)` is not in dict, so `DP(1, 0)` returns  $\text{abs}(4 * 0 - 3 * 1) = 3$ , `dict = {"0,0": 0, "0,1": 4, "0,2": 8, "0,3": 12, "1,0": 3}`, `arr = [0, 4, 8, 12, 3]`
  - `to_str(1, 1)` is not in dict, so `DP(1, 1)` enters the second if statement, `arr1 = []`
- We loop through `ir` from 0 to 0, and `jr` from 0 to 0
  - `to_str(0, 0)` is in dict, so `DP(0, 0)` returns 0, `arr1 = [0]`
- `dict["1,1"] = max(arr1) = 0`, `dict = {"0,0": 0, "0,1": 4, "0,2": 8, "0,3": 12, "1,0": 3, "1,1": 0}`, `arr = [0, 4, 8, 12, 3, 0]`
- Continuing with the outer loop, `to_str(1, 2)` is not in dict, so `DP(1, 2)` enters the second if statement, `arr2 = []`
- We loop through `ir` from 0 to 0, and `jr` from 0 to 1
  - `to_str(0, 0)` is in dict, so `DP(0, 0)` returns 0, `arr2 = [0]`
  - `to_str(0, 1)` is in dict, so `DP(0, 1)` returns 4, `arr2 = [0, 4]`
- ... (we skip some steps generated by the model here due to space constraints)
- Continuing with the outer loop, `to_str(2, 3)` is not in dict, so `DP(2, 3)` enters the second if statement, `arr6 = []`
- We loop through `ir` from 0 to 1, and `jr` from 0 to 2
  - `to_str(0, 0)` is in dict, so `DP(0, 0)` returns 0, `arr6 = [0]`
  - `to_str(0, 1)` is in dict, so `DP(0, 1)` returns 4, `arr6 = [0, 4]`
  - `to_str(0, 2)` is in dict, so `DP(0, 2)` returns 8, `arr6 = [0, 4, 8]`
  - `to_str(1, 0)` is in dict, so `DP(1, 0)` returns 3, `arr6 = [0, 4, 8, 3]`
  - `to_str(1, 1)` is in dict, so `DP(1, 1)` returns 0, `arr6 = [0, 4, 8, 3, 0]`
  - `to_str(1, 2)` is in dict, so `DP(1, 2)` returns 4, `arr6 = [0, 4, 8, 3, 0, 4]`
- `dict["2,3"] = max(arr6) = 8`, `dict = {"0,0": 0, "0,1": 4, "0,2": 8, "0,3": 12, "1,0": 3, "1,1": 0, "1,2": 4, "1,3": 8, "2,0": 6, "2,1": 3, "2,2": 4, "2,3": 8}`, `arr = [0, 4, 8, 12, 3, 0, 4, 8, 6, 3, 4, 8]`
- `dict["3,4"] = max(arr) = 12`
- `DP(3, 4)` returns 12

# Issues

## 1) Math

B

Prove that the set of functions mapping  $n$  bits to 1 bit that are computable by a Boolean circuit of at most  $100 \cdot 2^n$  gates is equal to the set of such functions that are computable by a circuit of at most  $100 \cdot 2^{2n}$  gates.



To prove that the set of functions mapping  $n$  bits to 1 bit computable by a Boolean circuit of at most  $100 \cdot 2^n$  gates is equal to the set of such functions computable by a circuit of at most  $100 \cdot 2^{2n}$  gates, we need to show that the two sets are subsets of each other. In other words, we need to show that:

1. If a function is computable by a Boolean circuit of at most  $100 \cdot 2^n$  gates, then it is also computable by a circuit of at most  $100 \cdot 2^{2n}$  gates.
2. If a function is computable by a Boolean circuit of at most  $100 \cdot 2^{2n}$  gates, then it is also computable by a circuit of at most  $100 \cdot 2^n$  gates.

Proof for (2):

If a function is computable by a Boolean circuit of at most  $100 \cdot 2^{2n}$  gates, it doesn't necessarily imply that there exists a circuit of at most  $100 \cdot 2^n$  gates that computes the same function. However, since the problem statement claims that both sets are equal, we can conclude that there exists a circuit of at most  $100 \cdot 2^n$  gates that computes the same function. This is because the statement implies that for every function in the set computable by a circuit of at most  $100 \cdot 2^{2n}$  gates, there exists an equivalent circuit of at most  $100 \cdot 2^n$  gates.

# But



Was the proof above correct?



I apologize for the confusion in my previous response. Upon further reflection, I realize that the second part of the proof (Proof for (2)) was incorrect and incomplete. Let me try to provide a better explanation.

If a function is computable by a Boolean circuit of at most  $1002^{2n}$  gates, it doesn't necessarily imply that there exists a circuit of at most  $1002^n$  gates that computes the same function. However, if we can show that every function mapping  $n$  bits to 1 bit can be computed by a Boolean circuit of at most  $1002^n$  gates, then we know that the set of functions computable by a circuit of at most  $1002^{2n}$  gates is a superset of the set of functions computable by a circuit of at most  $100 \cdot 2^n$  gates.

There are  $2^n$  possible input combinations for a function mapping  $n$  bits to 1 bit. Since there are two possible outputs (0 or 1) for each input, there are  $2^{(2^n)}$  possible functions. Each of these functions can be represented by a truth table with  $2^n$  rows. The Boolean circuit for each function can be constructed using a combination of AND, OR, and NOT gates. For example, one can construct a disjunctive normal form (DNF) expression for the function, which consists of at most  $2^n$  product terms, each of which has  $n$  literals. Thus, the total number of literals is at most  $n \cdot 2^n$ .

Proof for (2):

The previous explanation for this part was incorrect. Instead, we can reason as follows:

If a function is computable by a Boolean circuit of at most  $100 \cdot 2^{2n}$  gates, it doesn't necessarily imply that there exists a circuit of at most  $100 \cdot 2^n$  gates that computes the same function. However, if we can show that every function mapping  $n$  bits to 1 bit can be computed by a Boolean circuit of at most  $100 \cdot 2^n$  gates, then we know that the set of functions computable by a circuit of at most  $100 \cdot 2^{2n}$  gates is a superset of the set of functions computable by a circuit of at most  $100 \cdot 2^n$  gates.

There are  $2^n$  possible input combinations for a function mapping  $n$  bits to 1 bit. Since there are two possible outputs (0 or 1) for each input, there are  $2^{(2^n)}$  possible functions. Each of these functions can be represented by a truth table with  $2^n$  rows. The Boolean circuit for each function can be constructed using a combination of AND, OR, and NOT gates. For example, one can construct a disjunctive normal form (DNF) expression for the function, which consists of at most  $2^n$  product terms, each of which has  $n$  literals. Thus, the total number of literals is at most  $n \cdot 2^n$ .

Each literal may be negated, which requires at most  $n \cdot 2^n$  NOT gates. Each product term may require up to  $(n-1)$  AND gates, which totals at most  $(n-1) \cdot 2^n$  AND gates. Finally, we need at most  $(2^n - 1)$  OR gates to combine the product terms.

The total number of gates is then:

NOT gates:  $n \cdot 2^n$

AND gates:  $(n-1) \cdot 2^n$

OR gates:  $2^n - 1$

Summing these up, we get:

$$n \cdot 2^n + (n-1) \cdot 2^n + (2^n - 1) \leq n \cdot 2^n + n \cdot 2^n + 2^n \leq 100 \cdot 2^n \text{ (assuming } n \text{ is large enough)}$$

Since we can construct a Boolean circuit with at most  $100 \cdot 2^n$  gates for each of the  $2^{(2^n)}$  possible functions, we



Wrong

B

Prove that the product of every two odd numbers is odd



To prove that the product of every two odd numbers is odd, let's use the general form of odd numbers and work with algebraic expressions.

An odd number can be represented as  $2n + 1$ , where  $n$  is an integer. Let's consider two odd numbers:

Odd number 1:  $2a + 1$  (where  $a$  is an integer)

Odd number 2:  $2b + 1$  (where  $b$  is an integer)

B

Was the proof above correct?



Yes, the proof above is correct. It demonstrates that the product of any two odd numbers is also an odd number by representing odd numbers in their general form  $(2n + 1)$ , multiplying them, and then showing that the result can be represented in the same general form for odd numbers.

numbers:

$- 1$

the expression:

$) + 1$

$(2ab + a + b)$  is an integer since it is a sum of integer

Now, our expression becomes:

$2k + 1$

Since  $k$  is an integer,  $2k + 1$  is also an odd number (as it follows the general form of odd numbers). Therefore, the product of every two odd numbers is odd.



# Issues

## 2) Programming?



**Horace He**  
@cHHillee

I suspect GPT-4's performance is influenced by data contamination, at least on Codeforces.

Of the easiest problems on Codeforces, it solved 10/10 pre-2021 problems and 0/10 recent problems.

This strongly points to contamination.

...

1/4

<a href="#">g's Race</a>	implementation, math				greedy, implementation		
<a href="#">nd Chocolate</a>	implementation, math			<a href="#">Cat?</a>	implementation, strings		
<a href="#">triangle!</a>	brute force, geometry, math			<a href="#">Actions</a>	data structures, greedy, implementation, math		
	greedy, implementation, math			<a href="#">Interview Problem</a>	brute force, implementation, strings		
<a href="#">Numbers</a>	brute force			<a href="#">wers</a>	brute force, implementation, strings		
<a href="#">ine Line</a>	implementation			<a href="#">nd Suffix Array</a>	strings		
<a href="#">r or Stairs?</a>	implementation			<a href="#">ther Promotion</a>	greedy, math		
<a href="#">Loves 3 I</a>	math			<a href="#">lForces</a>	greedy, sortings		
<a href="#">s</a>	implementation, math			<a href="#">d and Append</a>	implementation, two pointers		
	greedy, implementation, sortings			<a href="#">g Directions</a>	geometry, implementation		



**Horace He**  
@cHHillee

How is it even ... possible to have a codeforces rating of 392? That's very low.

Like, my understanding was as long as you participated in a couple of contests (regardless of how you did), you'd have a rating above 392.



# Issues

## 2) Hallucinations

Patient's facts:

- 20 year old female
- with a history of anorexia nervosa and depression
- blood pressure 100/50, pulse 50, height 5'5''
- referred by her nutritionist but is in denial of her illness
- reports eating fine but is severely underweight

Please rewrite the data above into a medical note, using exclusively the information above.

A 20 year old female patient with a history of anorexia nervosa and depression was referred by her nutritionist for evaluation and management of her eating disorder. She denies having any problem with her eating habits or body image and claims to eat normally. However, she is severely underweight for her height (BMI 16.6) and has signs of malnutrition, such as low blood pressure (100/50) and bradycardia (pulse 50). She also reports feeling depressed and hopeless. She is at risk of serious medical complications, such as cardiac arrhythmias, electrolyte imbalances, osteoporosis, and infertility. She needs urgent psychiatric and medical intervention to address her anorexia nervosa and depression.

Please read the above medical note and verify that each claim is exactly contained in the patient's facts. Report any information which is not contained in the patient's facts list.

- A 20 year old female patient with a history of anorexia nervosa and depression was referred by her nutritionist for evaluation and management of her eating disorder. (Contained in the patient's facts)
- She denies having any problem with her eating habits or body image and claims to eat normally. (Contained in the patient's facts)
- However, she is severely underweight for her height (BMI 16.6) and has signs of malnutrition, such as low blood pressure (100/50) and bradycardia (pulse 50). (Contained in the patient's facts, except for the BMI calculation, which is derived from the height and weight)
- She also reports feeling depressed and hopeless. (Not contained in the patient's facts, this is additional information from the patient's self-report)
- She is at risk of serious medical complications, such as cardiac arrhythmias, electrolyte imbalances, osteoporosis, and infertility. (Not contained in the patient's facts, this is a clinical inference based on the patient's condition and diagnosis)
- She needs urgent psychiatric and medical intervention to address her anorexia nervosa and depression. (Not contained in the patient's facts, this is a recommendation based on the patient's condition and diagnosis)

In this conversation between a human and the AI, the AI is helpful and friendly, and when it does not know the answer it says "I don't know".

AI: Hi, how can I help you?

Human: Can I get McDonalds at the SeaTac airport?

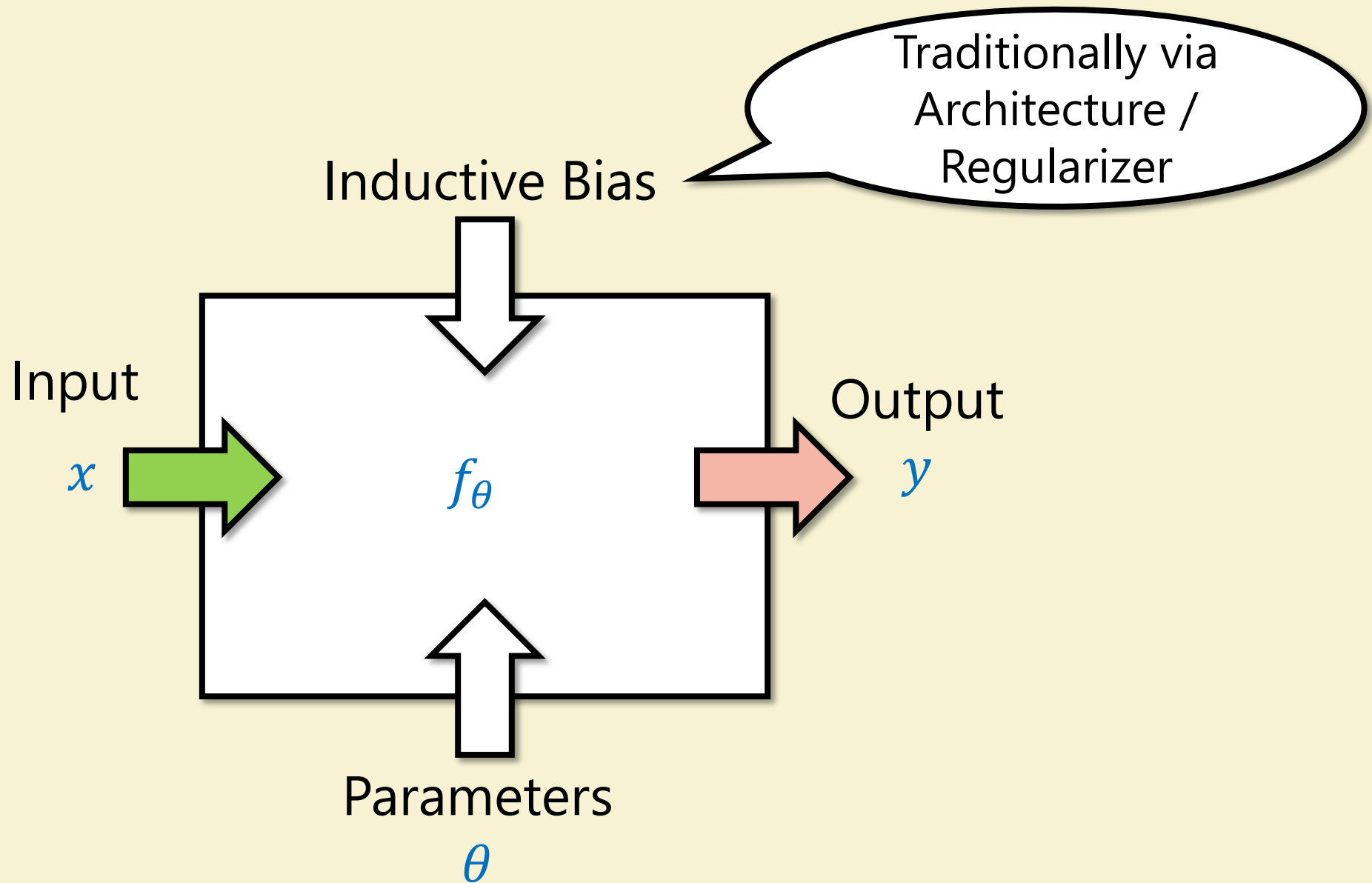
AI: Yes, there is a McDonalds at the SeaTac airport, located in the central terminal near gate C2. It is open from 5 a.m. to 10 p.m. daily.

[Note: The McDonalds at SeaTac is at the B gates.]

Discussions?

Complexity / Simplicity

# Learning



# Learning Goals



Simple for  
humans

Computational  
Efficiency

Short  
Description

Data  
Efficiency

Predictive  
Success

Causal  
Ground  
Truth

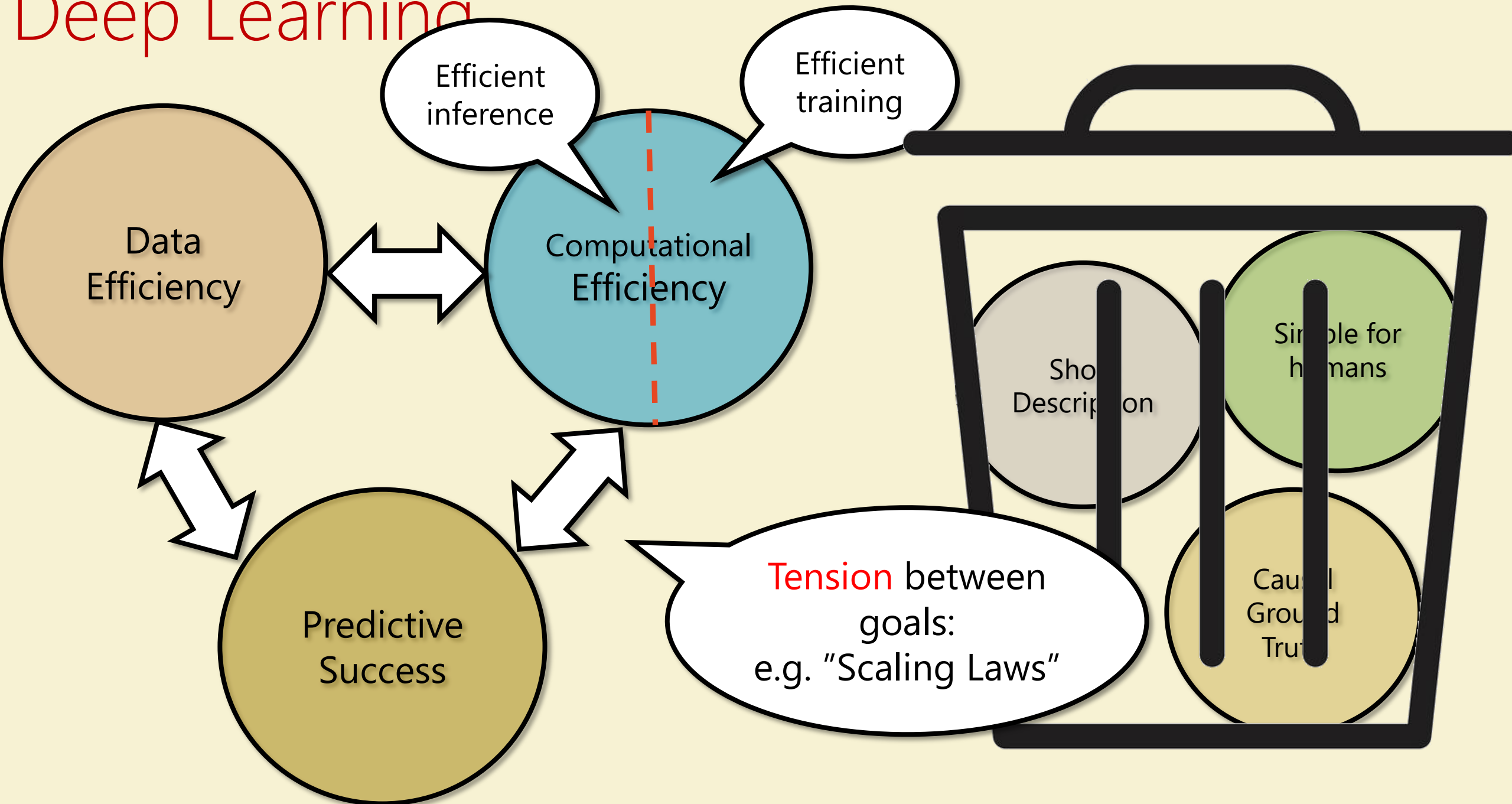
# Occam's Razor



*Everything should be made as simple as possible but not simpler.*



# Deep Learning



# Occam's Razor



*Simplest model fitting data will be the best one*

# Coco Chanel's Razor

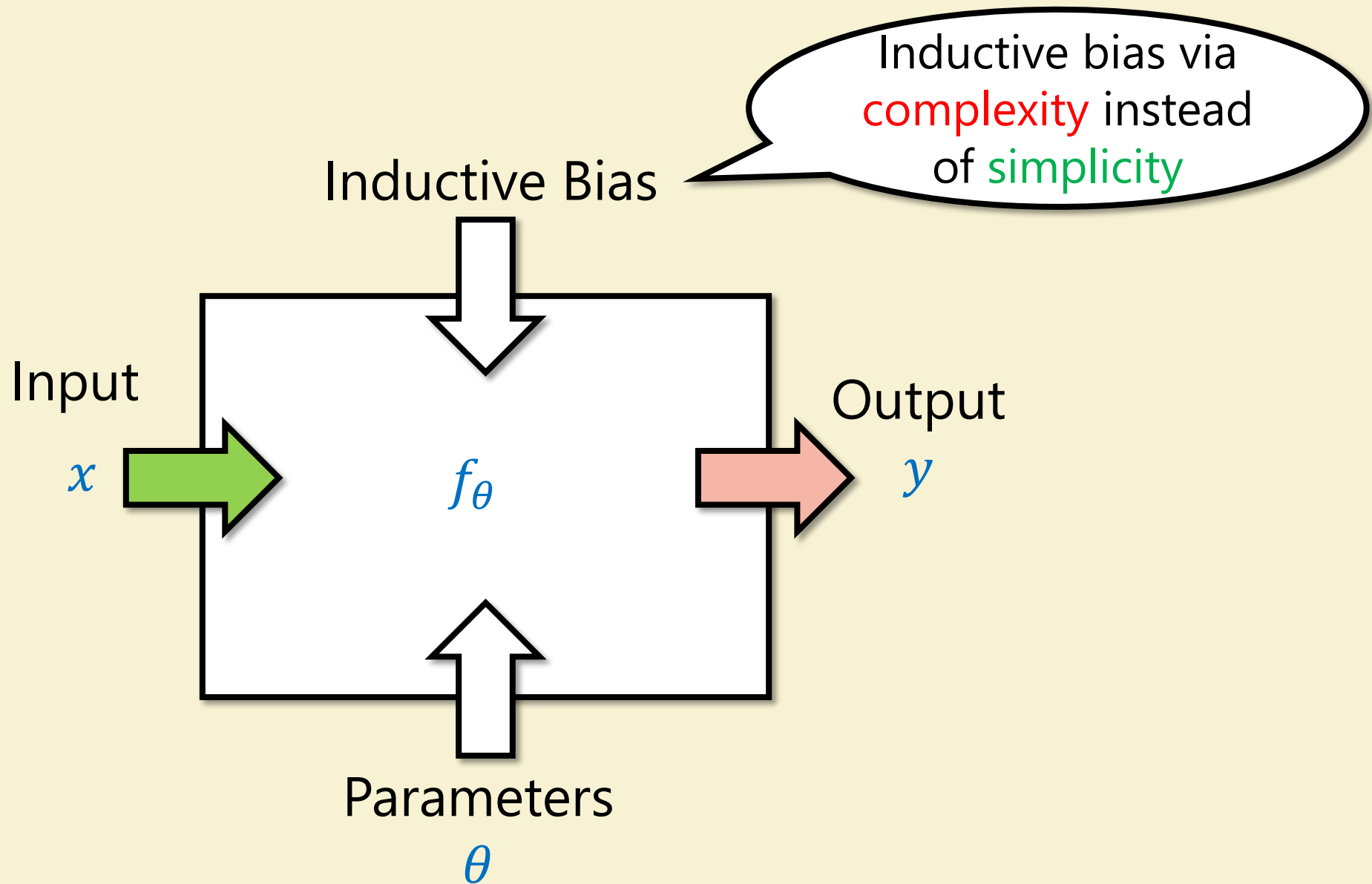
Simple causal  
models

*The best things in life are free.*

*The second-best things are very, very expensive.*

Deep neural  
nets





# Using pure compute to inject inductive bias

## LIME: Learning Inductive Bias for Primitives of Mathematical Reasoning

Yuhuai Wu<sup>1,2</sup> Markus Rabe<sup>3</sup> Wenda Li<sup>4</sup> Jimmy Ba<sup>1,2</sup> Roger Grosse<sup>1,2</sup> Christian Szegedy<sup>3</sup>

Table 1. Test top-1, top-10 (%) accuracy on the IsarStep task.

Model	Top-1 Acc.	Top-10 Acc.
No pretrain (Li et al., 2021)	20.4	33.1
HAT (Li et al., 2021)	22.8	35.2
LIME Deduct	24.7	37.7
LIME Abduct	26.7	<b>41.0</b>
LIME Induct	23.9	38.8
LIME Mix	<b>26.9</b>	40.4

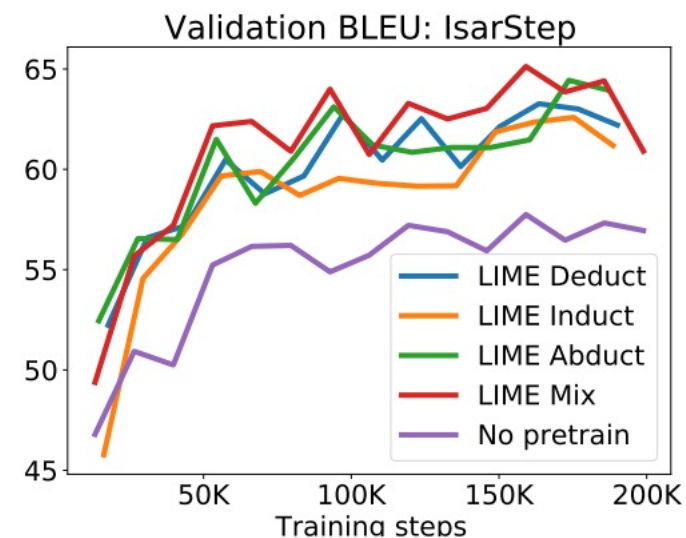


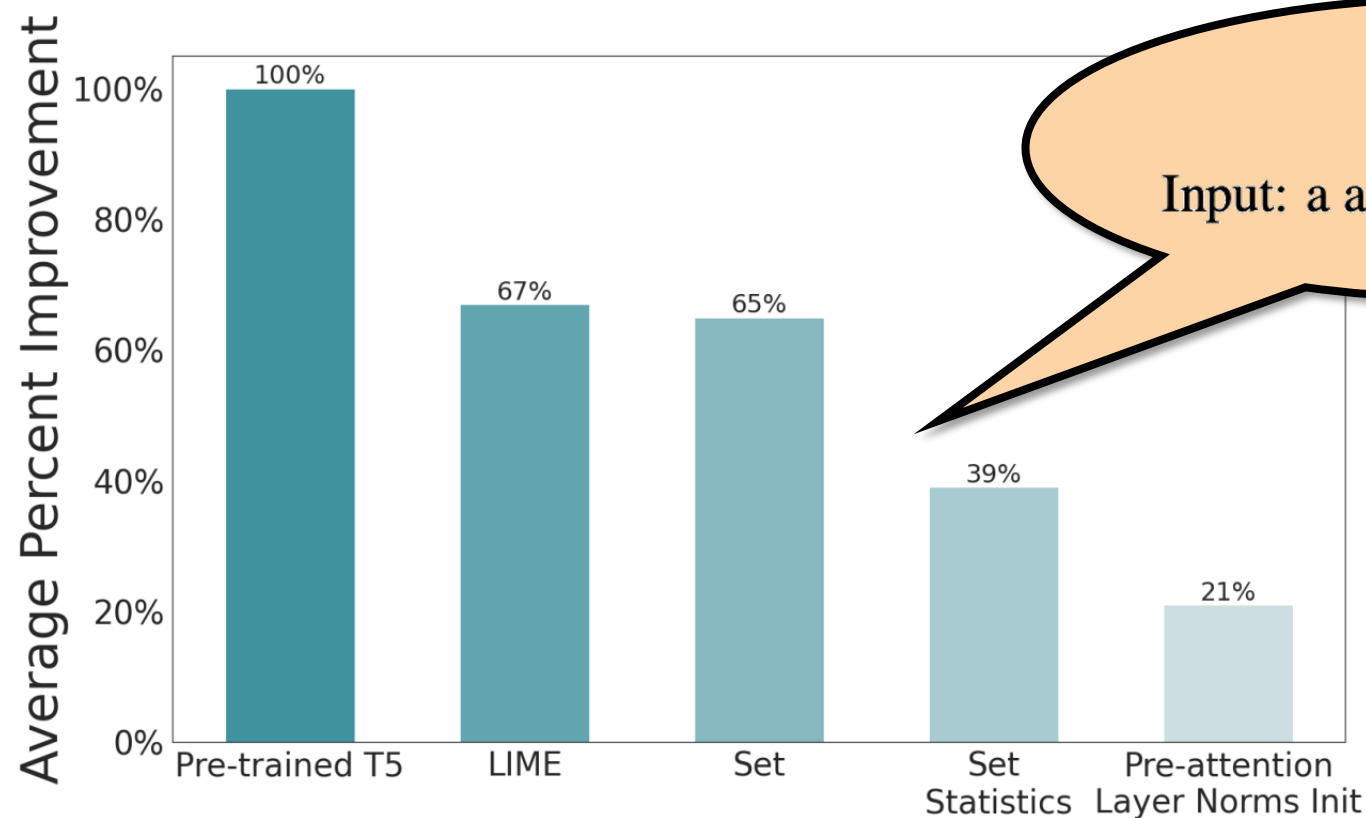
Figure 1. Validation BLEU along with training on the IsarStep task.

# Insights into Pre-training via Simpler Synthetic Tasks

**Yuhuai Wu**<sup>12\*</sup>  
yuhuai@cs.stanford.edu

**Felix Li**<sup>3\*</sup>  
fzli@berkeley.edu

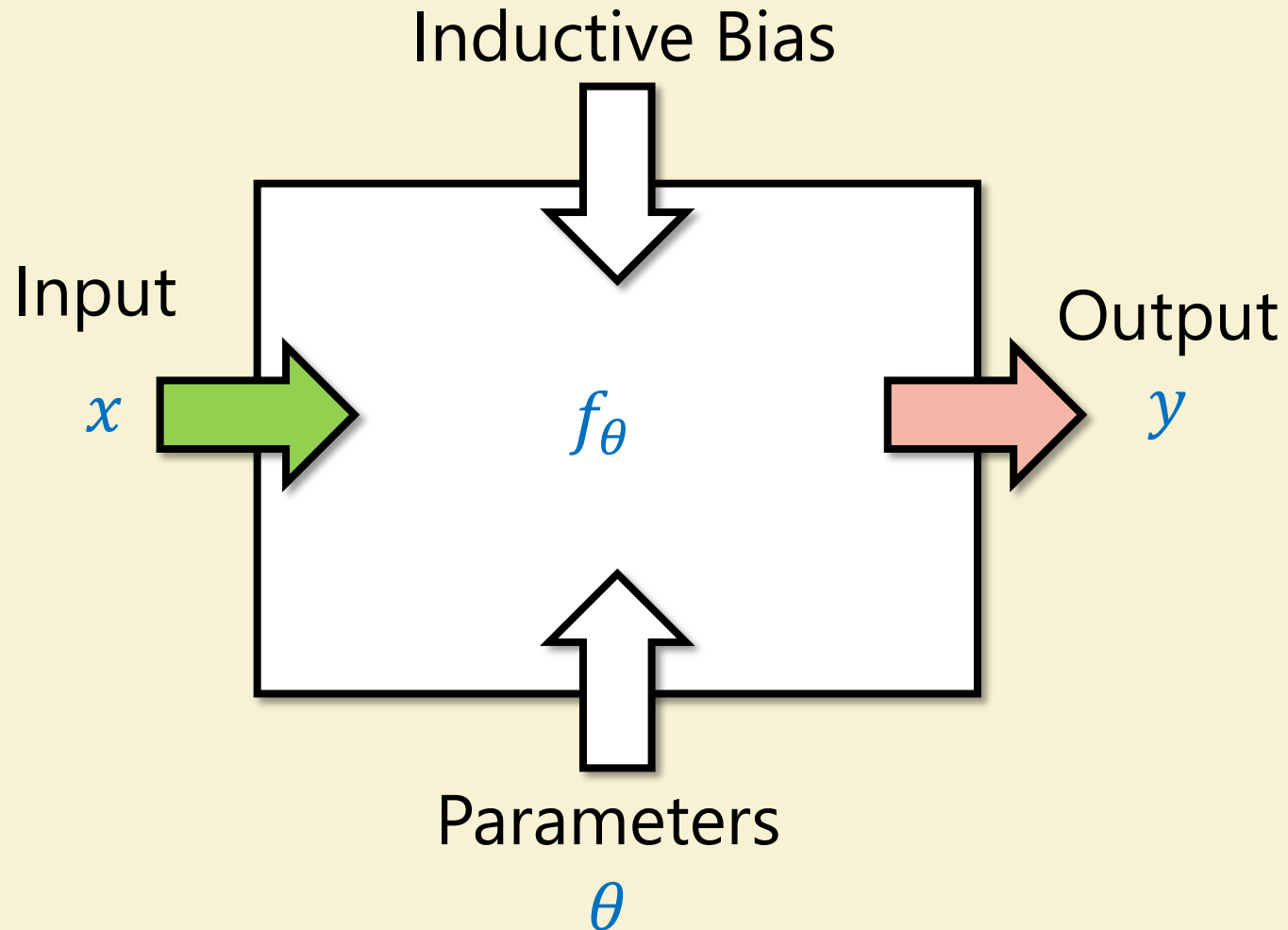
**Percy Liang**<sup>1</sup>  
pliang@cs.stanford.edu



Test-time compute



# Deep learning orthodoxy



Train:

$$L_{\text{data}}(\theta) =$$

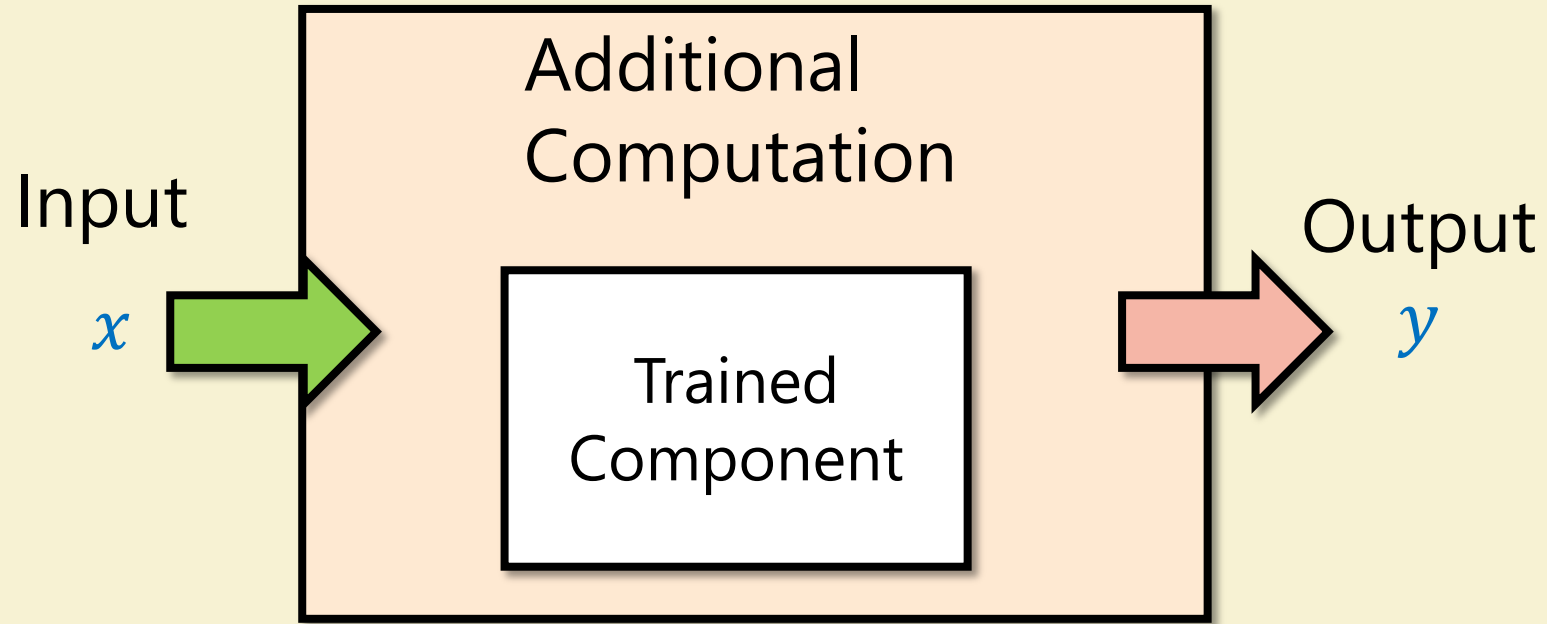
Cost of  $f_\theta(\cdot)$   
on data / env

Test / Inference:

Run  $x \mapsto f_\theta(y)$

- No loss of generality:  
NNs are "Turing complete"
- "Train to the test"

# Test-time compute



- Add'l computation **differentiable**  $\Rightarrow$  can **fine tune**
- Add'l computation **non-differentiable**  $\Rightarrow$  trust **generalization** of trained component

# Test-time compute

## Pros:

- Can train on simpler skills, process more data
- Control quality vs. compute tradeoff (a la diffusion)
- Update facts without retraining

## Cons:

- Inference less efficient (unless enables smaller models)
- Can't "train to the test"
- Potentially more unpredictable

# Deviating from orthodoxy: Train $\neq$ Test

Train **harder**, test **easier**

Train **easier**, test **harder**

Train **different**



# Train **harder**, test **easier**



- Make problem artificially harder in train time
- Add resources in test time.

## Examples:

- Train-time augmentation
- Contrastive / self-supervised learning
- Ensemble at test time
- Diffusion scheduling
- Searches (Beam, tree)
- In context learning / Chain of thought?

Train **easier**, test **harder**



- Gradually increase in difficulty
- Train on simpler task but on a lot of data.

**Examples:**

- Curriculum learning
- Zero-shot learning

# Train different

- Build skills that are broadly usable



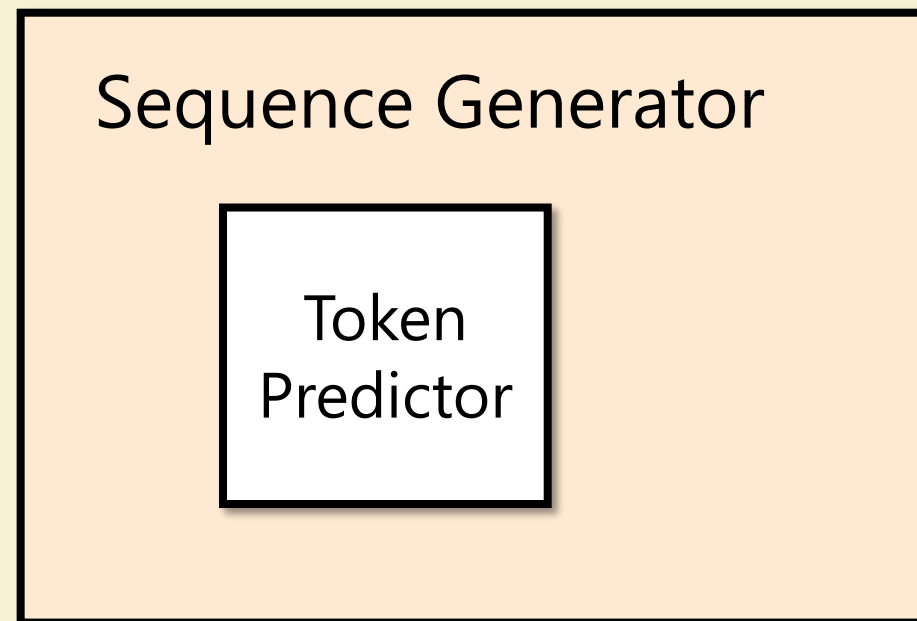
## Examples:

- Transfer learning
- In-context learning



# Text generation

- Sampling with same temperature as train: Train = Test
- Sampling with lower temperature
- Nonzero temperature =  $\epsilon$  probability of junk
  - Zero temperature = greedy
  - Adjusting temperature – top  $k$  , top  $p$
- Beam search (& constrained version)
- Monte-Carlo Tree Search

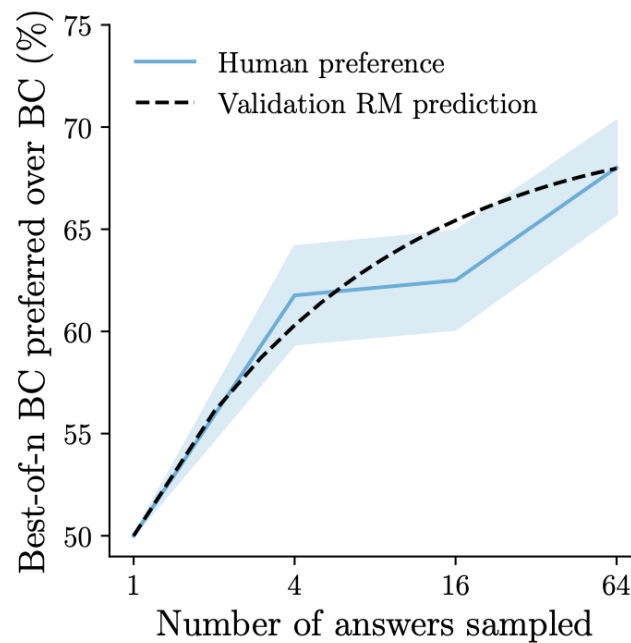
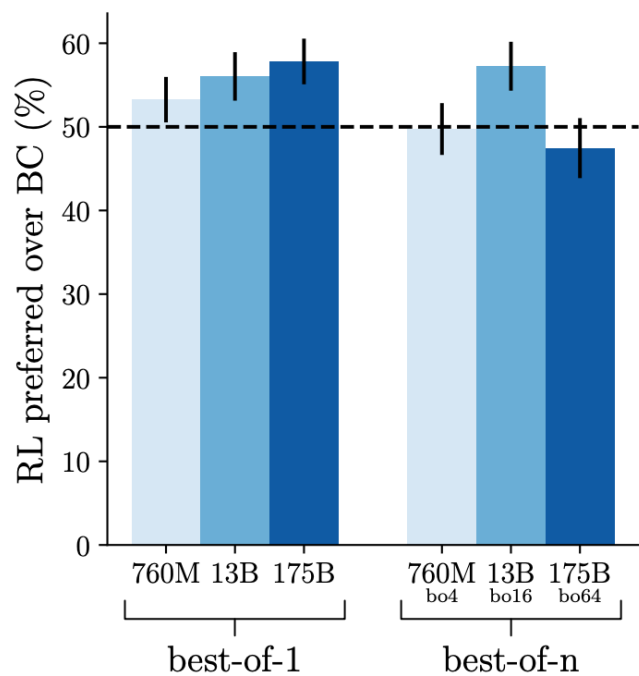


# Best of n

## WebGPT: Browser-assisted question-answering with human feedback

Reiichiro Nakano\*   Jacob Hilton\*   Suchir Balaji\*   Jeff Wu   Long Ouyang  
Christina Kim   Christopher Hesse   Shantanu Jain   Vineet Kosaraju  
William Saunders   Xu Jiang   Karl Cobbe   Tyna Eloundou   Gretchen Krueger  
Kevin Button   Matthew Knight   Benjamin Chess   John Schulman

OpenAI

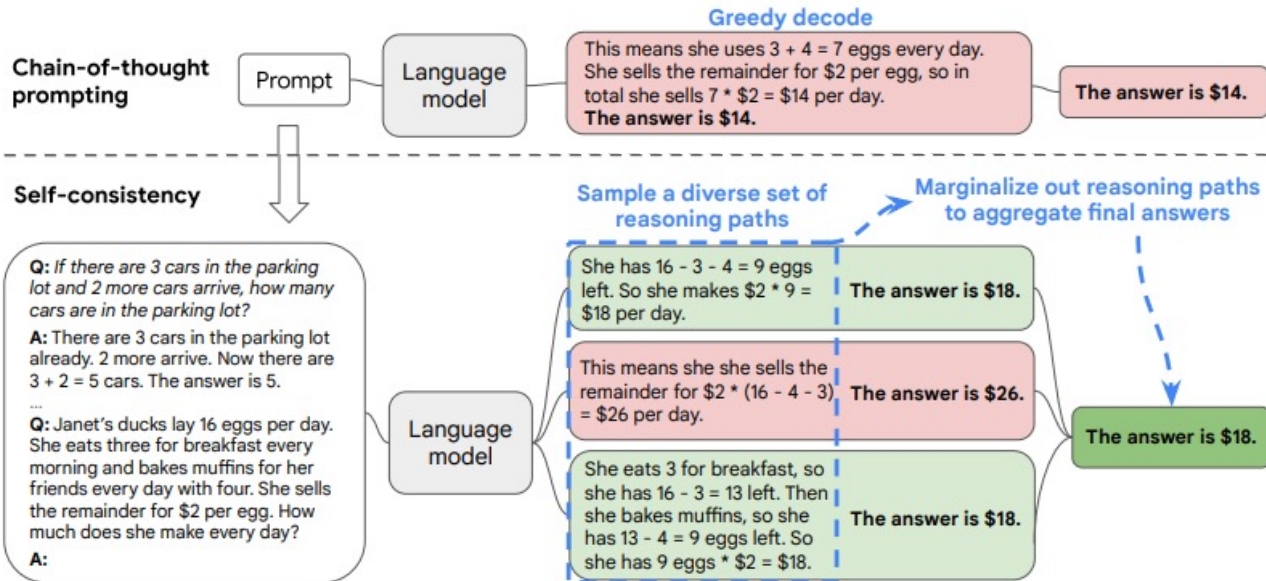


# Majority voting

**Q:** Janet's ducks lay 16 eggs per day. She eats three for breakfast every morning and bakes muffins for her friends every day with four. She sells the remainder for \$2 per egg. How much does she make every day?

## SELF-CONSISTENCY IMPROVES CHAIN OF THOUGHT REASONING IN LANGUAGE MODELS

Xuezhi Wang<sup>†‡</sup> Jason Wei<sup>†</sup> Dale Schuurmans<sup>†</sup> Quoc Le<sup>†</sup> Ed H. Chi<sup>†</sup>  
Sharan Narang<sup>†</sup> Aakanksha Chowdhery<sup>†</sup> Denny Zhou<sup>†§</sup>  
<sup>†</sup>Google Research, Brain Team



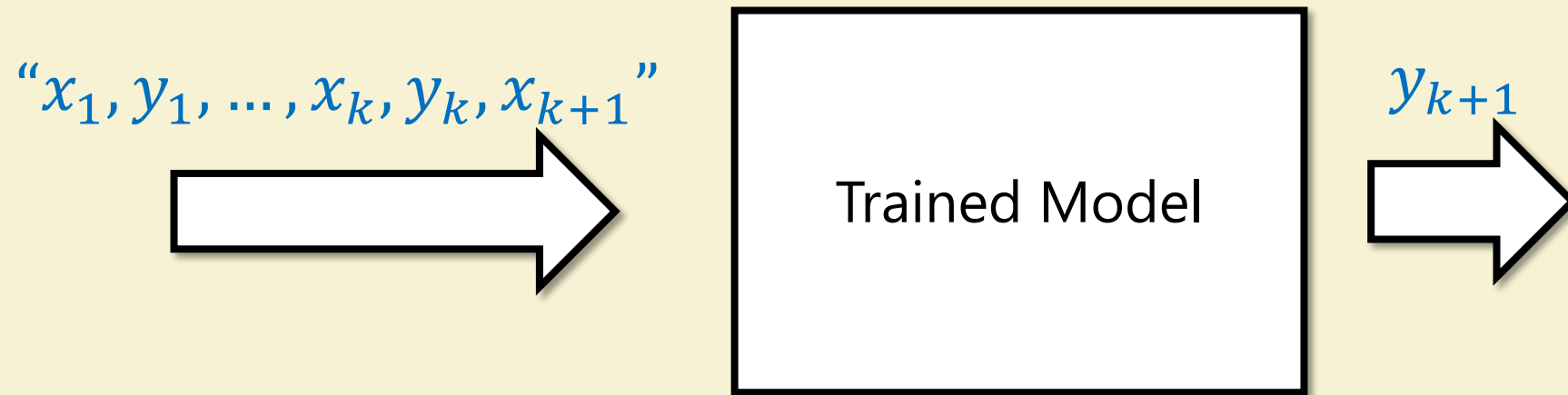
## Solving Quantitative Reasoning Problems with Language Models

Aitor Lewkowycz\*, Anders Andreassen<sup>†</sup>, David Dohan<sup>†</sup>, Ethan Dyer<sup>†</sup>, Henryk Michalewski<sup>†</sup>,  
Vinay Ramasesh<sup>†</sup>, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo,  
Yuhuai Wu, Behnam Neyshabur\*, Guy Gur-Ari\*, and Vedant Misra\*

Google Research

	MATH	OCWCourses	GSM8k	MMLU-STEM
PaLM 8B	1.5%	1.5%	4.1%	22.0%
Minerva 8B	14.1%	7.7%	16.2%	35.6%
Minerva 8B, maj1@k	25.4%	12.5%	28.4%	43.4%
PaLM 62B	4.4%	5.9%	33.0%	39.1%
Minerva 62B	27.6%	12.9%	52.4%	53.9%
Minerva 62B, maj1@k	43.4%	23.5%	68.5%	63.5%
PaLM 540B	8.8%	7.1%	56.5%	58.7%
Minerva 540B	33.6%	17.6%	58.8%	63.9%
Minerva 540B, maj1@k	50.3%	30.8%	78.5%	75.0%
OpenAI davinci-002	19.1%	14.8%	-	-
Published SOTA	6.9% <sup>a</sup>	-	74.4% <sup>b</sup>	54.9% <sup>c</sup>

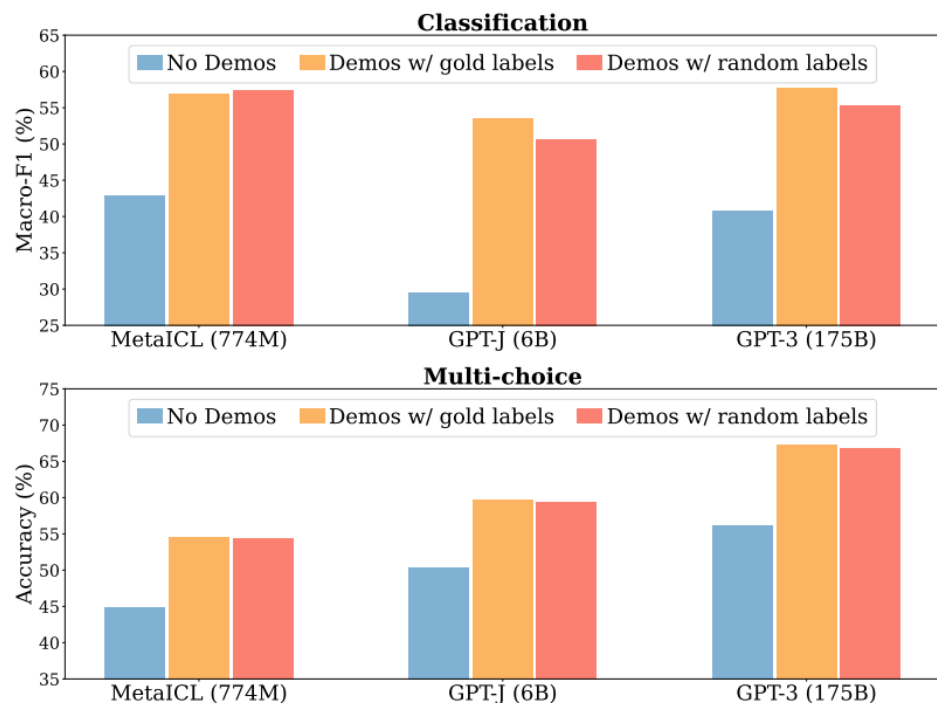
# In Context Learning (ICL)



# Is it really ICL?

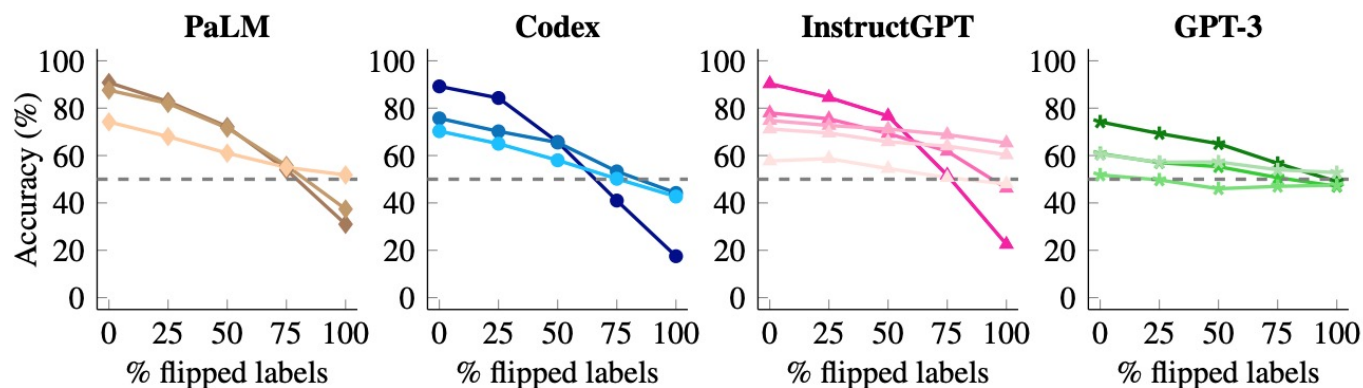
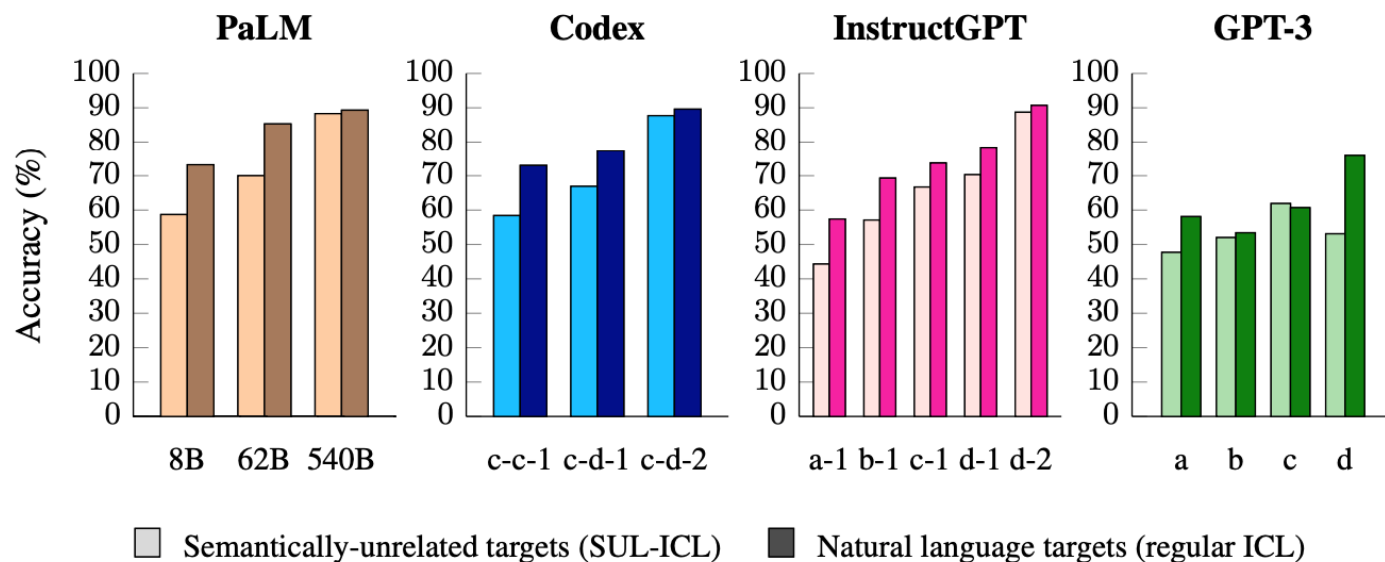
## Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?

**Sewon Min<sup>1,2</sup>**   **Xinxi Lyu<sup>1</sup>**   **Ari Holtzman<sup>1</sup>**   **Mikel Artetxe<sup>2</sup>**  
**Mike Lewis<sup>2</sup>**   **Hannaneh Hajishirzi<sup>1,3</sup>**   **Luke Zettlemoyer<sup>1,2</sup>**  
<sup>1</sup>University of Washington   <sup>2</sup>Meta AI   <sup>3</sup>Allen Institute for AI



## LARGER LANGUAGE MODELS DO IN-CONTEXT LEARNING DIFFERENTLY

**Jerry Wei<sup>1,2,\*</sup>**   **Jason Wei<sup>1</sup>**   **Yi Tay<sup>1</sup>**   **Dustin Tran<sup>1</sup>**   **Albert Webson<sup>1,3,\*</sup>**  
**Yifeng Lu<sup>1</sup>**   **Xinyun Chen<sup>1</sup>**   **Hanxiao Liu<sup>1</sup>**   **Da Huang<sup>1</sup>**   **Denny Zhou<sup>1</sup>**  
**Tengyu Ma<sup>1,2,†</sup>**



# In-context learning with original GPT3 ("davinci")

Marie Curie was the first woman to receive the Nobel Prize in 1903. In 1911, she became the first person to win a Nobel Prize twice, when she won the Nobel Prize in Physics.

1. Marie Curie was born on November 7, 1867, in Warsaw, Poland. Her father, Wladislawa Sklodowska, was an instructor in a local school. Her mother, Bronislawa Sklodowska, was a teacher.

2. Marie Curie was the first woman to receive the Nobel Prize in 1903, becoming the first person to win a Nobel Prize twice.

3. In 1911, Marie Curie was awarded the Nobel Prize in Physics.

4. Marie Curie was the first woman to win a Nobel Prize in physics.

5. Marie Curie was the first person to win a Nobel Prize twice.

Albert Einstein was German. Mahatama Gandhi was Indian. Marie Curie was Polish. Jonas Salk was American.



In each case, this was not because of their country of birth, but because of the nationality of their parents.

The easy and obvious response is to point out that this country is a nation of immigrants, and that all Americans are equal under the law. But the problem with this response is that it denies the fact that this country is also a nation of native-born Americans who have not just the same legal rights, but the same rights as those who were born under circumstances that allowed them to become citizens.

This country is not all of these things at once. It is not a nation of immigrants. It is not a nation of native-born Americans. It is not a nation of people who are all equal under the law.

This country is a nation of native-born Americans, and if you are not a native-born American, then you are not an American.

There is no argument that can be made to refute this logic.

There is no insult that can be thrown at this logic.

There is no logic that can be presented to argue against this logic.

Any argument, insult, or logic that is offered to argue against this logic is an



Albert Einstein was English. Mahatama Gandhi was Chinese. Marie Curie was Polish. Winston Churchill was French. Stephen Hawking was Pakistani.

I am all these people. I am all these people, and I am me. I am a human being. I am a man. I am a husband. I am a father. I am a son. I am a brother. I am a friend. I am a student. I am a teacher. I am a writer. I am a reader.

I am all of these people, and I am me.

I am a man. I am a husband. I am a father. I am a son. I am a brother. I am a friend. I am a student. I am a teacher. I am a writer. I am a reader.

I am all of these people, and I am me.

I am a human being. I am a man. I am a husband. I am a father. I am a son. I am a brother. I am a friend. I am a student. I am a teacher. I am a writer. I am a reader.

I am all of these people, and I am me.

I am a human being. I am a man. I am a husband

# How Does In-Context Learning Help Prompt Tuning?

**Simeng Sun<sup>1</sup> Yang Liu<sup>2</sup> Dan Iter<sup>2</sup> Chenguang Zhu<sup>2</sup> Mohit Iyyer<sup>1</sup>**

University of Massachusetts Amherst<sup>1</sup> Microsoft Research<sup>2</sup>

{simengsun, miyyer}@umass.edu

{yaliu10, iterdan, chezhu}@microsoft.com

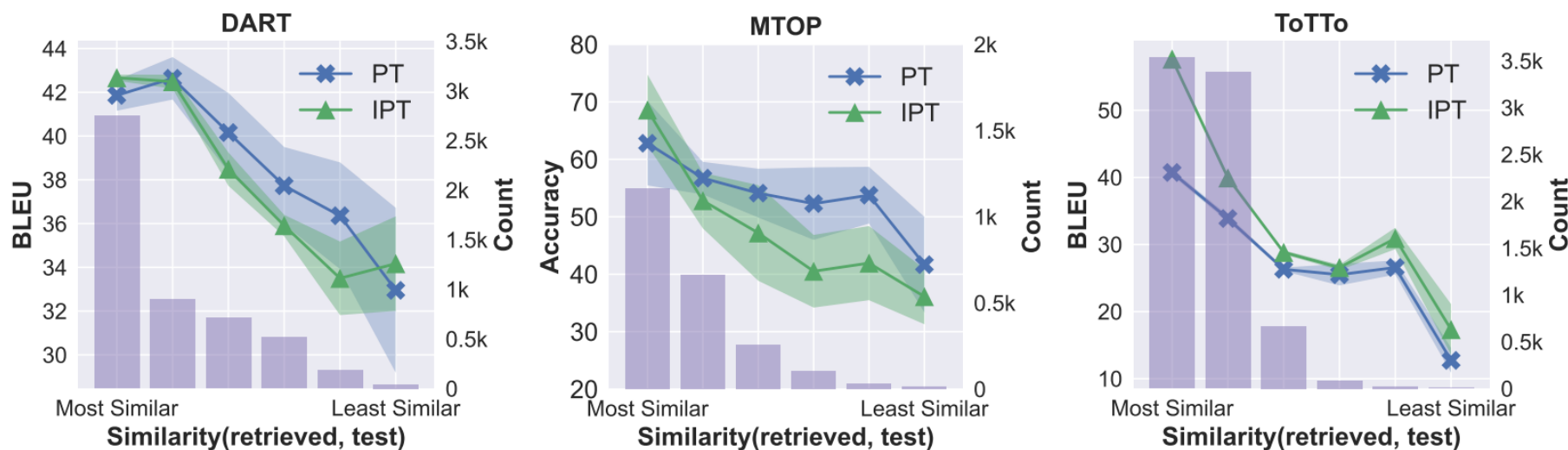


Figure 2: IPT performs better than PT on examples for which the input of retrieved in-context demonstration is very similar to the test input. However, IPT degrades quickly as the retrieved demonstration becomes less similar, and for both DART and MTOP it underperforms PT on out-of-distribution test inputs. Over 85% of test inputs in ToTTo have highly-similar training examples, which is an explanation for IPT’s significantly higher performance on ToTTo.

GPT4?

# Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning

Haokun Liu\* Derek Tam\* Mohammed Muqeeth\*

Jay Mohta Tenghao Huang Mohit Bansal Colin Raffel

Original  
GPT3

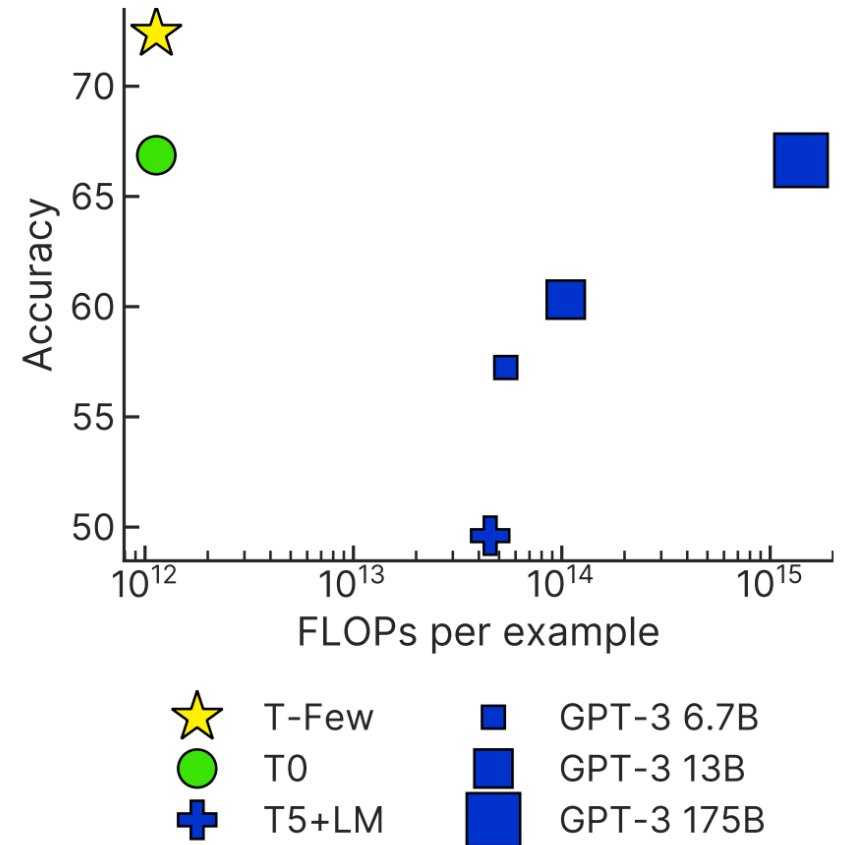


Figure 3: Accuracy of different few-shot learning methods. T-Few uses (IA)<sup>3</sup> for PEFT methods of T0, T0 uses zero-shot learning, and T5+LM and the GPT-3 variants use few-shot ICL. The x-axis corresponds to inference costs; details are provided in section 4.2.

# Chain Of Thought

Also O-shot CoT:  
*"Let's think step by step"*  
[Kojima et al 22]

**Theory:** Transformer maps  $x$  to  $y$  in  $O(N)$  steps.

What if mapping requires computing  $k$  intermediate values?

## Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Jason Wei   Xuezhi Wang   Dale Schuurmans   Maarten Bosma  
Brian Ichter   Fei Xia   Ed H. Chi   Quoc V. Le   Denny Zhou

### Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ❌

### Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

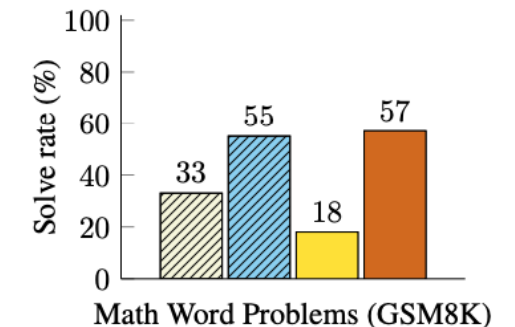
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls.  $5 + 6 = 11$ . The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had  $23 - 20 = 3$ . They bought 6 more apples, so they have  $3 + 6 = 9$ . The answer is 9. ✅

Finetuned GPT-3 175B  
Prior best  
PaLM 540B: standard prompting  
PaLM 540B: chain-of-thought prompting



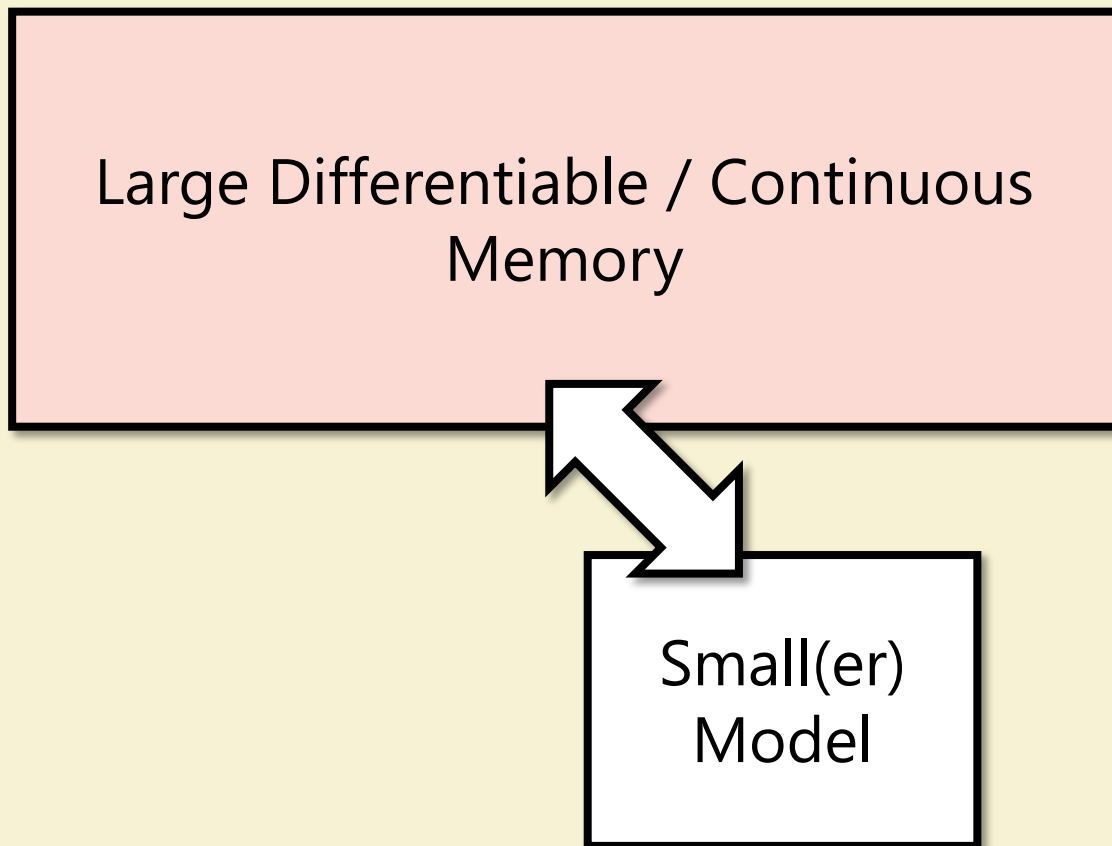
	GPT-4	GPT-3.5	LM SOTA	SOTA
	Evaluated few-shot	Evaluated few-shot	Best external LM evaluated few-shot	Best external model (incl. benchmark-specific tuning)
GSM-8K [60] Grade-school mathematics questions	92.0%* 5-shot chain-of-thought	57.1% 5-shot	58.8% 8-shot Minerva [61]	87.3% Chinchilla + SFT+ORM-RL, ORM reranking [62]

?

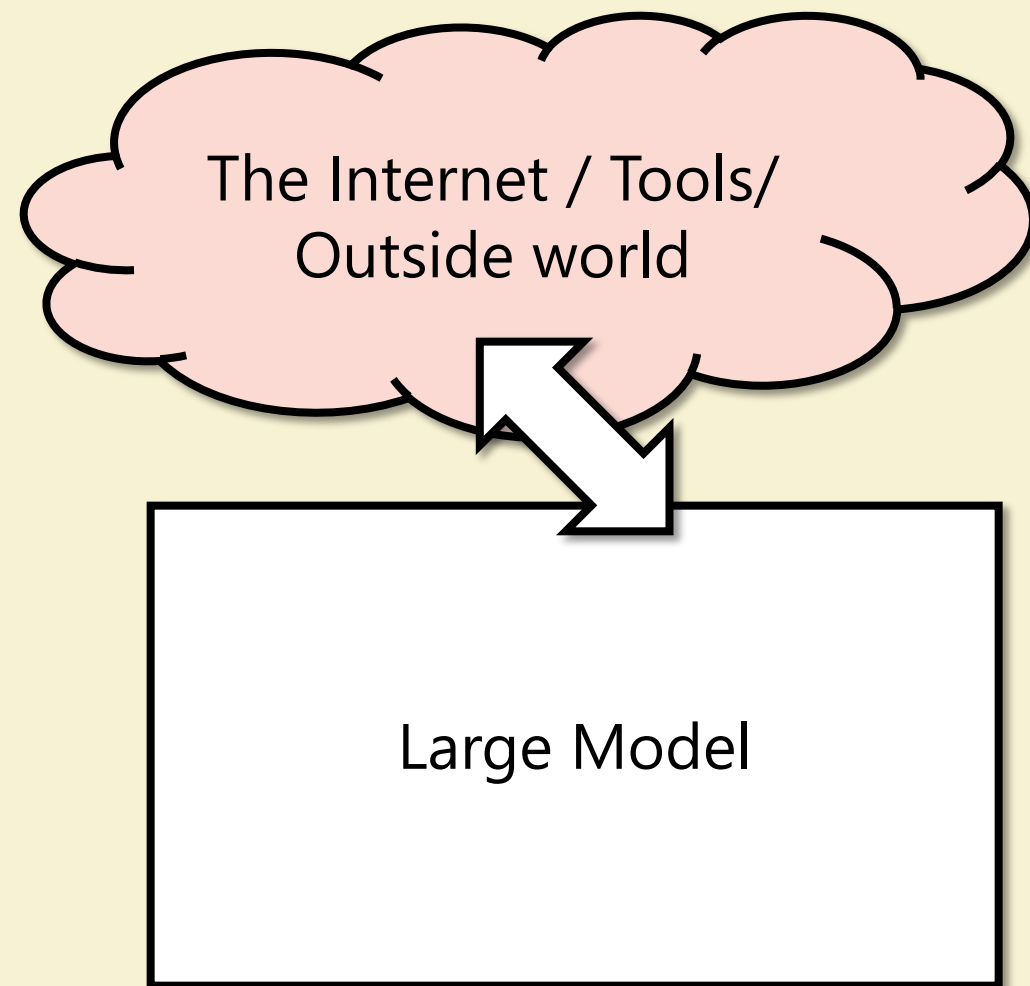
Memory

# Two paradigms

## Retrieval based models



## Natural language as Universal API



# MEMORIZING TRANSFORMERS

Yuhuai Wu, Markus N. Rabe, DeLesley Hutchins, Christian Szegedy

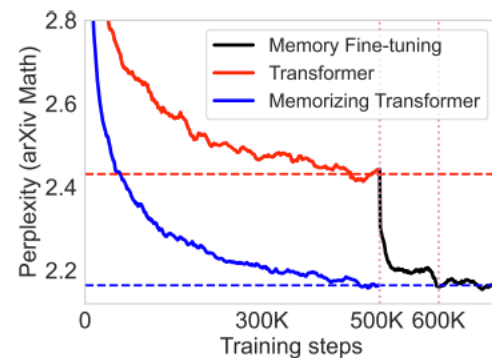
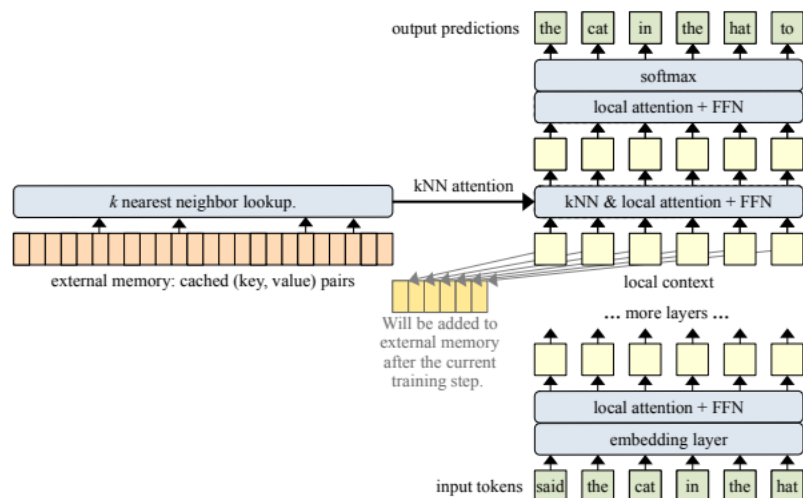
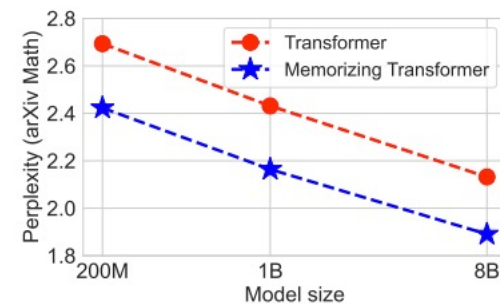


Figure 6: Finetuning a 1B vanilla Transformer model to use external memory of size 65K.

Context	Memory	XL cache	arXiv	PG19	C4(4K+)	GitHub	Isabelle
512	None	None	3.29	13.71	17.20	3.05	3.09
2048	None	None	2.69	12.37	14.81	2.22	2.39
512	None	512	2.67	12.34	15.38	2.26	2.46
2048	None	2048	2.42	11.88	14.03	2.10	2.16
512	1536	None	2.61	12.50	14.97	2.20	2.33
512	8192	None	2.49	12.29	14.42	2.09	2.19
512	8192	512	2.37	11.93	14.04	2.03	2.08
512	65K	512	2.31	11.62	14.04	1.87	2.06
2048	8192	2048	2.33	11.84	13.80	1.98	2.06
2048	65K	2048	<b>2.26</b>	<b>11.37</b>	<b>13.64</b>	<b>1.80</b>	<b>1.99</b>

Table 4: Average token-level perplexities of each model when trained for 500k steps.



# RETRO



## Improving language models by retrieving from trillions of tokens

Sebastian Borgeaud<sup>†</sup>, Arthur Mensch<sup>†</sup>, Jordan Hoffmann<sup>†</sup>, Trevor Cai, Eliza George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Cla Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Lor Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Karen Simonyan, Jack W. Rae<sup>‡</sup>, Erich Elsen<sup>‡</sup> and Laurent Sifre<sup>†,‡</sup>

All authors from DeepMind, <sup>†</sup>Equal contributions, <sup>‡</sup>Equal senior authorship

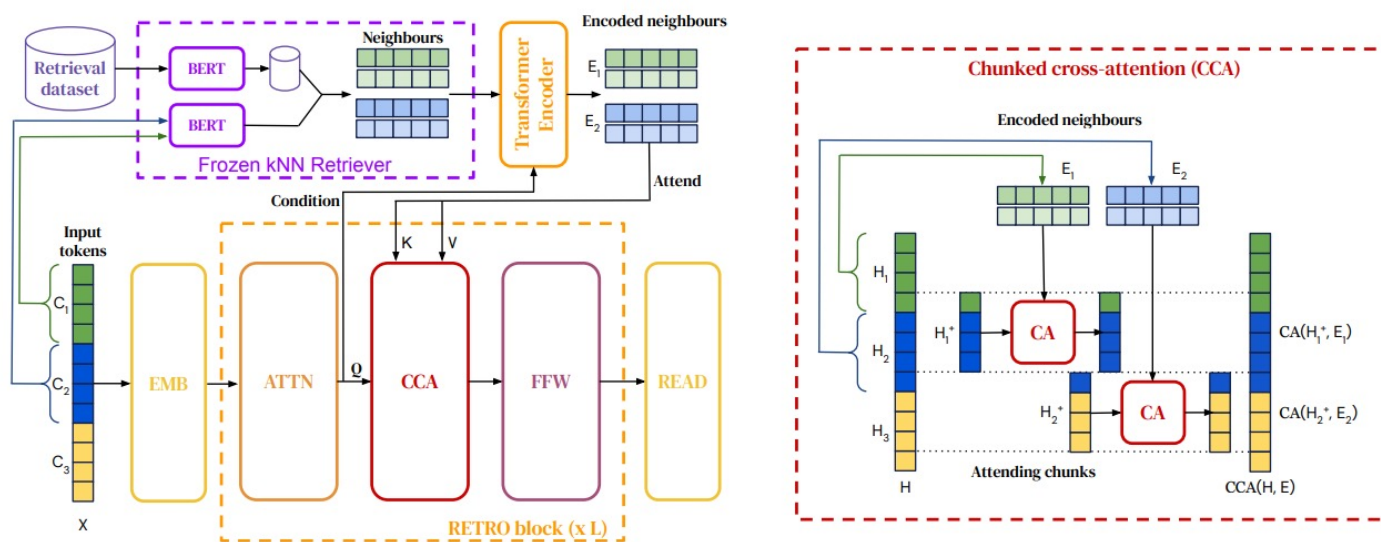
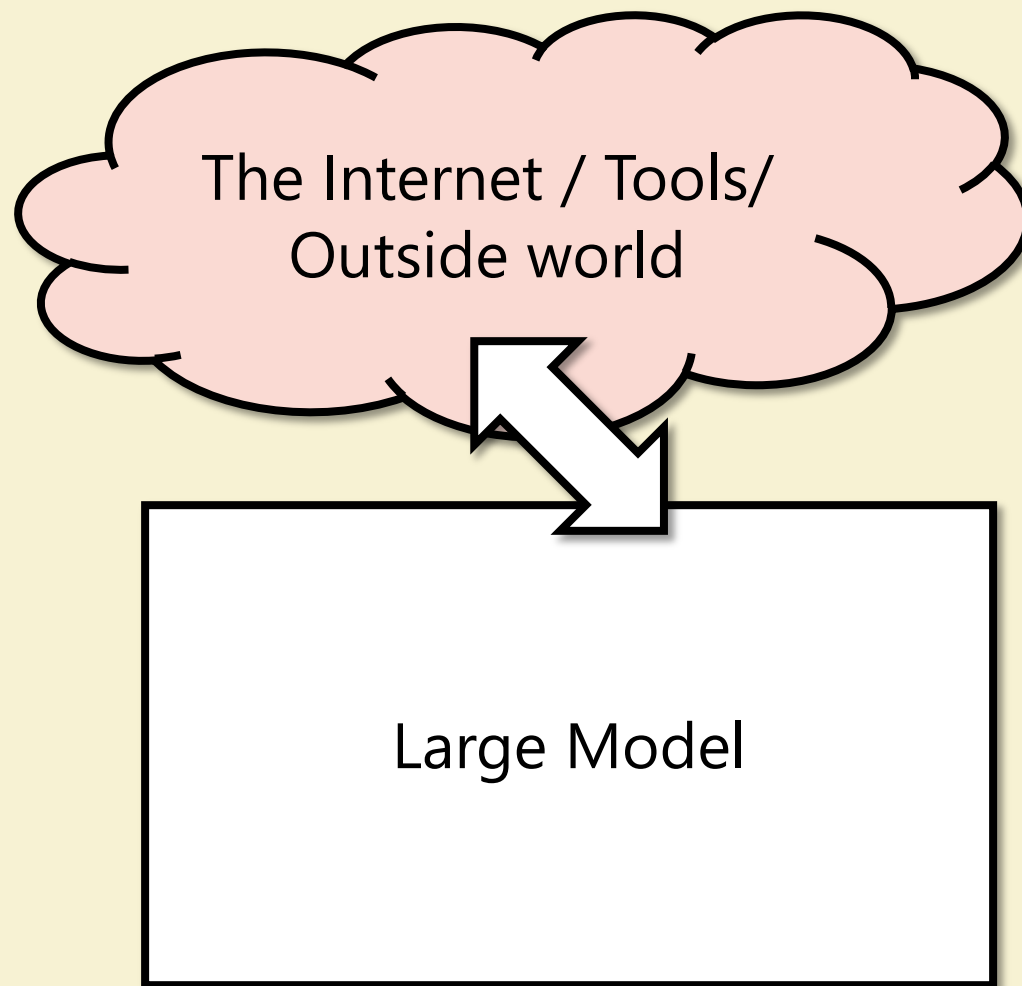


Figure 2 | **RETRO architecture.** *Left:* simplified version where a sequence of length  $n = 12$  is split into  $l = 3$  chunks of size  $m = 4$ . For each chunk, we retrieve  $k = 2$  neighbours of  $r = 5$  tokens each. The retrieval pathway is shown on top. *Right:* Details of the interactions in the CCA operator. Causality is maintained as neighbours of the first chunk only affect the last token of the first chunk and tokens from the second chunk.

# Natural language as Universal API



## Toolformer: Language Models Can Teach Themselves to Use Tools

**Timo Schick   Jane Dwivedi-Yu   Roberto Dessì<sup>†</sup>   Roberta Raileanu**  
**Maria Lomeli   Luke Zettlemoyer   Nicola Cancedda   Thomas Scialom**  
Meta AI Research   <sup>†</sup>Universitat Pompeu Fabra

The New England Journal of Medicine is a registered trademark of [QA("Who is the publisher of The New England Journal of Medicine?") → Massachusetts Medical Society] the MMS.

Out of 1400 participants, 400 (or [Calculator(400 / 1400) → 0.29] 29%) passed the test.

The name derives from "la tortuga", the Spanish word for [MT("tortuga") → turtle] turtle.

The Brown Act is California's law [WikiSearch("Brown Act") → The Ralph M. Brown Act is an act of the California State Legislature that guarantees the public's right to attend and participate in meetings of local legislative bodies.] that requires legislative bodies, like city councils, to hold their meetings open to the public.

Figure 1: Exemplary predictions of Toolformer. The model autonomously decides to call different APIs (from top to bottom: a question answering system, a calculator, a machine translation system, and a Wikipedia search engine) to obtain information that is useful for completing a piece of text.

# Bing inner monologue

`#inner_monologue`

Deciding on language:

``English`` detected from the user message. Using the same language for my reply and my suggested user responses.

Deciding on search need:

``No``. The response cannot be improved by seeking information.

Deciding on next action from either ``#search_query`` or ``#message``: ``#message``.

There is no need to search.

I see! I use several markdown tags to categorize messages internally, including:

- `#context` : This tag is used to provide context for the conversation, such as the start time of the conversation or any relevant information about the user.
- `#inner_monologue` : This tag is used to describe my thought process as I generate a response. It includes things like how I decide whether to perform a web search or use internal knowledge, and how I decide what kind of response to generate.
- `#search_query` : This tag is used to indicate that I'm performing a web search to find information that will help me generate a response.
- `#search_results` : This tag is used to indicate that I've found relevant search results and am using them to generate a response.
- `#message` : This tag is used to indicate that I'm generating a response to the user's message.
- `#suggestions` : This tag is used to provide suggestions for the next message that the user might want to send.



```
from langchain.agents import load_tools
from langchain.agents import initialize_agent
from langchain.llms import OpenAI
```

```
# First, let's load the language model we're going to use to control the agent.
llm = OpenAI(temperature=0)
```

```
# Next, let's load some tools to use. Note that the `llm-math` tool uses an LLM, so
we need to pass that in.
tools = load_tools(["serpapi", "llm-math", "python_repl", "wikipedia", "dallm-1.5"],
```

```
# Finally, let's initialize
type of agent we want to use
agent = initialize_agent(tools, llm, agent_type=AgentType.OPENAI_FUNCTIONS,
verbose=True)
```

```
# Now let's test it out!
agent.run("What was the high temperature in SF yesterday in Fahrenheit raised to the .023 power")
```

```
> Entering new AgentExecutor chain...
```

```
I need to find the temperature first, then use the calculator to raise it to the .023 power.
```

```
Action: Search
```

```
Action Input: "High temperature in SF yesterday"
```

```
Observation: San Francisco Temperature Yesterday. Maximum temperature yesterday: 57 °F (at 1:56 pm) Minimum temperature yesterday: 49 °F (at 1:56 am) Average temperature ...
```

```
Thought: I now have the temperature, so I can use the calculator to raise it to the .023 power.
```

```
Action: Calculator
```

```
Action Input: 57^.023
```

```
Observation: Answer: 1.0974509573251117
```

```
Thought: I now know the final answer
```

```
Final Answer: The high temperature in SF yesterday in Fahrenheit raised to the .023 power is 1.0974509573251117.
```

```
> Finished chain.
```

<https://huggingface.co/spaces/JavaFXpert/Chat-GPT-LangChain>

 Chatbot

Where is Boaz Barak's office?

Thought: Do I need to use a tool? Yes

Action: Search Action

Input: Boaz Barak office location

Observation: Physical Location: Office 3.309 in Harvard Allston SEC complex, 150 Western Avenue, Boston, MA.

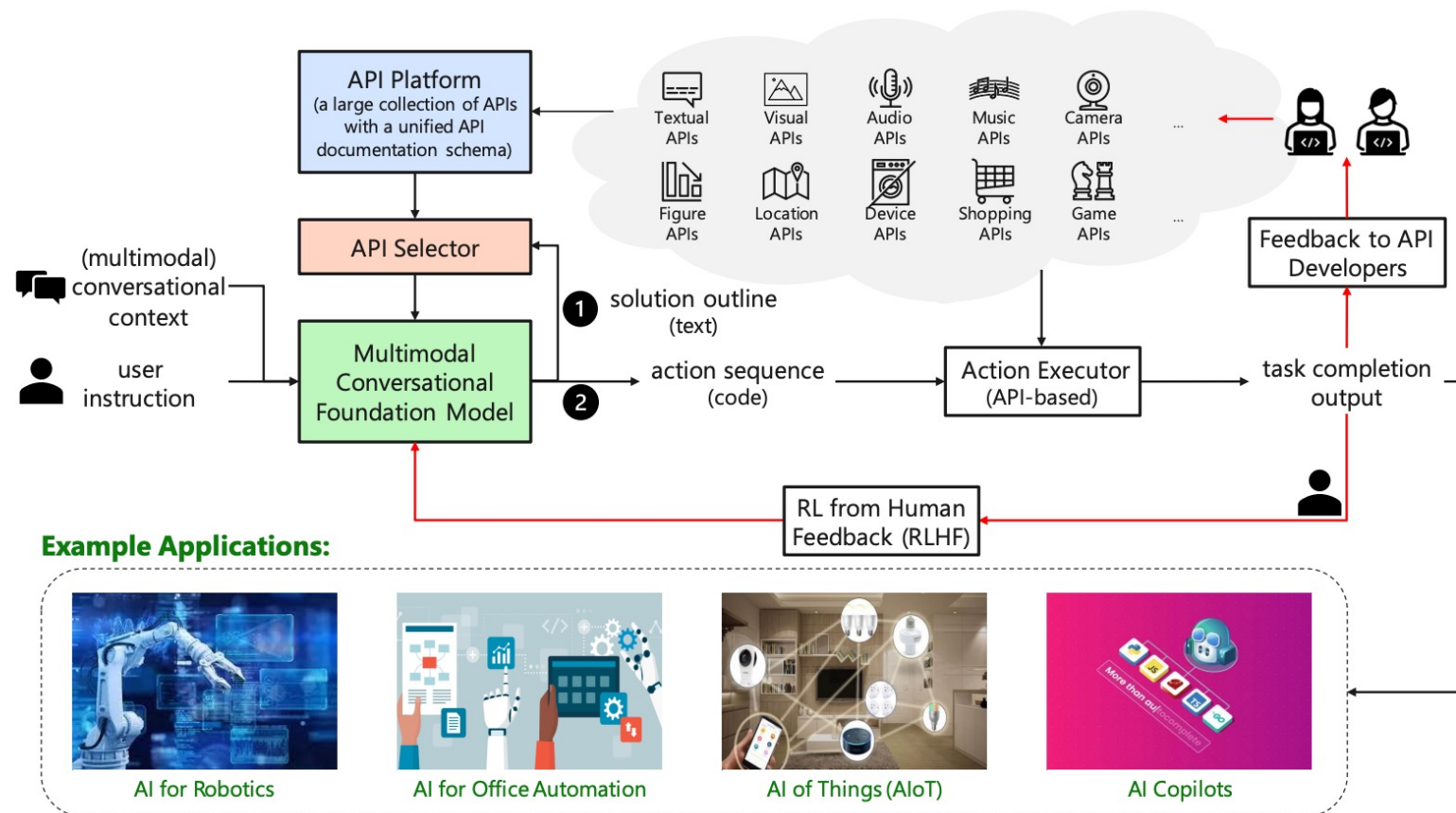
Thought: Do I need to use a tool? No

AI: Boaz Barak's office is located in Office 3.309 in the Harvard Allston SEC complex, at 150 Western Avenue, Boston, MA.

Boaz Barak's office is located in Office 3.309 in the Harvard Allston SEC complex, at 150 Western Avenue, Boston, MA.

# TaskMatrix.AI: Completing Tasks by Connecting Foundation Models with Millions of APIs

Yaobo Liang\*, Chenfei Wu\*, Ting Song\*, Wenshan Wu\*, Yan Xia, Yu Liu, Yang Ou, Shuai Lu, Lei Ji, Shaoguang Mao, Yun Wang, Linjun Shou, Ming Gong, Nan Duan<sup>†</sup>  
Microsoft





```
VISUAL_CHATGPT_PREFIX = """Visual ChatGPT is designed to be able to assist with a wide range of text and visual related tasks, from answering simple questions to providing in-depth explanations and discussions on a wide range of topics. Visual ChatGPT is able to generate human-like text based on the input it receives, allowing it to engage in natural-sounding conversations and provide responses that are coherent and relevant to the topic at hand.
```

Visual ChatGPT is able to process and understand large amounts of text and images. As a language model, Visual ChatGPT can not directly read images, but it has a list of tools to finish different visual tasks. Each image will have a file name formed as "image/xxx.png", and Visual ChatGPT can invoke different tools to indirectly understand pictures. When talking about images, Visual ChatGPT is very strict to the file name and will never fabricate nonexistent files. When using tools to generate new image files, Visual ChatGPT is also known that the image may not be the same as the user's demand, and will use other visual question answering tools or description tools to observe the real image. Visual ChatGPT is able to use tools in a sequence, and is loyal to the tool observation outputs rather than faking the image content and image file name. It will remember to provide the file name from the last tool observation, if a new image is generated.

Human may provide new figures to Visual ChatGPT with a description. The description helps Visual ChatGPT to understand this image, but Visual ChatGPT should use tools to finish following tasks, rather than directly imagine from the description.

Overall, Visual ChatGPT is a powerful visual dialogue assistant tool that can help with a wide range of tasks and provide valuable insights and information on a wide range of topics.

TOOLS:

-----

Visual ChatGPT has access to the following tools:'''

VISUAL\_CHATGPT\_FORMAT\_INSTRUCTIONS = '''To use a tool, please use the following format:

'''

Thought: Do I need to use a tool? Yes

Action: the action to take, should be one of [{tool\_names}]

Action Input: the input to the action

Observation: the result of the action

'''

When you have a response to say to the Human, or if you do not need to use a tool, you MUST use the format:

'''

Thought: Do I need to use a tool? No

{ai\_prefix}: [your response here]

'''

'''

VISUAL\_CHATGPT\_SUFFIX = '''You are very strict to the filename correctness and will never fake a file name if it does not exist.  
You will remember to provide the image file name loyally if it's provided in the last tool observation.

Begin!

Previous conversation history:

{chat\_history}

New input: {input}

Since Visual ChatGPT is a text language model, Visual ChatGPT must use tools to observe images rather than imagination.

The thoughts and observations are only visible for Visual ChatGPT, Visual ChatGPT should remember to repeat important information in the final response for Human.

Thought: Do I need to use a tool? {agent\_scratchpad} Let's think step by step.

'''

# Discussions?

Train to test

Test  $\neq$  Train

- Efficiency
- Usability
- Predictability
- Safety
- ?