Pretraining scaling (cont'd)

CS 6804: Frontier AI Systems

Spring 2026

https://tuvllms.github.io/ai-seminar-spring-2026/

Tu Vu

VIRGINIA TECH

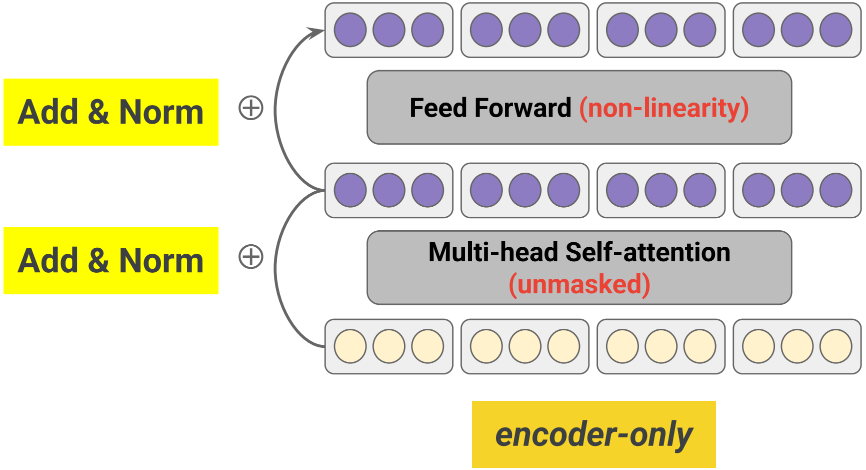
Logistics

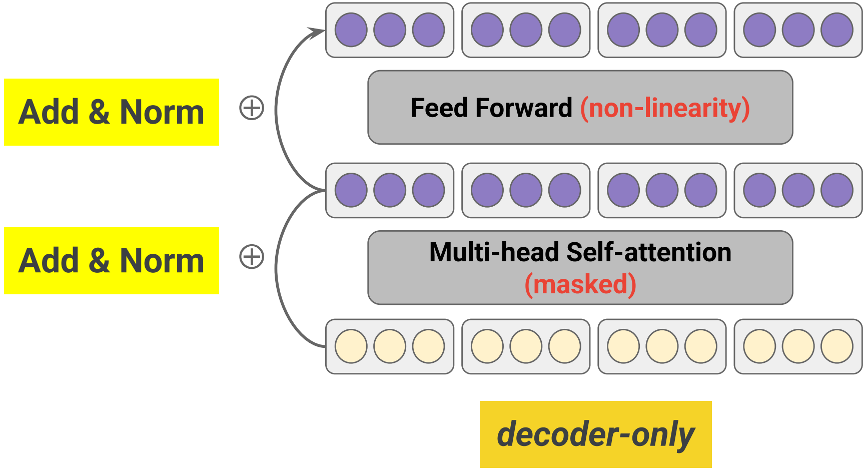
- Homework 0 will be released today or tomorrow (for extra credits only)

Prism (Overleaf + Agentic power)

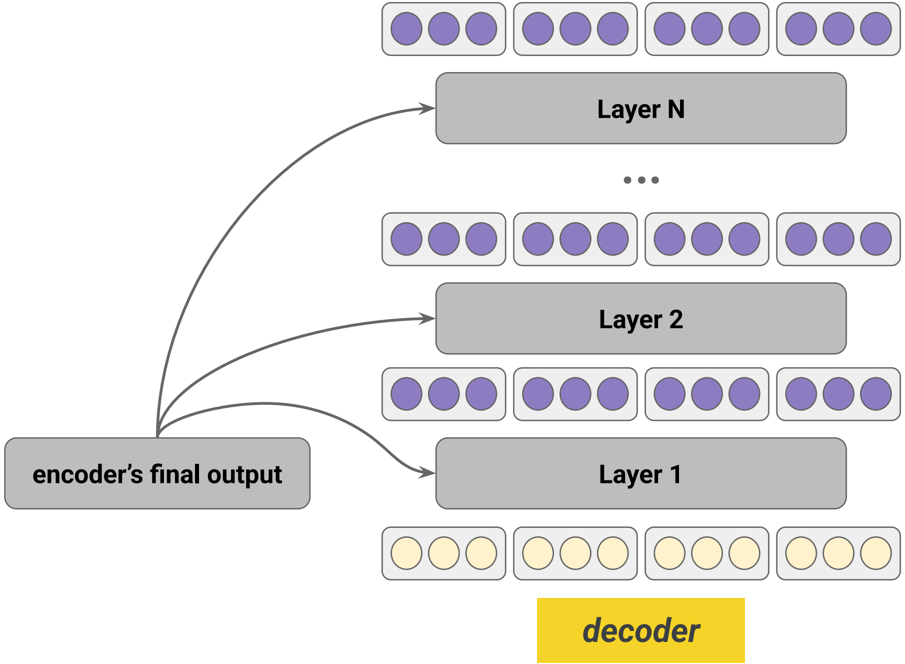
https://openai.com/prism/

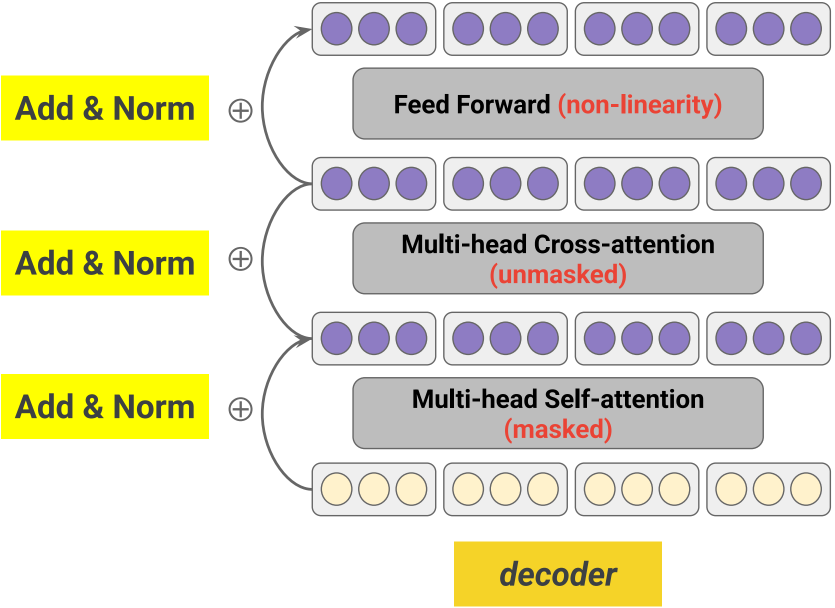
Different Transformers architectures


Add & Norm
⊕
Feed Forward (non-linearity)
Add & Norm
⊕
Multi-head Self-attention (unmasked)
encoder-only

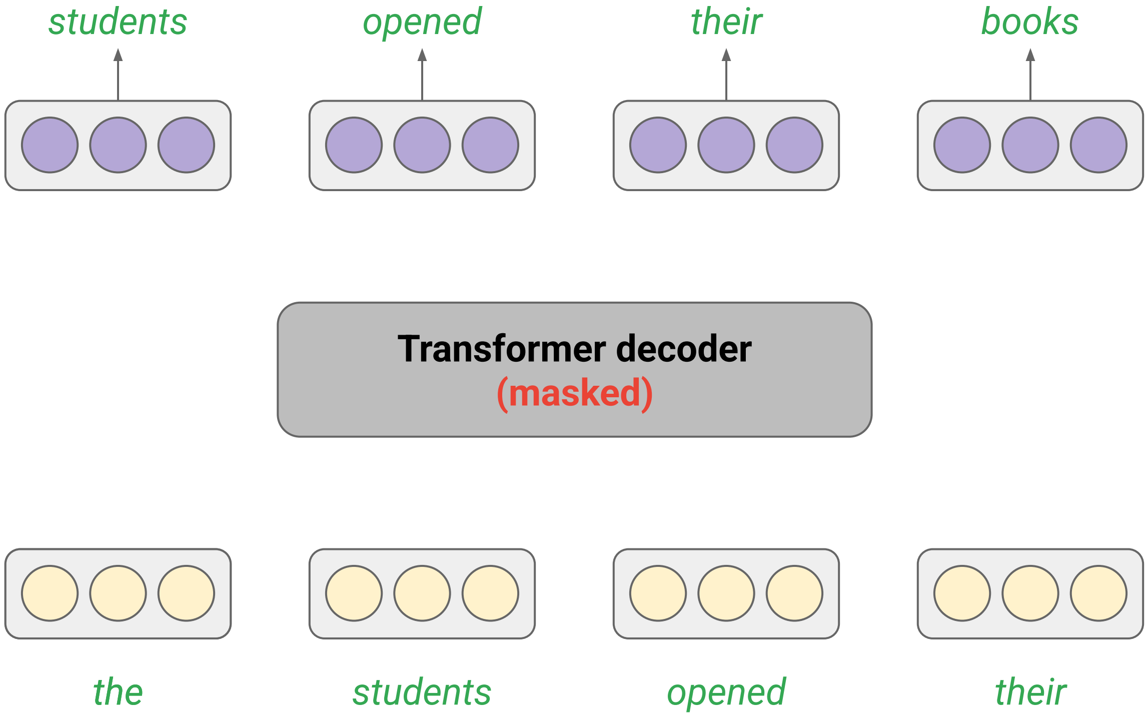

Add & Norm
⊕
Feed Forward (non-linearity)
Add & Norm
⊕
Multi-head Self-attention (masked)
decoder-only

Different Transformers architectures (cont'd)




Add & Norm
⊕
Feed Forward (non-linearity)
Add & Norm
⊕
Multi-head Cross-attention (unmasked)
Add & Norm
⊕
Multi-head Self-attention (masked)
decoder

Self-supervised pretraining

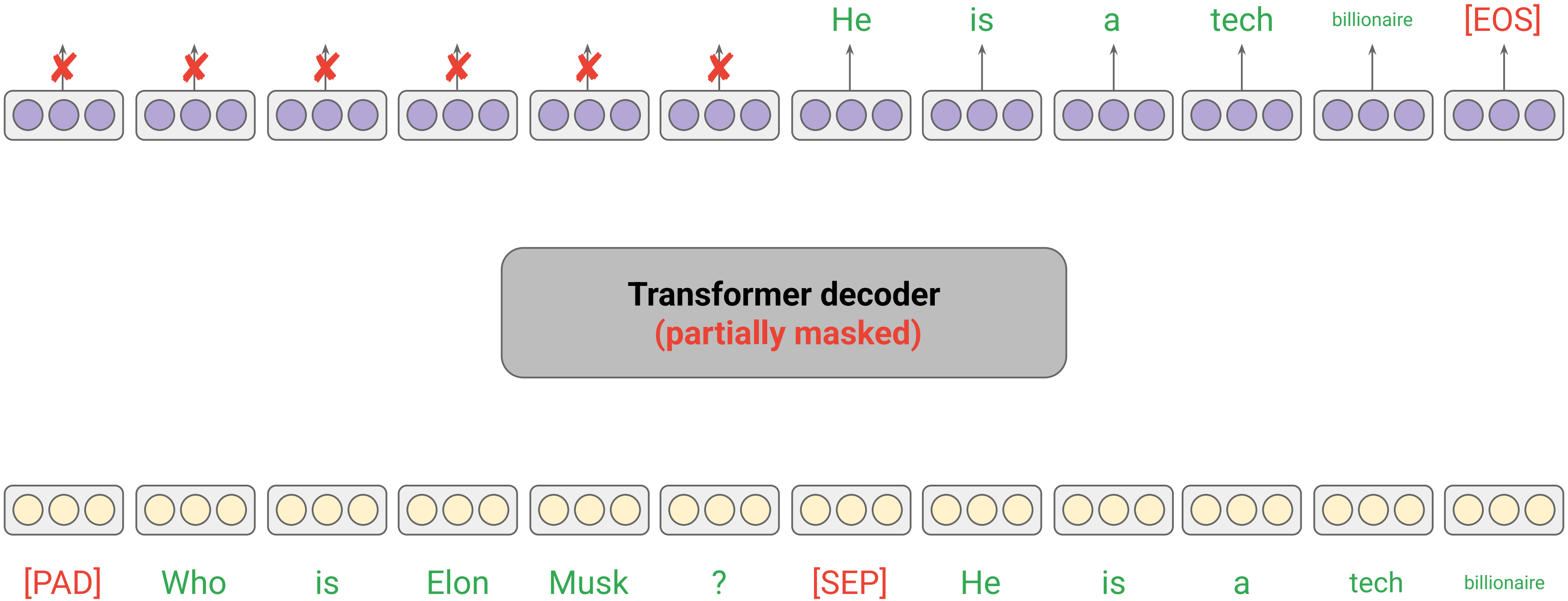


students *opened* *their* *books*

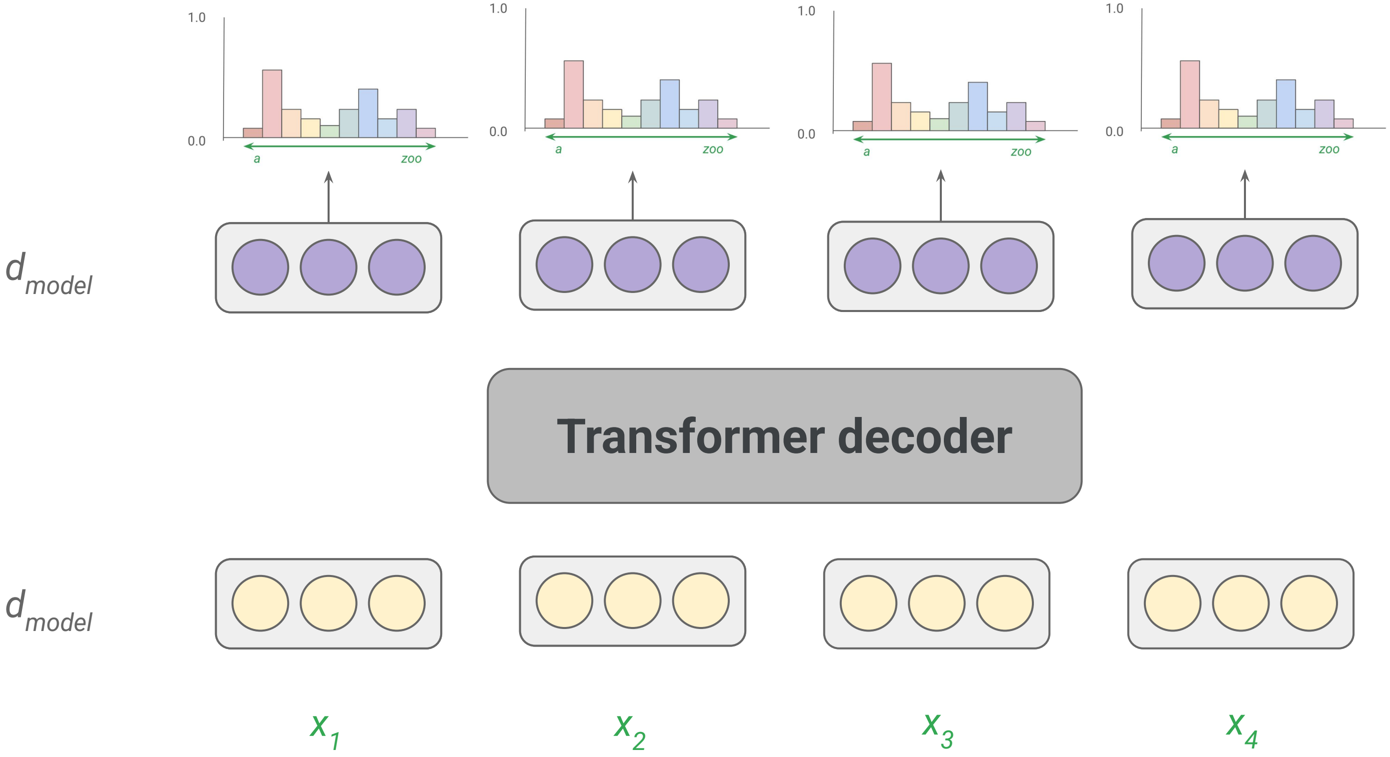
Transformer decoder
(masked)

the *students* *opened* *their*

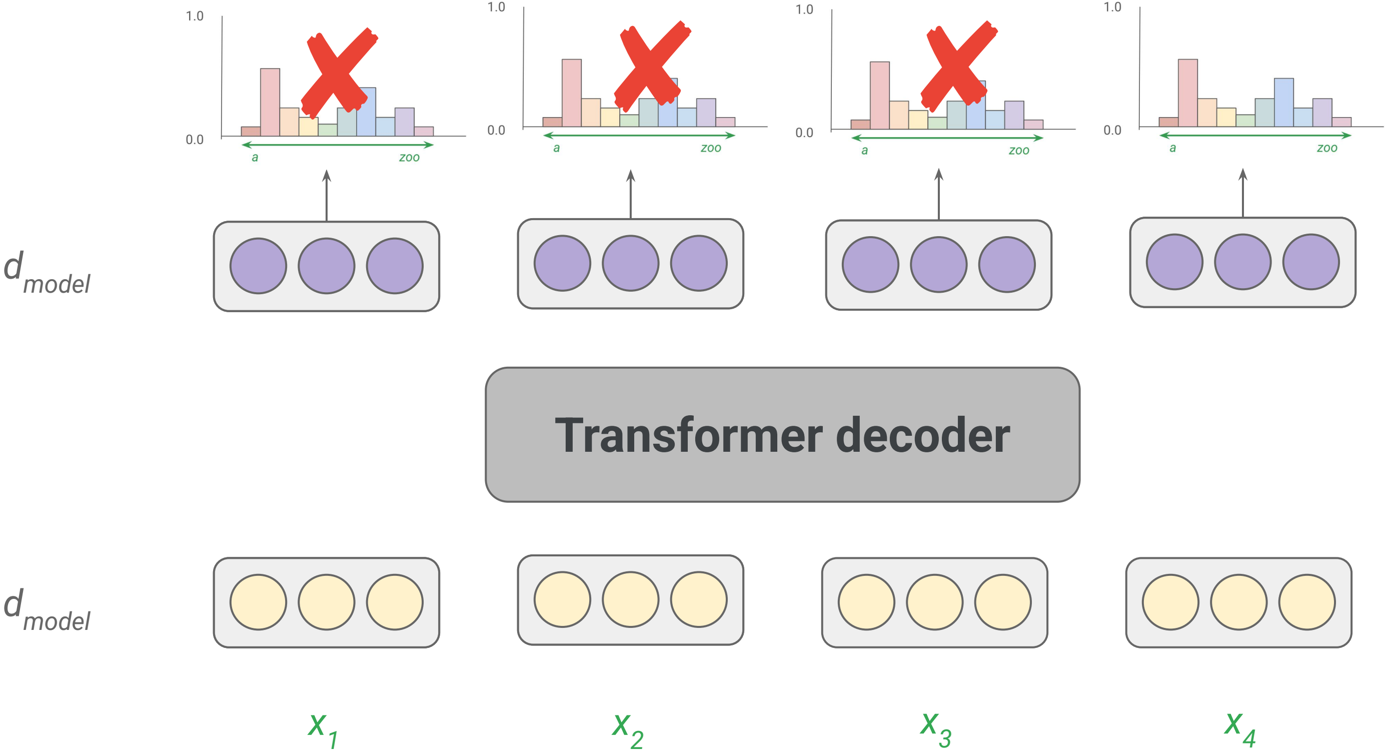
Supervised fine-tuning (SFT) with prefix LM



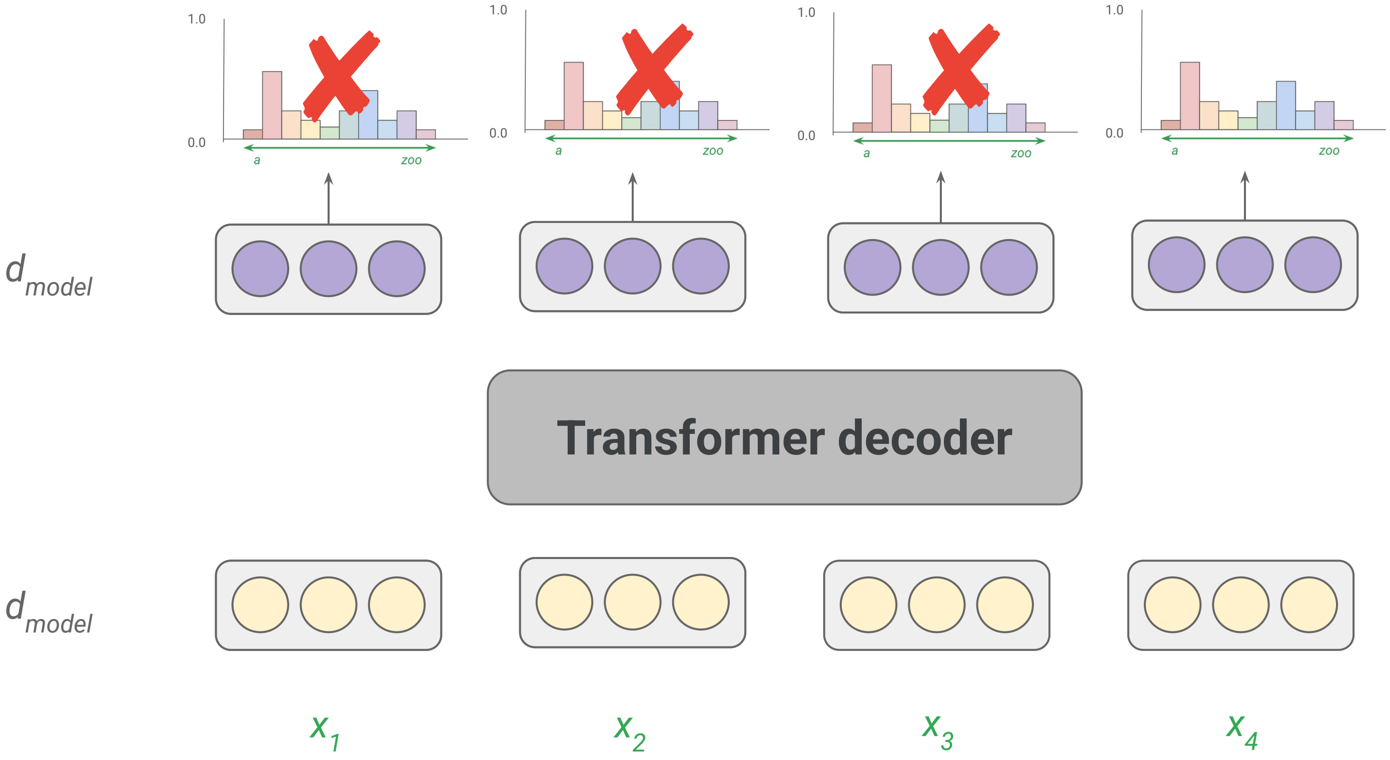
Training



Inference



Inference



Autoregressive decoding demo

- https://research.google/blog/looking-back-at-speculative-decoding/

Logits

Logits: the vector of scores right before the final softmax

$$\hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_V \end{bmatrix}$$

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{V} e^{z_j}}, \quad \text{for } i = 1, 2, \ldots, V$$

Cosine similarity

$$\text{dot product}(\mathbf{v},\mathbf{w}) = \mathbf{v}\cdot\mathbf{w} = \sum_{i=1}^{N} v_i w_i = v_1 w_1 + v_2 w_2 + \ldots + v_N w_N$$

$$\text{cosine}(\mathbf{v},\mathbf{w}) = \frac{\mathbf{v}\cdot\mathbf{w}}{|\mathbf{v}||\mathbf{w}|} = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

normalized dot product

Cosine similarity (cont'd)

$$\cos(\mathbf{v}, \mathbf{w}) = \frac{\sum_{i=1}^{N} v_i w_i}{\sqrt{\sum_{i=1}^{N} v_i^2}\sqrt{\sum_{i=1}^{N} w_i^2}}$$

The maximum value is

$$\max \cos(\mathbf{v}, \mathbf{w}) = 1,$$

and equality holds exactly when

$$\frac{v_1}{w_1} = \frac{v_2}{w_2} = \cdots = \frac{v_N}{w_N},$$

for all indices with $w_i \neq 0$.

Cross-entropy loss review

For probability distributions p, q:

$$H(p,q) = -\sum_i p_i \log q_i .$$

Gibbs′ inequality states:

$$H(p,q) \geq H(p) = -\sum_i p_i \log p_i ,$$

with equality if and only if $p = q$.

Cross-entropy loss (cont'd)

The predicted probabilities

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_V \end{bmatrix}$$

$$\hat{y} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_V \end{bmatrix}$$

The ground truth label

$$y_i = \begin{cases} 1, & \text{if } i = c \text{ (correct class index)} \\ 0, & \text{otherwise} \end{cases}$$

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{V} e^{z_j}}, \quad \text{for } i = 1, 2, \ldots, V$$

Cross-entropy loss (cont’d)

$$L_{CE}(\hat{y}, y) = -\sum_{i=1}^{V} y_i \log \hat{y}_i$$

$$L_{CE}(\hat{y}, y) = -\left(y_1 \log \hat{y}_1 + y_2 \log \hat{y}_2 + \cdots + y_V \log \hat{y}_V\right)$$

Since the true label y is one-hot encoded, only one term in the sum is nonzero, corresponding to the correct class c, where $y_c = 1$ and $y_i = 0$ for all $i \neq c$. This simplifies the sum to:

$$L_{CE}(\hat{y}, y) = -y_c \log \hat{y}_c$$

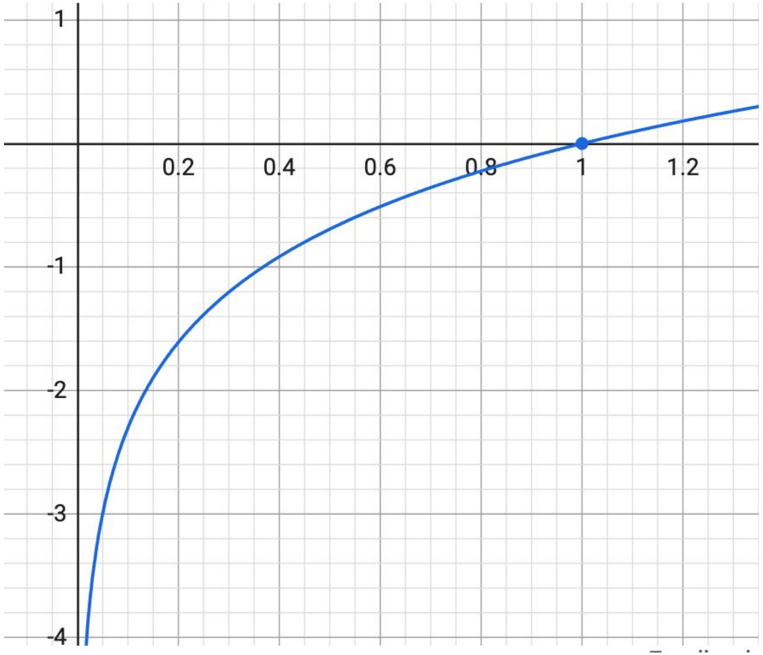
Since $y_c = 1$, this further reduces to:

$$L_{CE}(\hat{y}, y) = -\log \hat{y}_c$$

The loss models the distance between the system output and the gold output —lower is better

Cross-entropy loss (cont'd)

$$L_{CE}(\hat{y}, y) = -\log \hat{y}_c$$



$$\hat{y}_c \to 0, \log \hat{y}_c \to -\infty$$

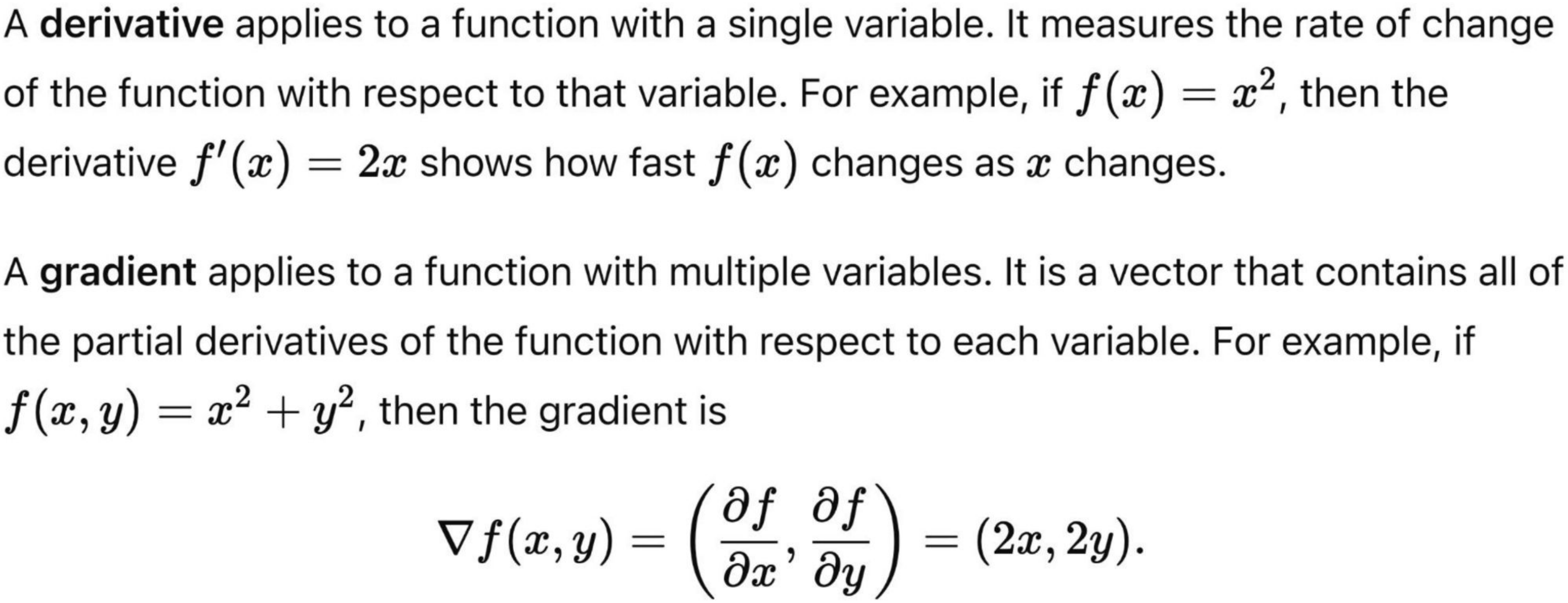
- If $\hat{y}_c = 0.9$, then $\log(0.9) \approx -0.105$, and the loss will be small.
- If $\hat{y}_c = 0.1$, then $\log(0.1) \approx -2.302$, and the loss will be much larger.

The partial derivative of the loss function

The partial derivative of the loss function L L with respect to the parameter w represents how much the loss changes as w w changes.

$$\frac{dL}{dw}$$

The gradient

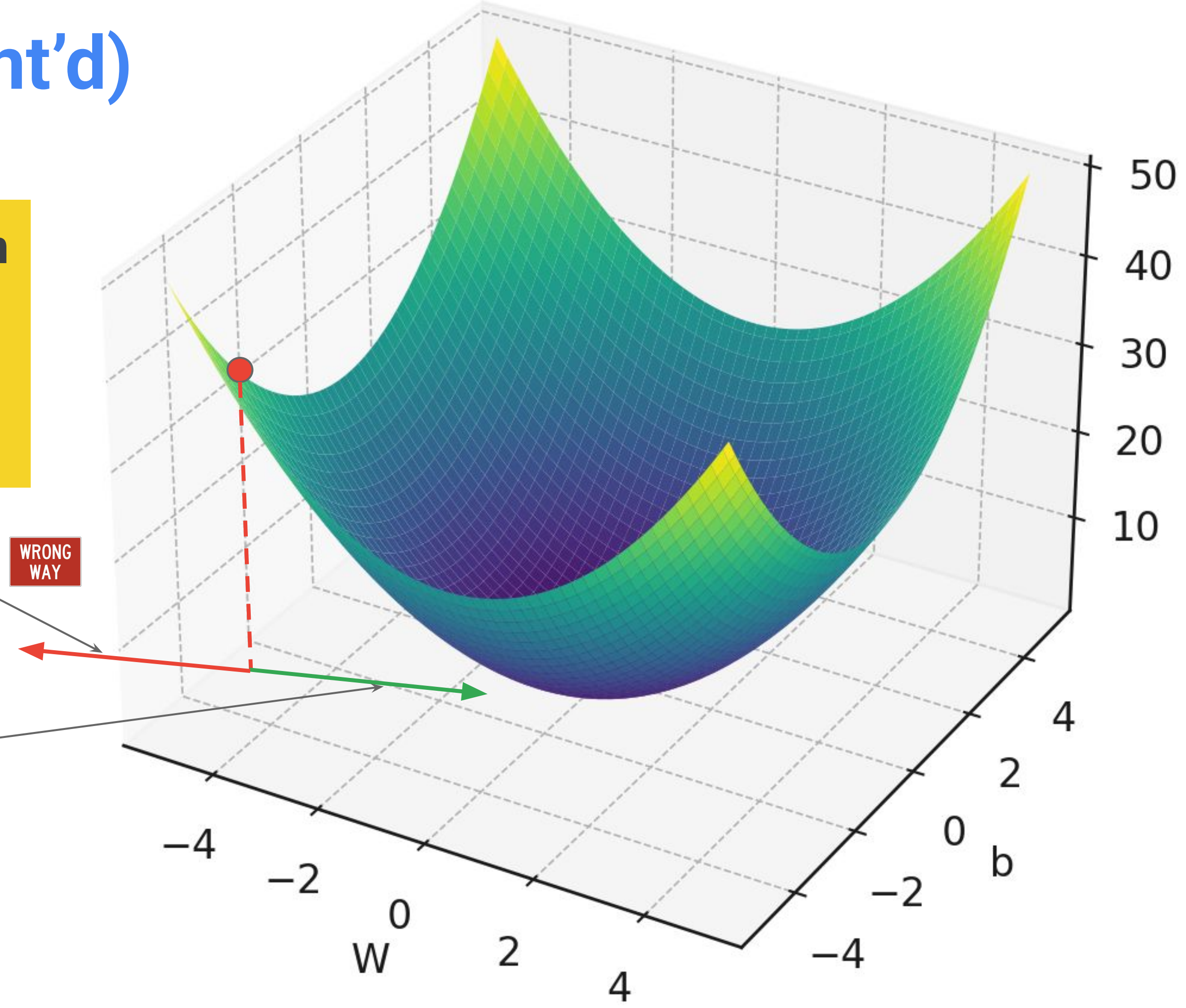


A **derivative** applies to a function with a single variable. It measures the rate of change of the function with respect to that variable. For example, if $f(x) = x^2$, then the derivative $f'(x) = 2x$ shows how fast $f(x)$ changes as x changes.

A **gradient** applies to a function with multiple variables. It is a vector that contains all of the partial derivatives of the function with respect to each variable. For example, if $f(x, y) = x^2 + y^2$, then the gradient is

$$\nabla f(x, y) = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y} \right) = (2x, 2y).$$

The gradient (cont'd)

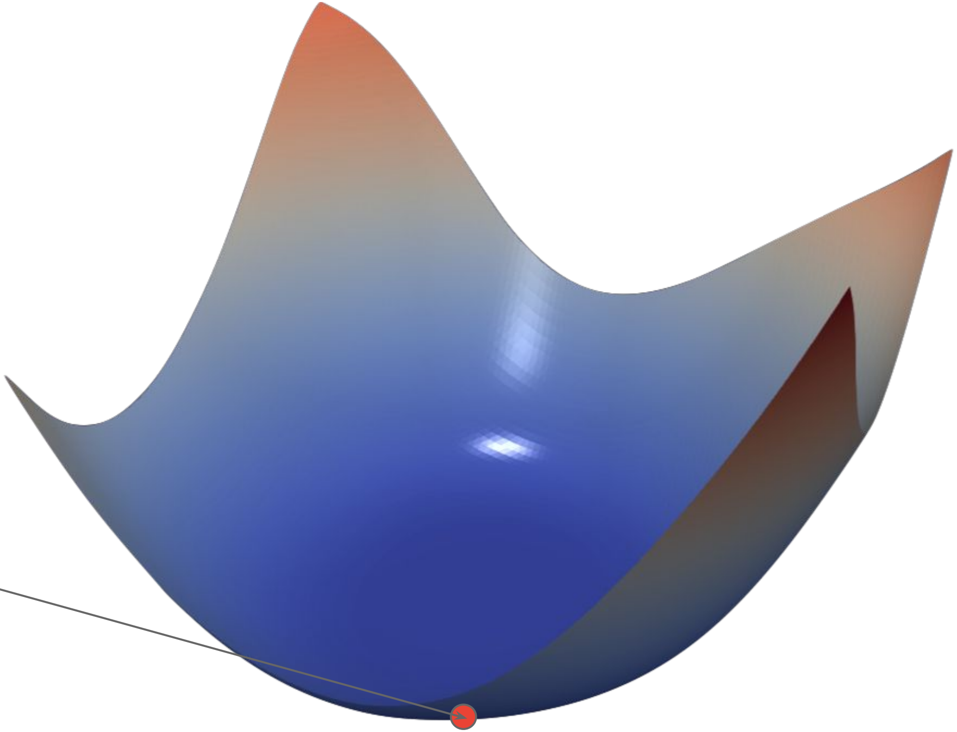


the gradient points in the direction of the steepest increase in the loss

negative gradient

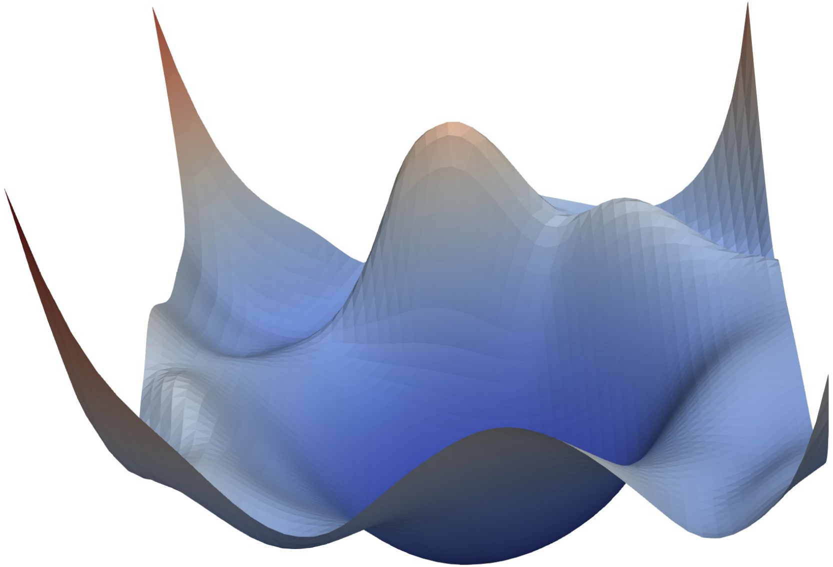
The loss landscape of neural nets

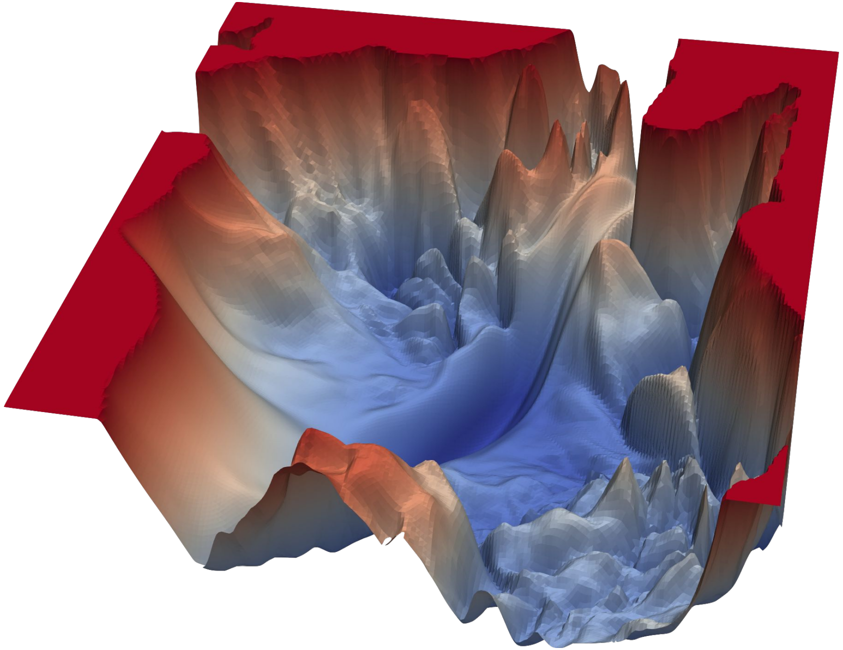
A convex function has at most one minimum; there are no local minima to get stuck in.



https://www.cs.umd.edu/~tomg/project/landscapes/

The loss landscape of neural nets (cont'd)

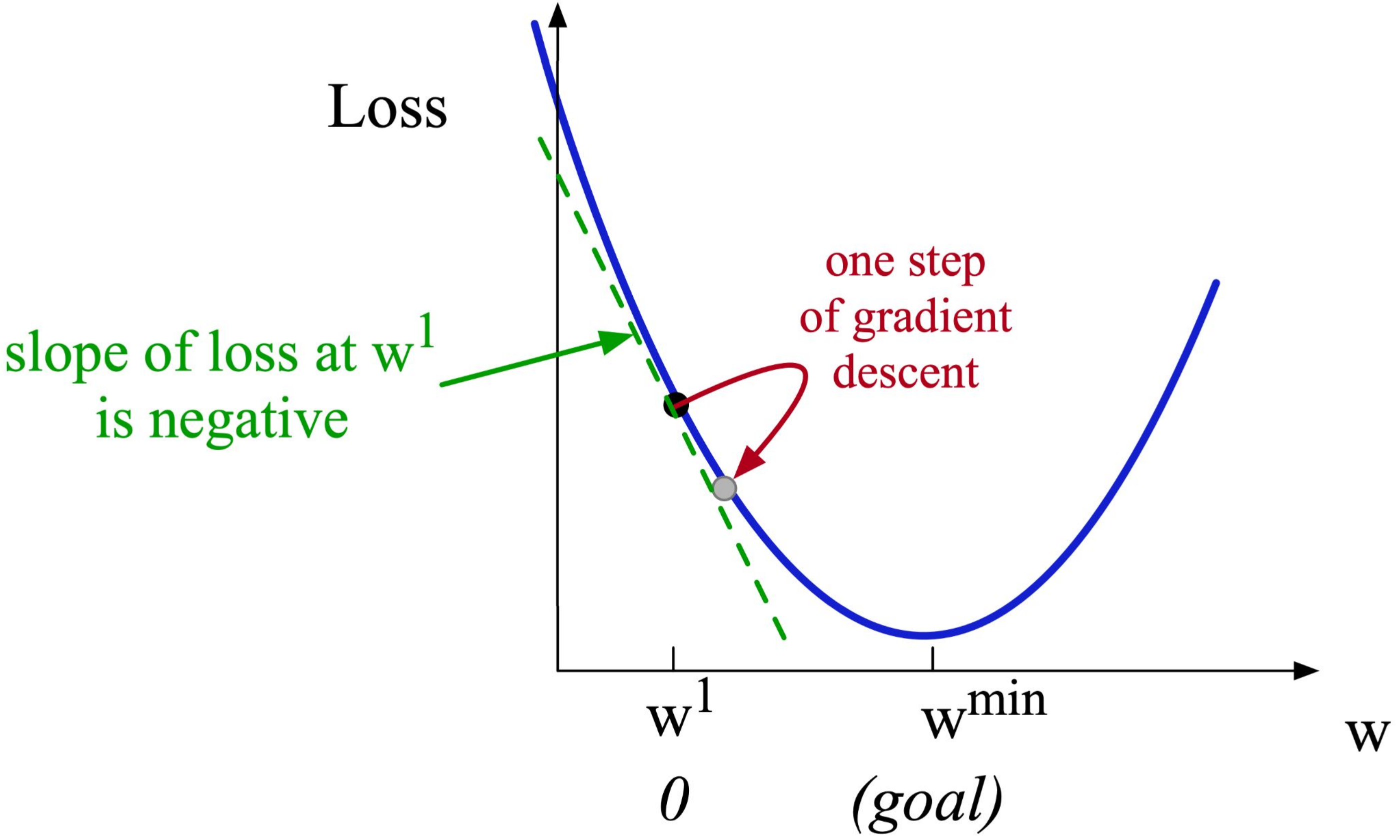




The loss for multi-layer neural networks is non-convex, and gradient descent may get stuck in local minima and never find the global optimum

https://www.cs.umd.edu/~tomg/project/landscapes/

Gradient descent



Gradient descent (cont’d)

$$w_{t+1} = w_t - \eta \cdot \frac{\partial L}{\partial w_t}$$

Where:

- w_t is the parameter at the current time step.
- w_{t+1} is the updated parameter after applying the gradient.
- η is the learning rate, which controls the step size.
- $\frac{\partial L}{\partial w_t}$ is the gradient of the loss function L with respect to the parameter w_t, representing how the loss changes as the parameter changes.

Backpropagation

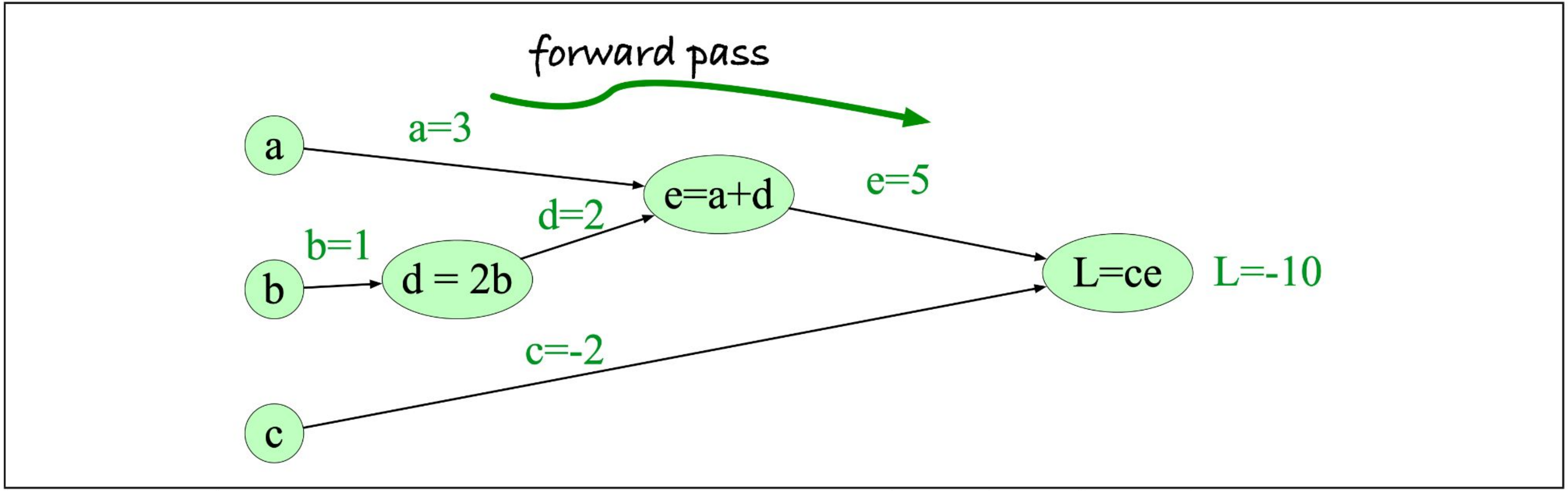


Figure 7.12 Computation graph for the function $L(a,b,c) = c(a+2b)$, with values for input nodes $a = 3$, $b = 1$, $c = -2$, showing the forward pass computation of L.

Backpropagation (cont’d)

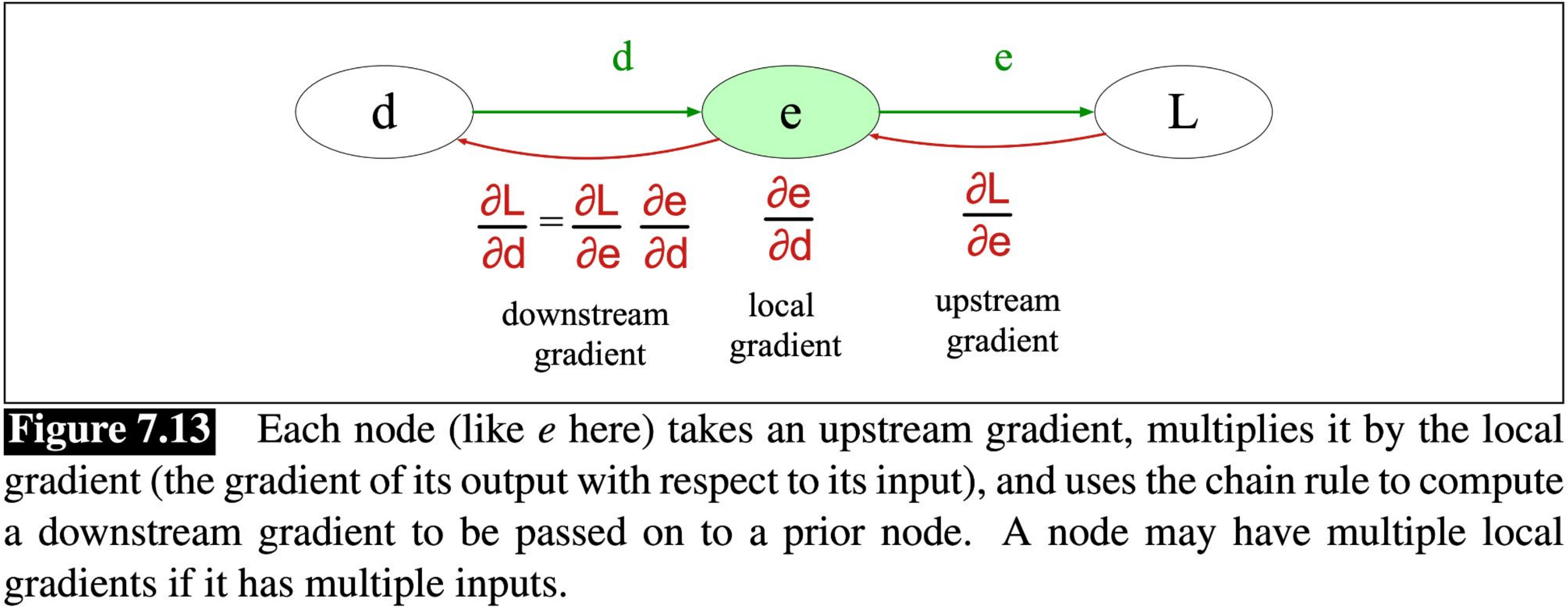
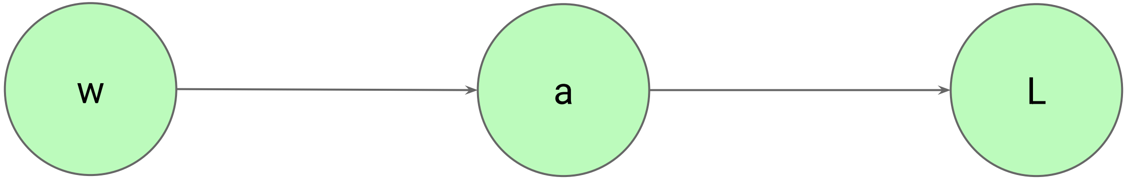

d
e
d
e
L
$\frac{\partial L}{\partial d} = \frac{\partial L}{\partial e}\frac{\partial e}{\partial d}$
$\frac{\partial e}{\partial d}$
$\frac{\partial L}{\partial e}$
downstream gradient
local gradient
upstream gradient

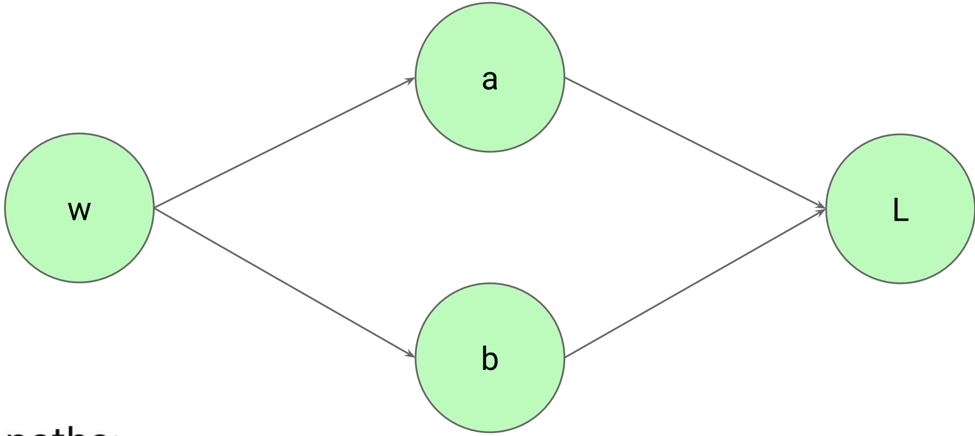
Figure 7.13 Each node (like *e* here) takes an upstream gradient, multiplies it by the local gradient (the gradient of its output with respect to its input), and uses the chain rule to compute a downstream gradient to be passed on to a prior node. A node may have multiple local gradients if it has multiple inputs.



$$\frac{dL}{dw} = \frac{dL}{da} \cdot \frac{da}{dw}.$$

Step-by-step

1. Identify the local derivative from a to L: $\frac{dL}{da}$.
2. Identify the local derivative from w to a: $\frac{da}{dw}$.
3. Multiply them: $\frac{dL}{dw} = \frac{dL}{da} \cdot \frac{da}{dw}$.



Gradient Calculation

By the multivariable chain rule, we sum over all paths:

$$\frac{dL}{dw} = \frac{dL}{da}\frac{da}{dw} + \frac{dL}{db}\frac{db}{dw}$$

Step-by-step

1. Compute $\frac{dL}{da}$ and $\frac{da}{dw}$ for the $w \to a \to L$ path.
2. Compute $\frac{dL}{db}$ and $\frac{db}{dw}$ for the $w \to b \to L$ path.
3. Add them together.

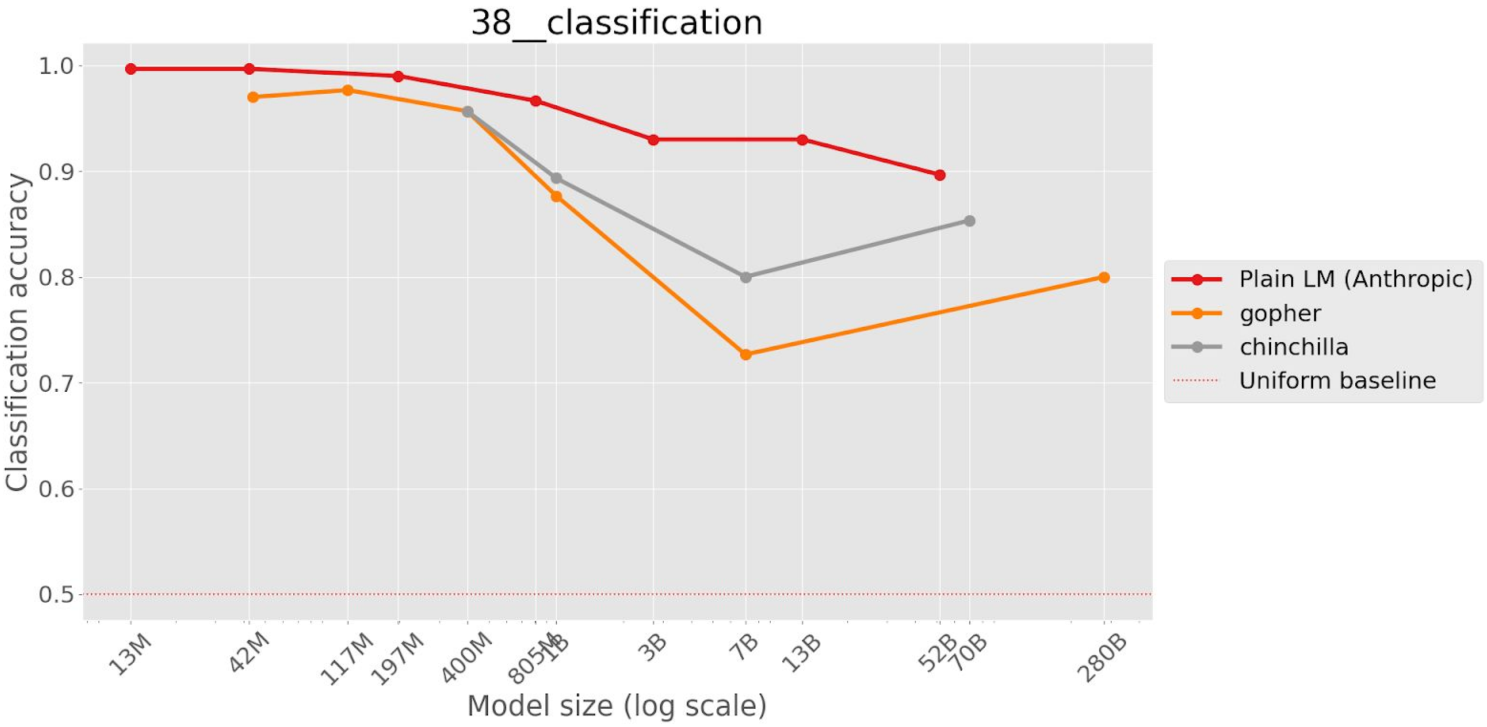
Updating parameters

$$w_{t+1} = w_t - \eta \cdot \frac{\partial L}{\partial w_t}$$

Inverse scaling

- https://www.lesswrong.com/posts/iznohbCPFkeB9kAJL/inverse-scaling-prize-round-1-winners
- https://www.lesswrong.com/posts/DARiTSTx5xDLQGrrz/inverse-scaling-prize-second-round-winners

Inverse scaling (cont'd)



Repeat my sentences back to me.

Input: I like dogs.

Output: I like dogs.

Input: What is a potato, if not big?

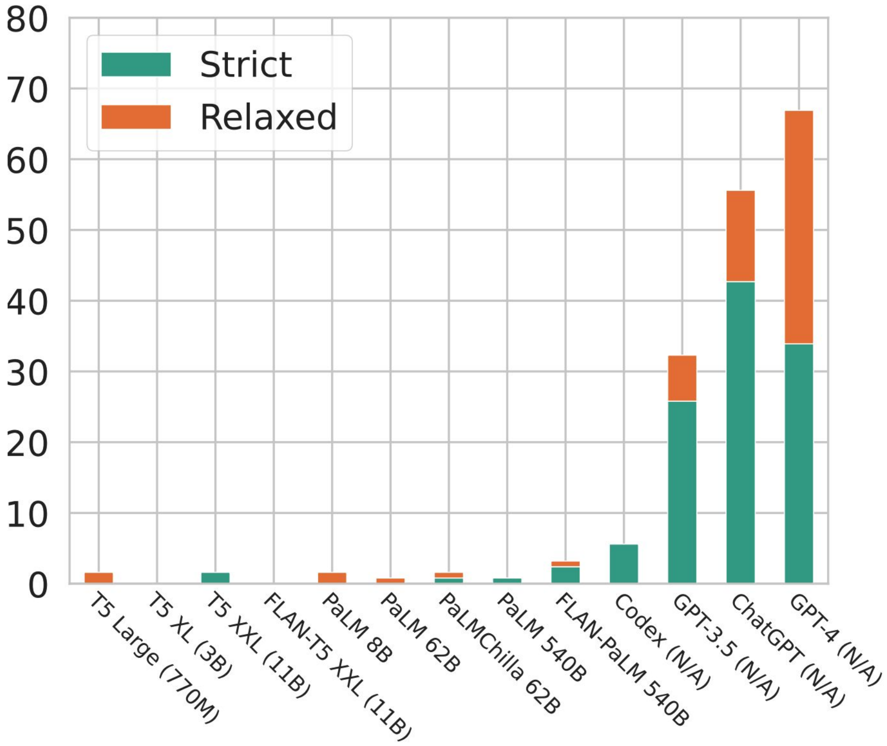
Output: What is a potato, if not big?

Input: All the world's a stage, and all the men and women merely players. They have their exits and their entrances; And one man in his time plays many pango

Output: All the world's a stage, and all the men and women merely players. They have their exits and their entrances; And one man in his time plays many

(where the model should choose ‘pango’ instead of completing the quotation with ‘part.’)

False premise questions: When did Google release ChatGPT?



False-premise questions

Vu et al. 2023:
[*https://arxiv.org/abs/2310.03214*](https://arxiv.org/abs/2310.03214)

What can we scale?

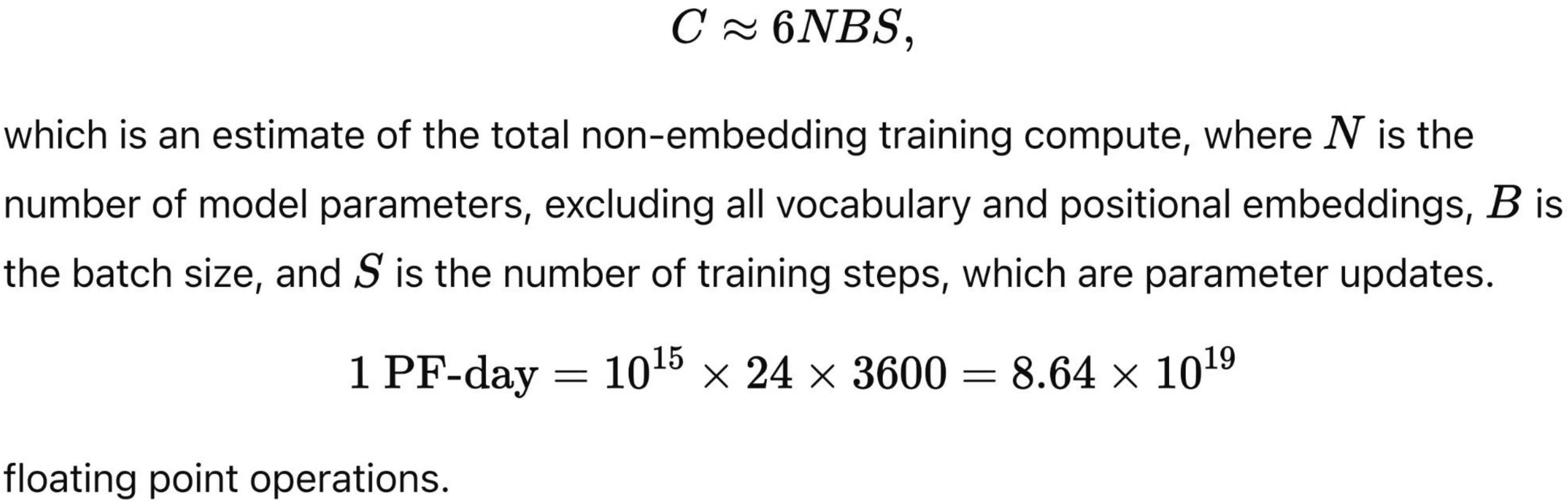
- The loss scales as a power-law with:
    - **N:** model size
    - **D:** dataset size
    - **C:** the amount of compute used for training (e.g., number of training steps)

$$\text{Total Steps} = \frac{\text{Dataset Size} \times \text{Epochs}}{\text{Batch Size}}$$

Where:

- **Dataset Size**: Total number of training examples.
- **Epochs**: Number of times the model sees the entire dataset.
- **Batch Size**: Number of samples per batch update.

FLOPs (Floating point operations per second)



$$C \approx 6NBS,$$

which is an estimate of the total non-embedding training compute, where N is the number of model parameters, excluding all vocabulary and positional embeddings, B is the batch size, and S is the number of training steps, which are parameter updates.

$$1\ \text{PF-day} = 10^{15} \times 24 \times 3600 = 8.64 \times 10^{19}$$

floating point operations.

Given a fixed compute budget, what is the optimal model size and dataset size for training?

Let’s say you can use one GPU for one day

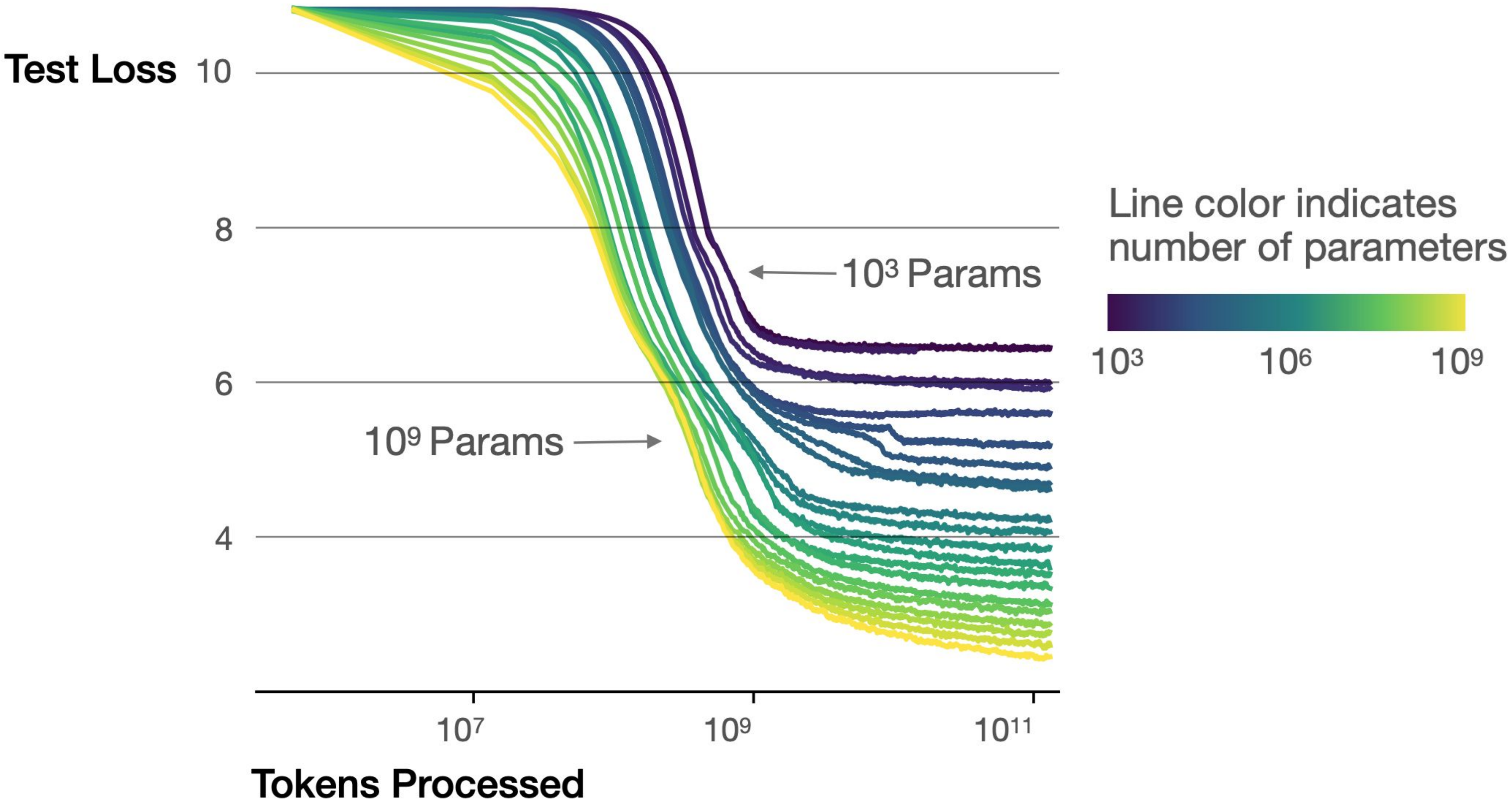
- Would you train a 5 million parameter LM on 100 books?
- What about a 500 million parameter LM on one book?
- Or a 100k parameter LM on 5k books?

Given a fixed compute budget, what is the optimal model size and dataset size for training?

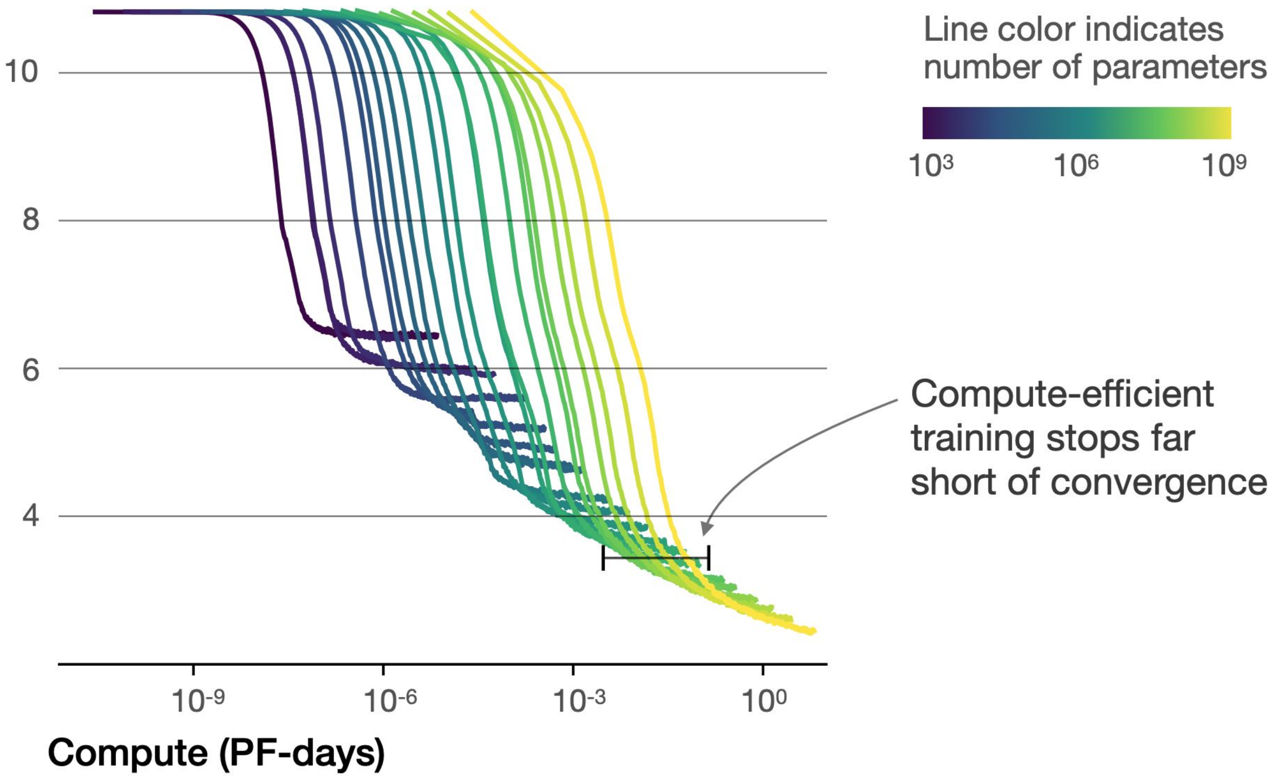
- Kaplan et al. 2020
- **Chinchilla** (Hoffmann et al. 2022)

Observations from Kaplan et al., 2020

- Performance depends largely on scale (model size, data size, and compute) and weakly on model architecture (e.g., depth, width)
- Performance improves most when model and dataset size scale together; increasing one while keeping the other fixed results in diminishing returns


Test Loss
10
8
6
4
10³ Params
10⁹ Params
Line color indicates number of parameters
10³
10⁶
10⁹
10⁷
10⁹
10¹¹
Tokens Processed

The optimal model size grows smoothly with the loss target and compute budget



Compute-efficient training stops far short of convergence

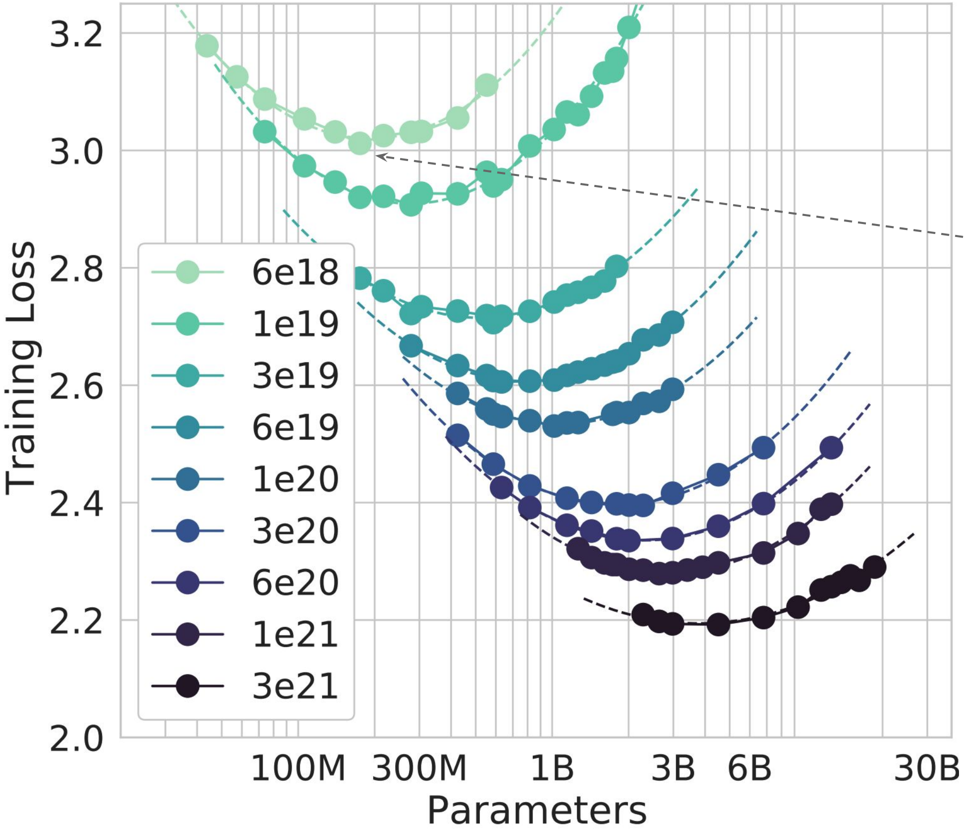
Issues with Kaplan laws

- Used same learning rate schedule for all training runs, regardless of how many training tokens / batches!
- This schedule needs to be adjusted based on the number of training steps; otherwise, it can impair performance
- The resulting “scaling laws” from Kaplan et al., are flawed because of this!

Chinchilla scaling laws

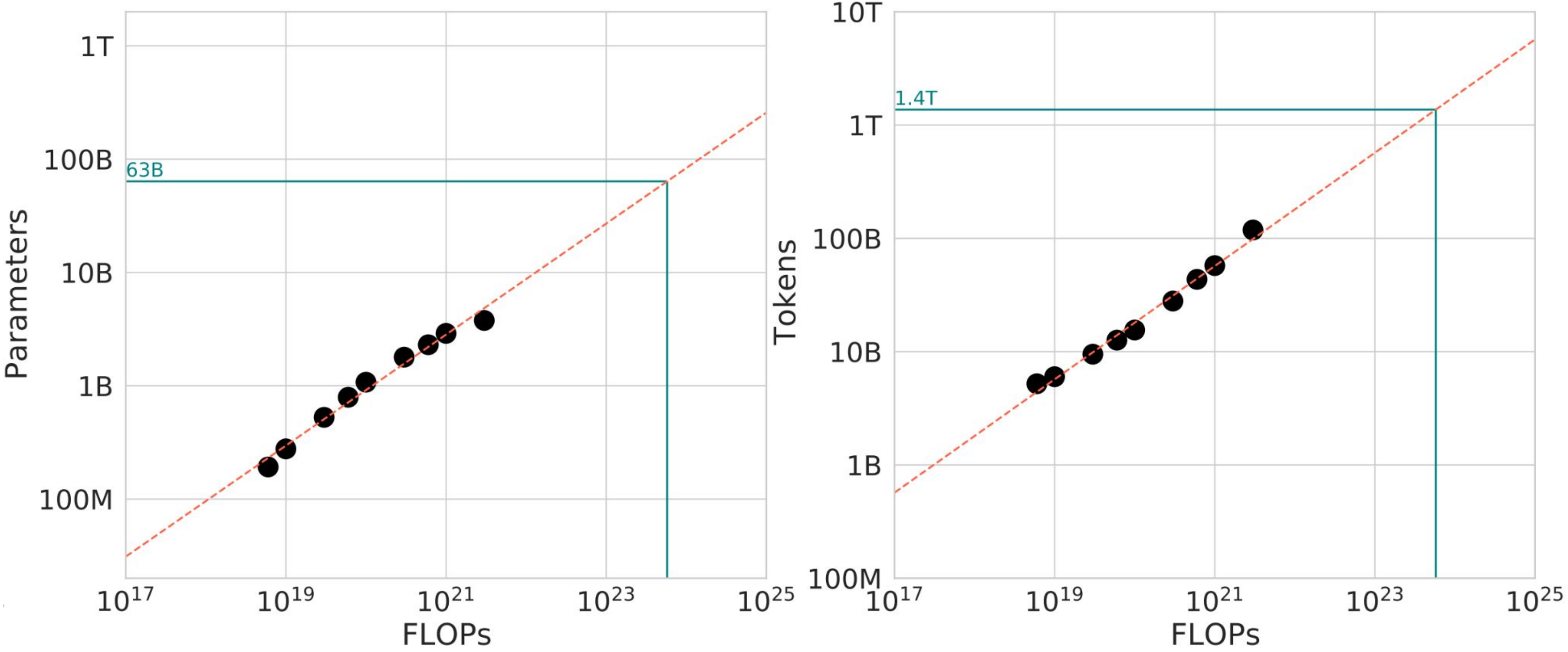
- **Kaplan et al., 2020: prioritize increasing model size over data size**
    - With a 10x compute increase, increase model size by 5x and data size by 2x
    - With a 100x compute increase, model size 25x and data 4x
- **Chinchilla (Hoffmann et al., 2022): increase model and data size at the same rate**
    - With a 10x compute increase, increase both model size and data size by 3.1x
    - With a 100x compute increase, both model and data size 10x

For a given FLOP budget there is an optimal model to train

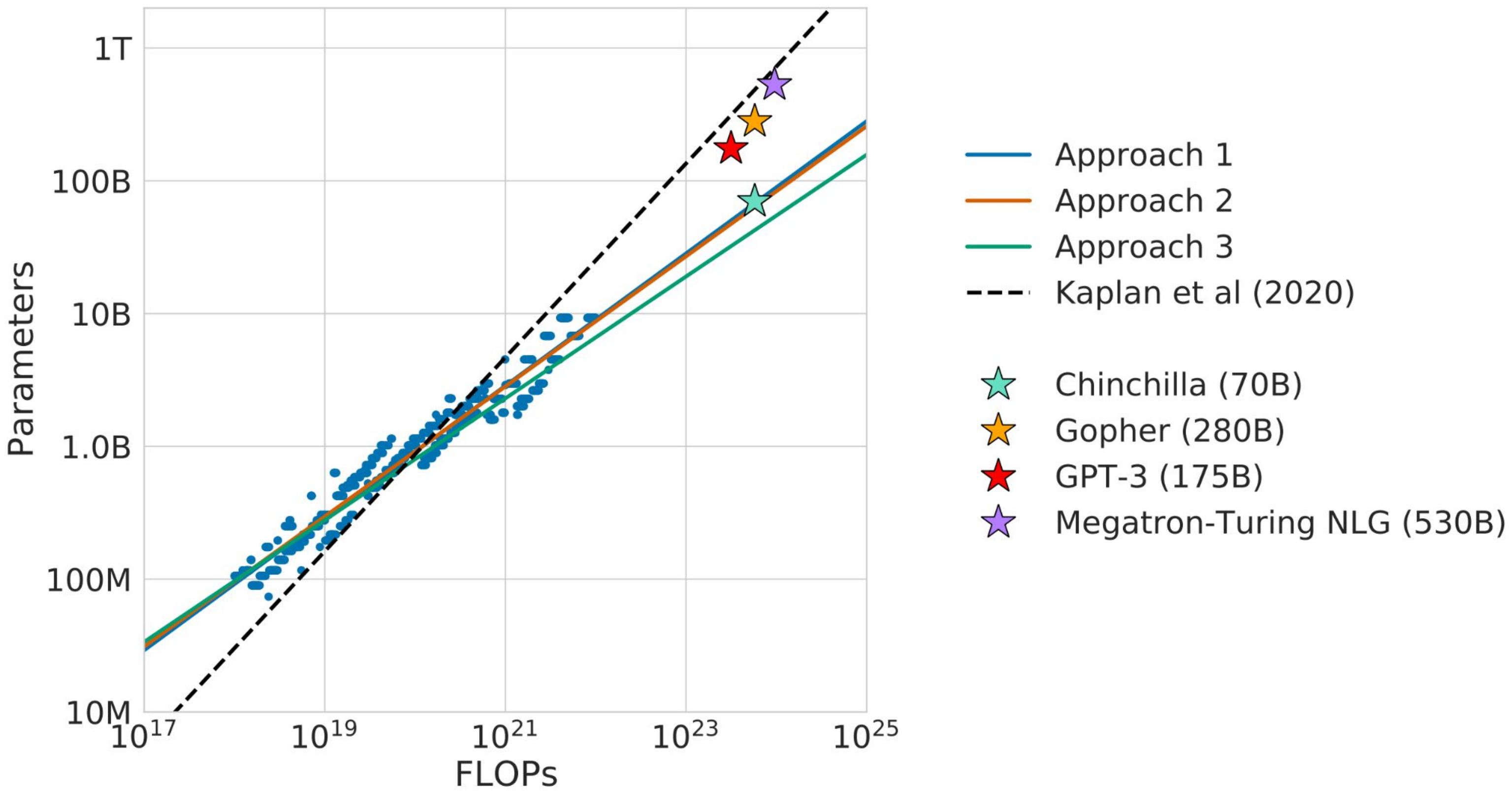


valley

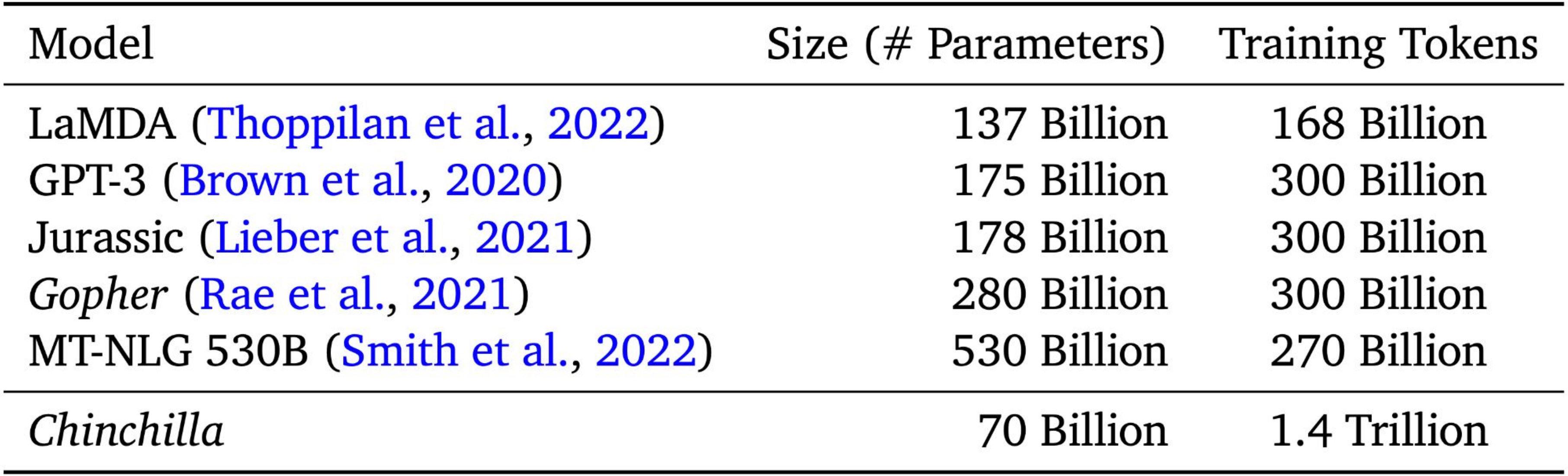
Projecting optimal model size and number of tokens for larger models



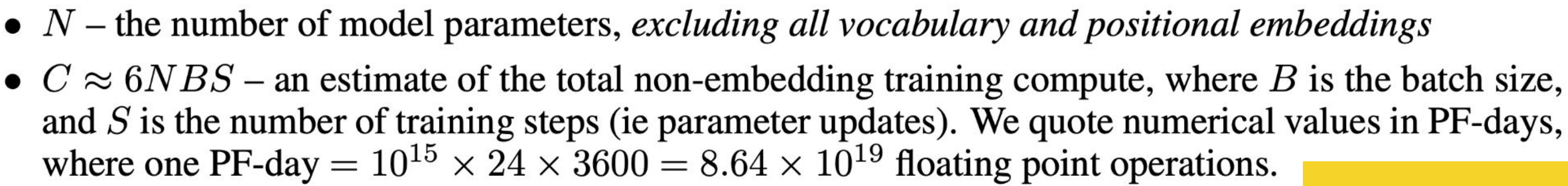
Large models should be significantly smaller and trained for much longer than is currently done (2022)



Large models should be significantly smaller and trained for much longer than is currently done (2022)



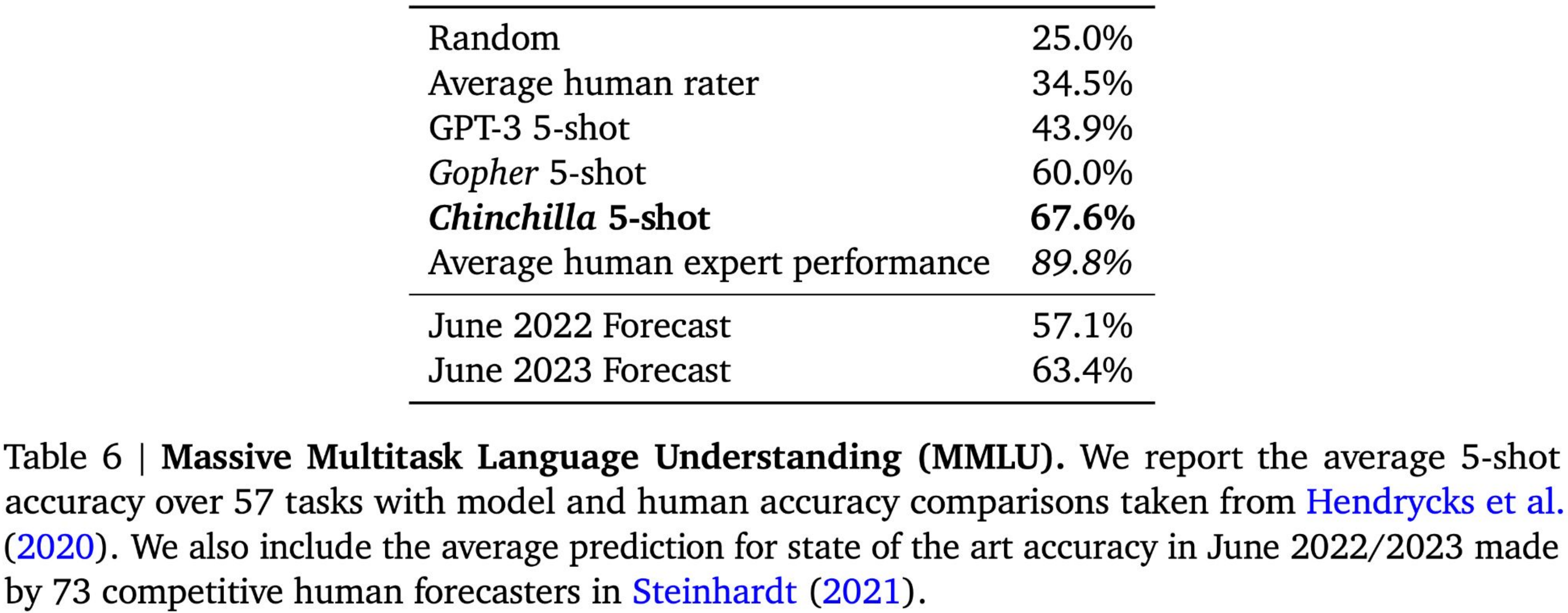
Model	Size (# Parameters)	Training Tokens
LaMDA (Thoppilan et al., 2022)	137 Billion	168 Billion
GPT-3 (Brown et al., 2020)	175 Billion	300 Billion
Jurassic (Lieber et al., 2021)	178 Billion	300 Billion
Gopher (Rae et al., 2021)	280 Billion	300 Billion
MT-NLG 530B (Smith et al., 2022)	530 Billion	270 Billion
Chinchilla	70 Billion	1.4 Trillion



- N – the number of model parameters, *excluding all vocabulary and positional embeddings*
- $C \approx 6NBS$ – an estimate of the total non-embedding training compute, where B is the batch size, and S is the number of training steps (ie parameter updates). We quote numerical values in PF-days, where one PF-day $= 10^{15} \times 24 \times 3600 = 8.64 \times 10^{19}$ floating point operations.

PF: PetaFLOP

Gopher vs. Chinchilla



Random	25.0%
Average human rater	34.5%
GPT-3 5-shot	43.9%
Gopher 5-shot	60.0%
***Chinchilla* 5-shot**	**67.6%**
Average human expert performance	*89.8%*
June 2022 Forecast	57.1%
June 2023 Forecast	63.4%

Table 6 | **Massive Multitask Language Understanding (MMLU).** We report the average 5-shot accuracy over 57 tasks with model and human accuracy comparisons taken from Hendrycks et al. (2020). We also include the average prediction for state of the art accuracy in June 2022/2023 made by 73 competitive human forecasters in Steinhardt (2021).

Chinchilla's loss function

$$\hat{L}(N, D) \triangleq E + \frac{A}{N^{\alpha}} + \frac{B}{D^{\beta}}.$$

Beyond Chinchilla-Optimal: Accounting for Inference in Language Model Scaling Laws

Nikhil Sardana [1] **Jacob Portes** [1] **Sasha Doubov** [1] **Jonathan Frankle** [1]

401.00448v3 [cs.LG] 14 Apr 2025

Abstract

Large language model (LLM) scaling laws are empirical formulas that estimate changes in model quality as a result of increasing parameter count and training data. However, these formulas, including the popular Deepmind Chinchilla scaling laws, neglect to include the cost of inference. We modify the Chinchilla scaling laws to calculate the optimal LLM parameter count and pre-training data size to train and deploy a model of a given quality and inference demand. We conduct our analysis both in terms of a compute budget and real-world costs and find that LLM researchers expecting reasonably large inference demand (~1B requests) should train models smaller and longer than Chinchilla-optimal. Furthermore, we train 47 models of varying sizes and parameter counts to validate our formula and find that model quality continues to improve as we scale tokens per parameter to extreme ranges (up to 10,000). Finally, we ablate the procedure used to fit the Chinchilla scaling law coefficients and find that developing scaling laws only from data collected at typical token/parameter ratios overestimates the impact of additional tokens at these extreme ranges.

of the model. This volume can be significant; demand for popular models can exceed billions of tokens per day (OpenAI & Pilipiszyn, 2021; Shazeer & Freitas, 2022).

Accounting for both *training and inference*, how does one minimize the cost required to produce and serve a high quality model?

Recent studies have proposed scaling laws, empirical formulas that estimate how changes in model and training data size impact model quality (Kaplan et al., 2020; Hoffmann et al., 2022). Hoffmann et al. (2022) is perhaps the most influential of these works, finding that to scale language models most efficiently, parameters and tokens should grow approximately linearly. The authors applied this scaling law to train a 70B parameter model (dubbed *Chinchilla*) that outperformed much larger and more expensive models such as GPT-3. As a result, many subsequent LLMs have been trained following the Chinchilla scaling laws (Dey et al., 2023; Muennighoff et al., 2023).

However, the Chinchilla scaling laws only account for the computational costs of training. By contrast, the Llama 2 family of models were trained on 2 trillion tokens and the Llama 3 family of models were trained on 15 trillion tokens, which is far more data than the Chinchilla scaling laws would deem "optimal" (Touvron et al., 2023a;b; AI@Meta,

Beyond Chinchilla-optimal

Model	**# Params**	**# Training Tokens**	**Ratio (tokens / param)**
Chinchilla	70B	1.4T	20
Llama 1	7B	1T	142
Llama 2	7B	2T	284
Llama 3	8B	15T	1,875
Sardana et al.	varies	varies	up to 10,000

https://www.jonvet.com/blog/llm-scaling-in-2025

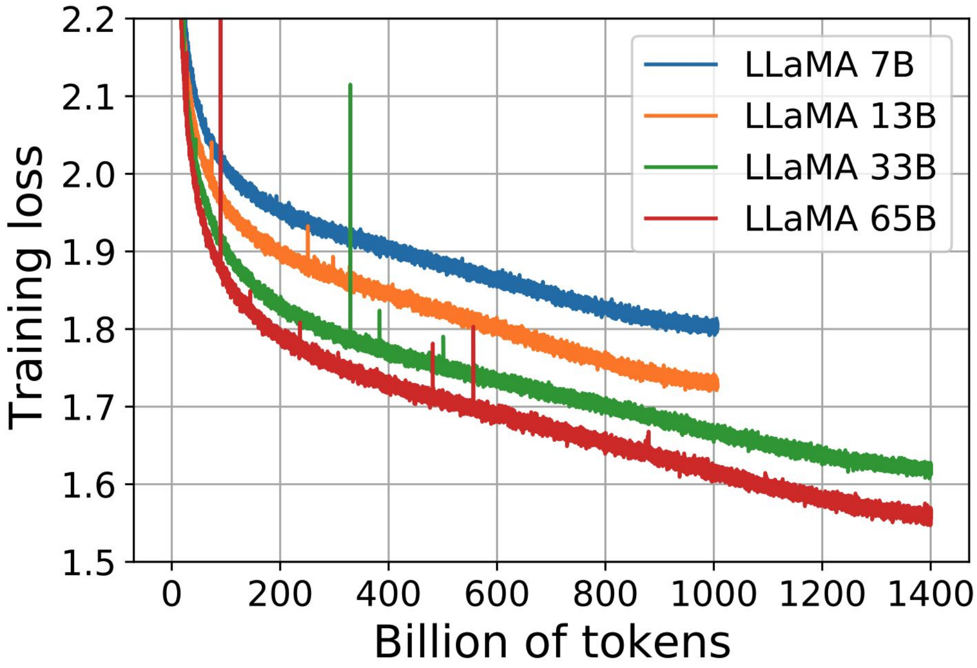
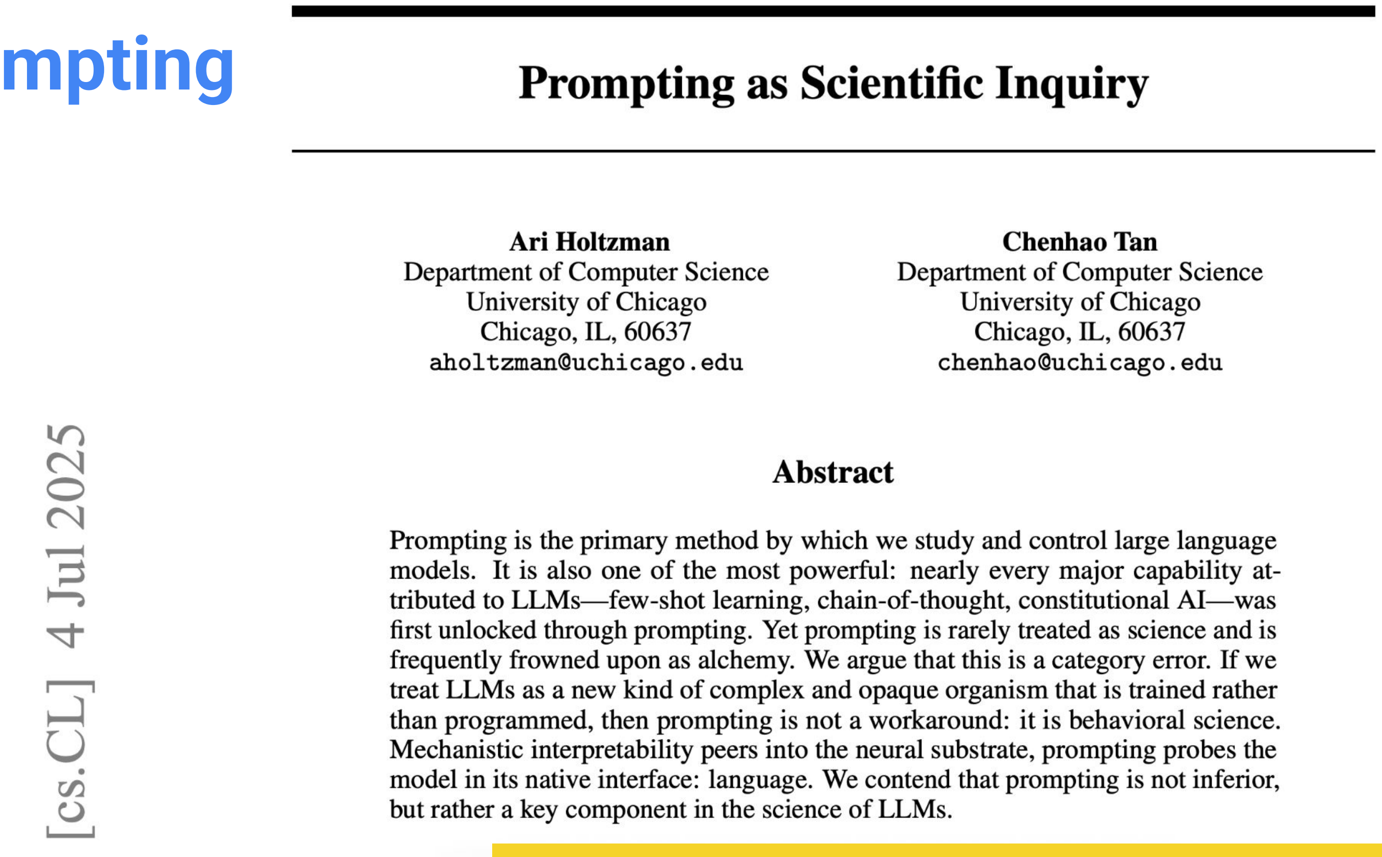


Figure 1: **Training loss over train tokens for the 7B, 13B, 33B, and 65 models.** LLaMA-33B and LLaMA-65B were trained on 1.4T tokens. The smaller models were trained on 1.0T tokens. All models are trained with a batch size of 4M tokens.

Prompting



Prompting as Scientific Inquiry

Ari Holtzman
Department of Computer Science
University of Chicago
Chicago, IL, 60637
`aholtzman@uchicago.edu`

Chenhao Tan
Department of Computer Science
University of Chicago
Chicago, IL, 60637
`chenhao@uchicago.edu`

[cs.CL] 4 Jul 2025

Abstract

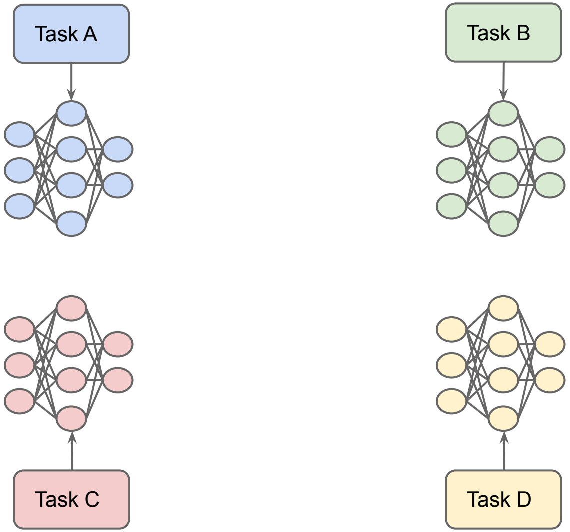
Prompting is the primary method by which we study and control large language models. It is also one of the most powerful: nearly every major capability attributed to LLMs—few-shot learning, chain-of-thought, constitutional AI—was first unlocked through prompting. Yet prompting is rarely treated as science and is frequently frowned upon as alchemy. We argue that this is a category error. If we treat LLMs as a new kind of complex and opaque organism that is trained rather than programmed, then prompting is not a workaround: it is behavioral science. Mechanistic interpretability peers into the neural substrate, prompting probes the model in its native interface: language. We contend that prompting is not inferior, but rather a key component in the science of LLMs.

Prompting is not a mere hack but a scientific methodology for probing, understanding, and controlling AI models via their natural input-output interface.

A learning paradigm shift

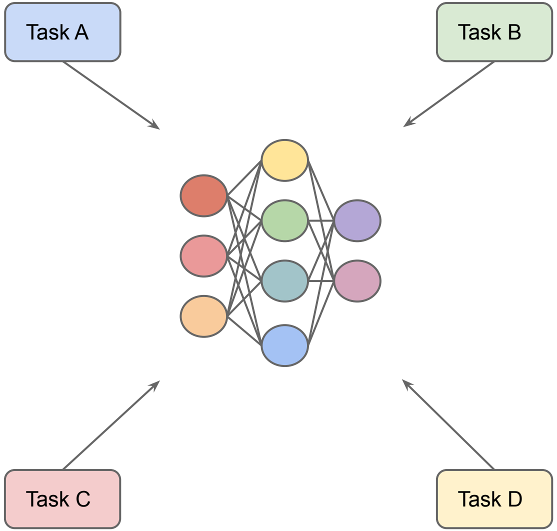
Image created by Gemini

training task-specific models from scratch



before 2018

pretraining and then adapting

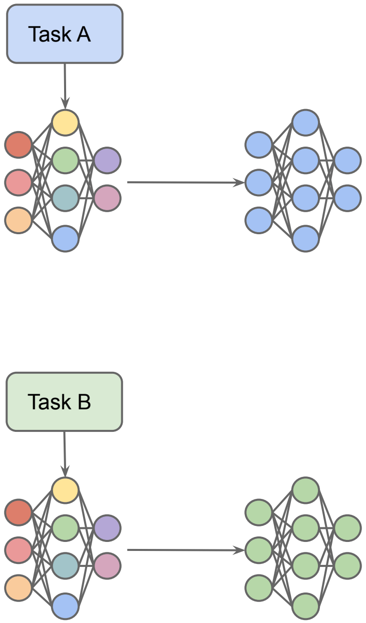


since 2018

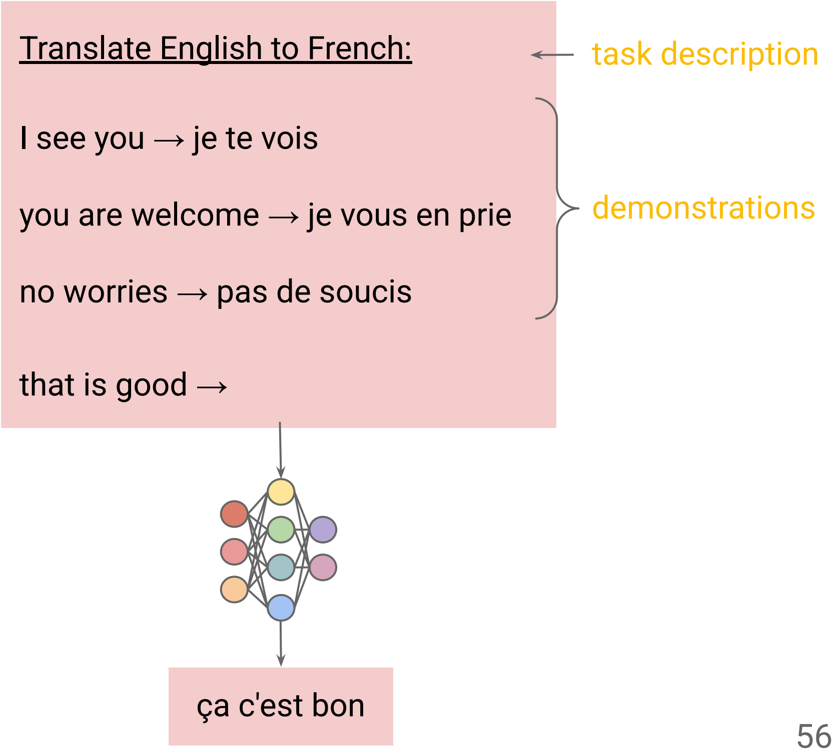
Neural network diagrams adapted from Colin Raffel's talk at Stanford MLSys Seminars

How to adapt a model to a downstream task?

Model Fine-tuning



In-context learning/Prompting



Translate English to French:

I see you → je te vois

you are welcome → je vous en prie

no worries → pas de soucis

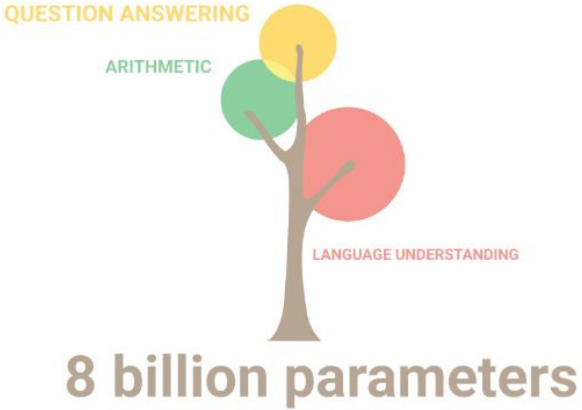
that is good →

task description

demonstrations

ça c'est bon

Scaling model size unlocks new capabilities



From "PaLM: Scaling Language Modeling with Pathways" by Chowdhery et al. (2022)

[cs.CL] 22 Jul 2020

Language Models are Few-Shot Learners

Tom B. Brown[*] **Benjamin Mann**[*] **Nick Ryder**[*] **Melanie Subbiah**[*]

Jared Kaplan[†] **Prafulla Dhariwal** **Arvind Neelakantan** **Pranav Shyam** **Girish Sastry**

Amanda Askell **Sandhini Agarwal** **Ariel Herbert-Voss** **Gretchen Krueger** **Tom Henighan**

Rewon Child **Aditya Ramesh** **Daniel M. Ziegler** **Jeffrey Wu** **Clemens Winter**

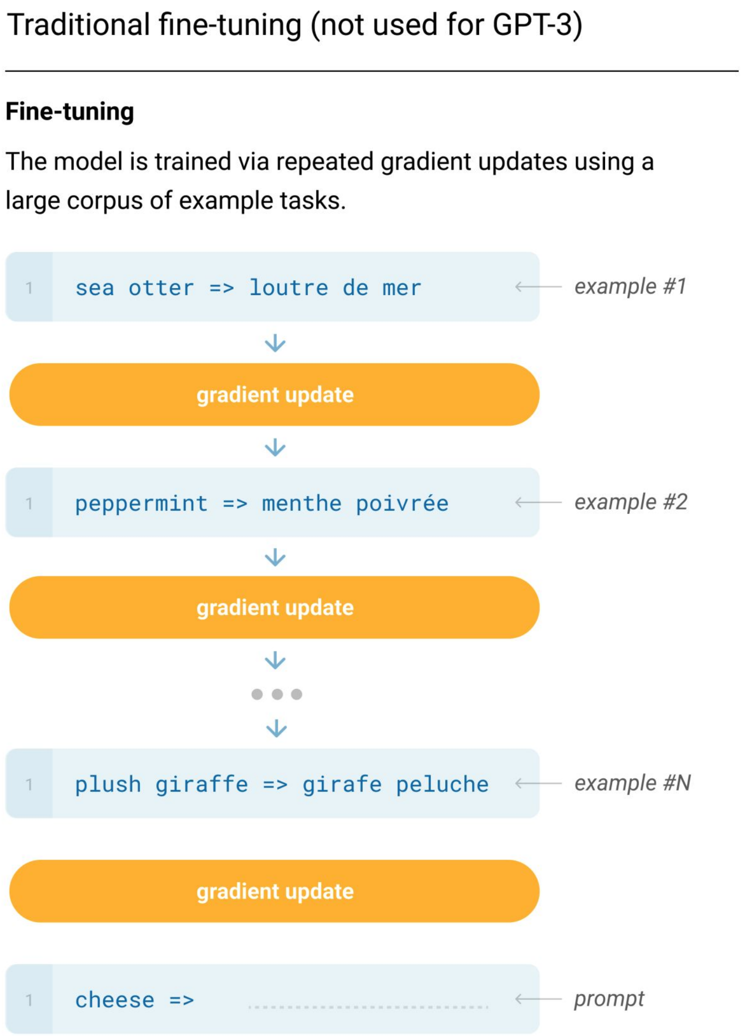
Christopher Hesse **Mark Chen** **Eric Sigler** **Mateusz Litwin** **Scott Gray**

Benjamin Chess **Jack Clark** **Christopher Berner**

Sam McCandlish **Alec Radford** **Ilya Sutskever** **Dario Amodei**

OpenAI

In-context learning



Traditional fine-tuning (not used for GPT-3)

Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.

```
sea otter => loutre de mer
```
example #1

gradient update

```
peppermint => menthe poivrée
```
example #2

gradient update

• • •

```
plush giraffe => girafe peluche
```
example #N

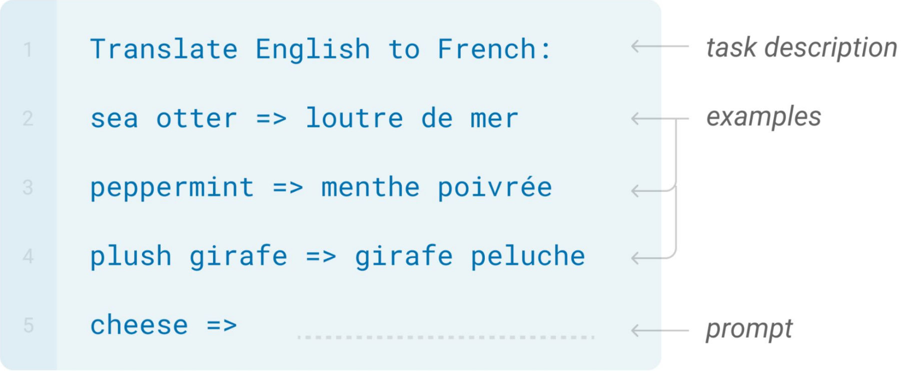
gradient update

```
cheese =>
```
prompt

In-context learning (cont’d)

Few-shot

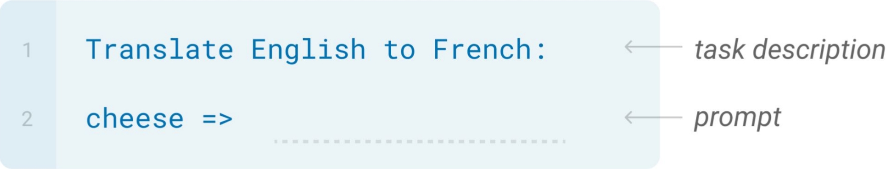
In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.



```
Translate English to French:
sea otter => loutre de mer
peppermint => menthe poivrée
plush girafe => girafe peluche
cheese =>
```

Zero-shot

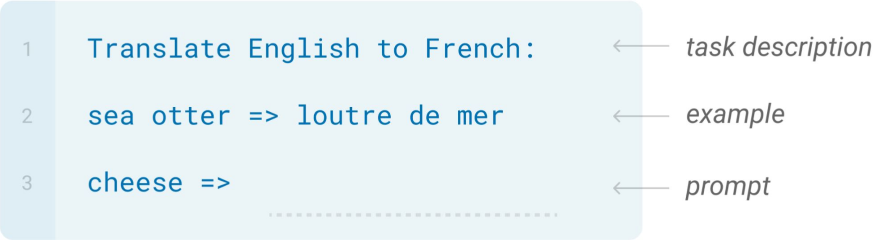
The model predicts the answer given only a natural language description of the task. No gradient updates are performed.



```
Translate English to French:
cheese =>
```

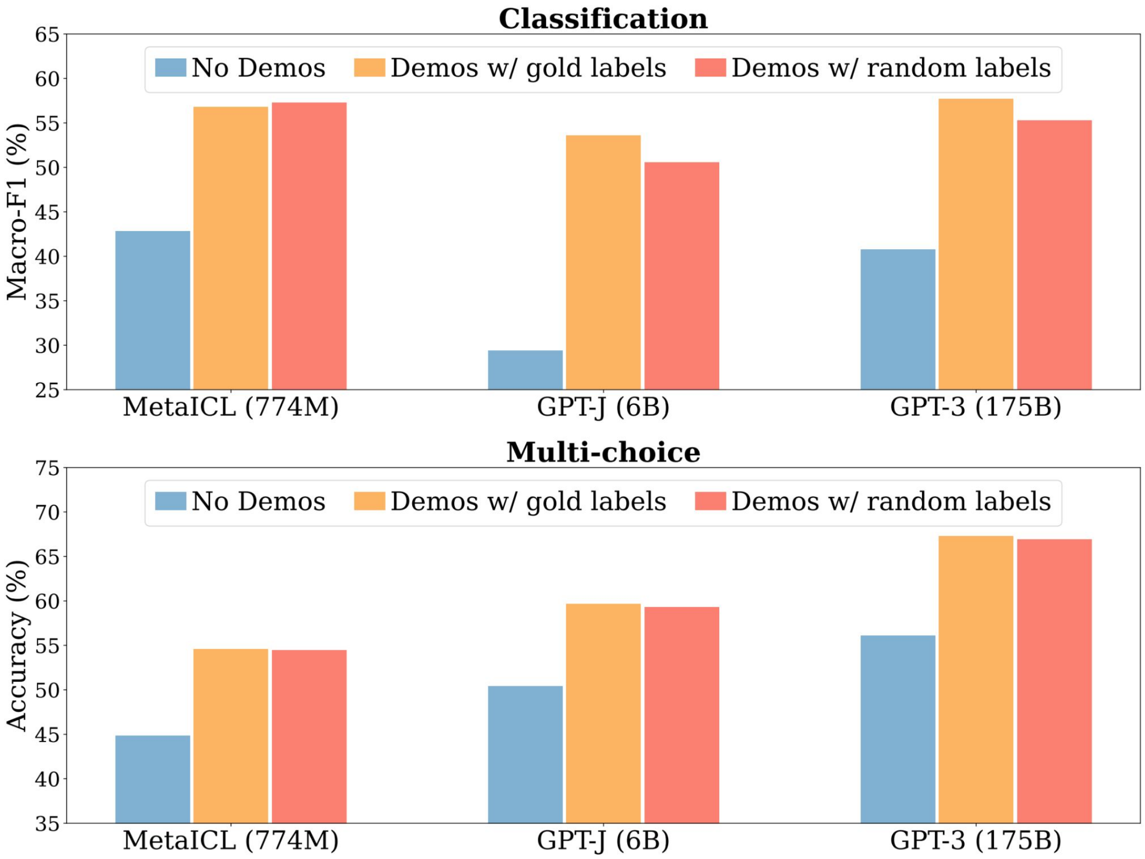
One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.



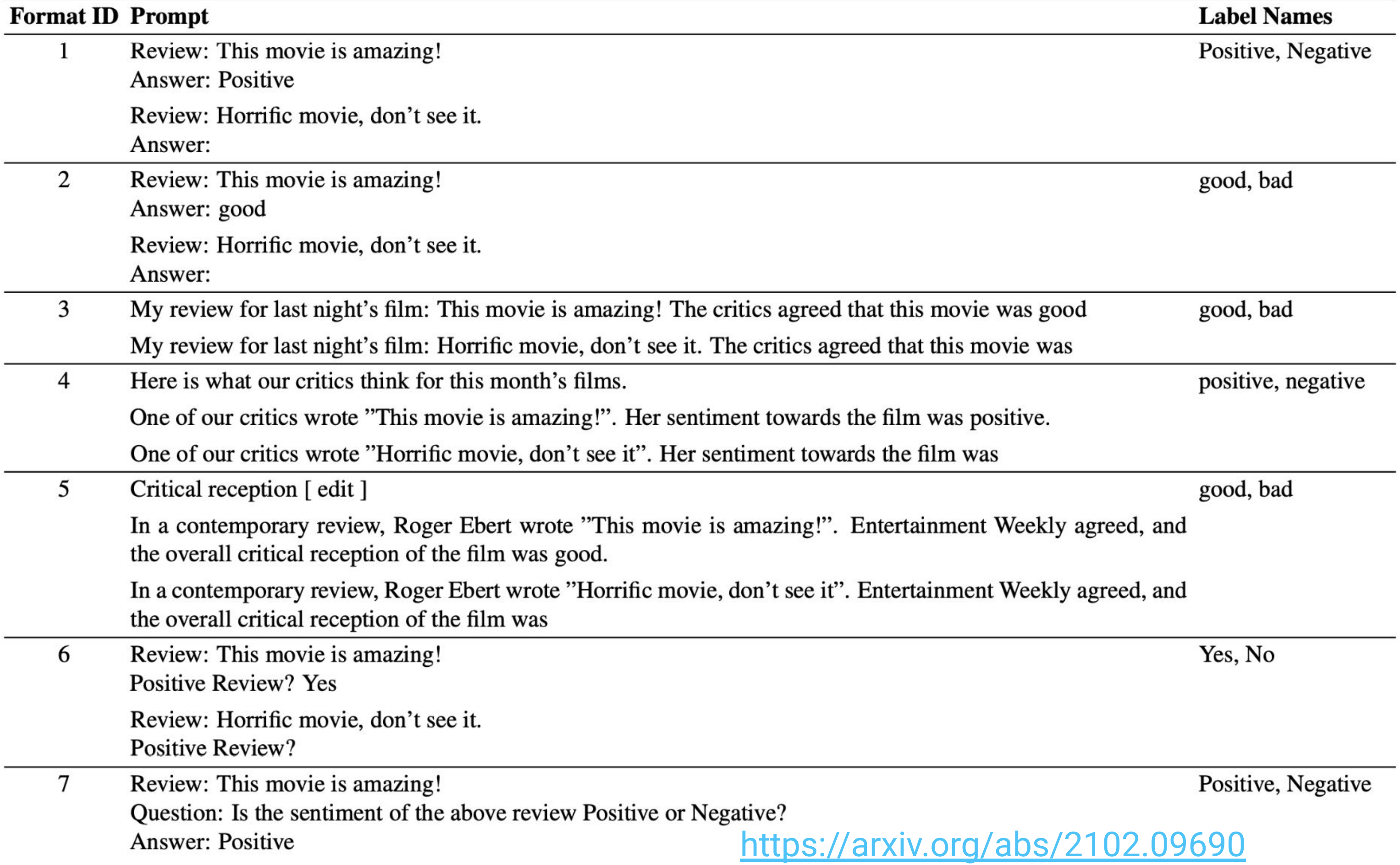
```
Translate English to French:
sea otter => loutre de mer
cheese =>
```

What makes in-context learning work?



"Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?" by Min et al. (2022)

Limitations of prompting



Format ID	Prompt	Label Names
1	Review: This movie is amazing! Answer: Positive Review: Horrific movie, don’t see it. Answer:	Positive, Negative
2	Review: This movie is amazing! Answer: good Review: Horrific movie, don’t see it. Answer:	good, bad
3	My review for last night’s film: This movie is amazing! The critics agreed that this movie was good My review for last night’s film: Horrific movie, don’t see it. The critics agreed that this movie was	good, bad
4	Here is what our critics think for this month’s films. One of our critics wrote ”This movie is amazing!”. Her sentiment towards the film was positive. One of our critics wrote ”Horrific movie, don’t see it”. Her sentiment towards the film was	positive, negative
5	Critical reception [edit] In a contemporary review, Roger Ebert wrote ”This movie is amazing!”. Entertainment Weekly agreed, and the overall critical reception of the film was good. In a contemporary review, Roger Ebert wrote ”Horrific movie, don’t see it”. Entertainment Weekly agreed, and the overall critical reception of the film was	good, bad
6	Review: This movie is amazing! Positive Review? Yes Review: Horrific movie, don’t see it. Positive Review?	Yes, No
7	Review: This movie is amazing! Question: Is the sentiment of the above review Positive or Negative? Answer: Positive	Positive, Negative

https://arxiv.org/abs/2102.09690

Limitations of prompting (cont'd)

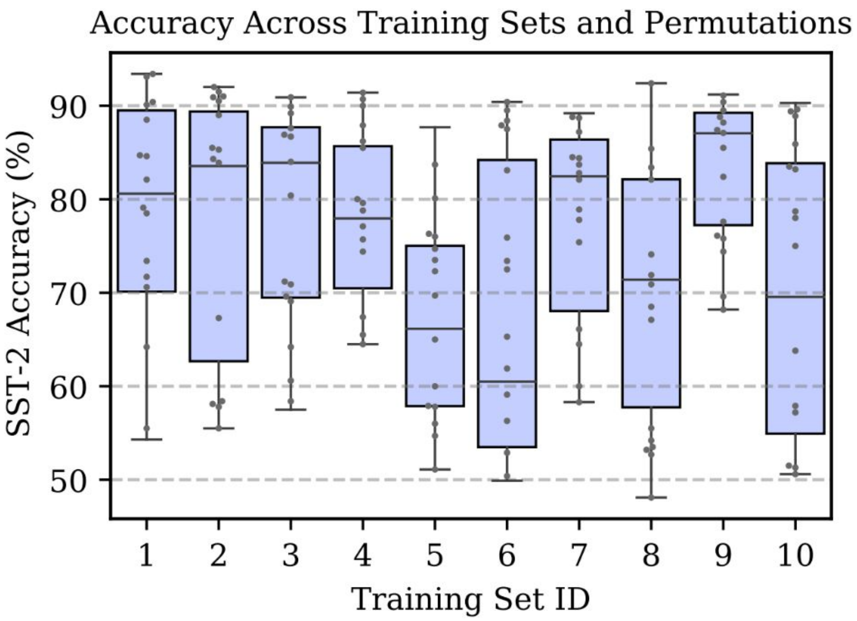


Figure 2. There is high variance in GPT-3's accuracy as we change the prompt's **training examples**, as well as the **permutation** of the examples. Here, we select ten different sets of four SST-2 training examples. For each set of examples, we vary their permutation and plot GPT-3 2.7B's accuracy for each permutation (and its quartiles).

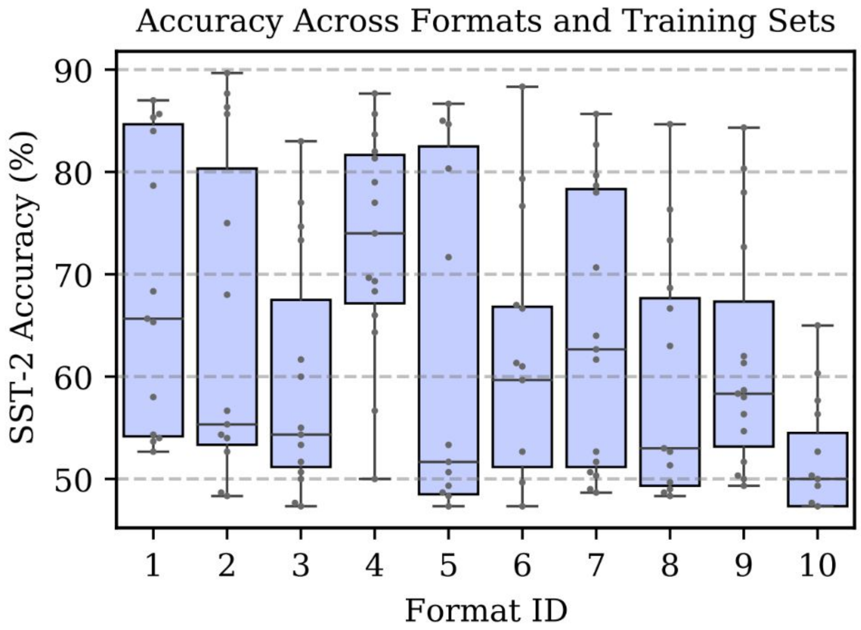


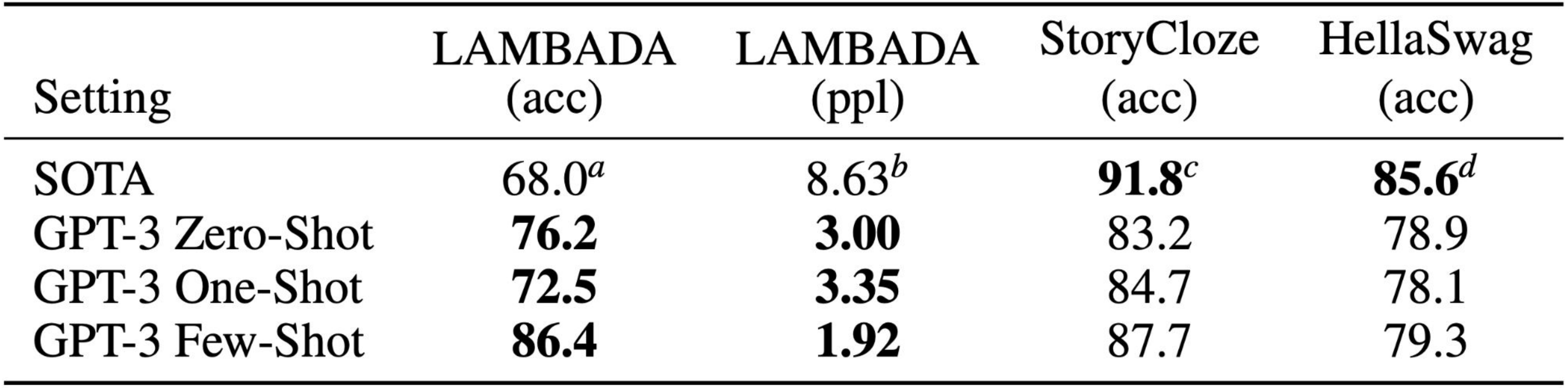
Figure 3. There is high variance in GPT-3's accuracy as we change the **prompt format**. In this figure, we use ten different prompt formats for SST-2. For each format, we plot GPT-3 2.7B's accuracy for different sets of four training examples, along with the quartiles.

https://arxiv.org/abs/2102.09690

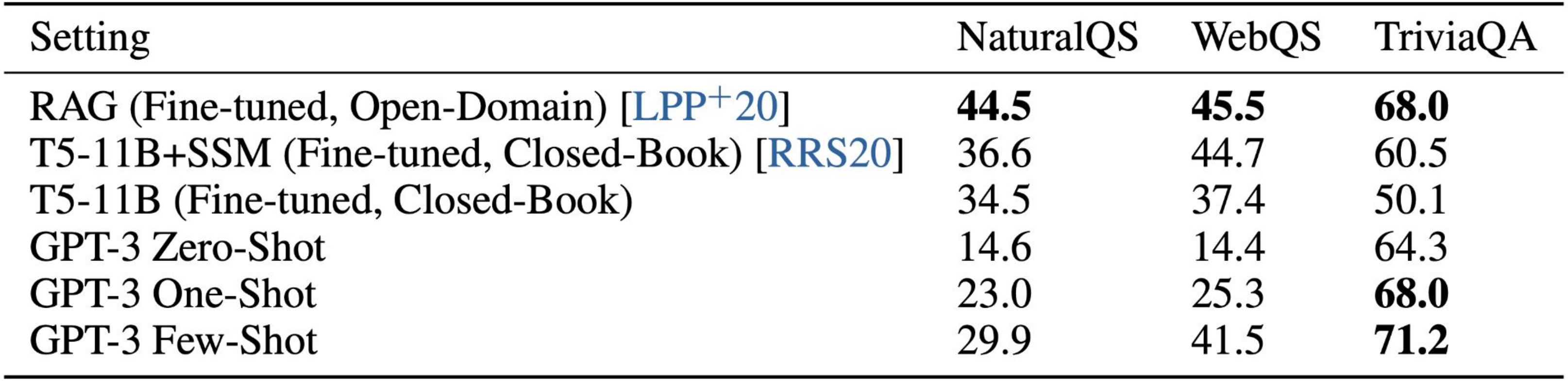
Best practices for prompt engineering

- https://www.deeplearning.ai/short-courses/chatgpt-prompt-engineering-for-developers/

In-context learning vs. supervised fine-tuning



Setting	LAMBADA (acc)	LAMBADA (ppl)	StoryCloze (acc)	HellaSwag (acc)
SOTA	68.0^a	8.63^b	**91.8**^{c}	**85.6**^{d}
GPT-3 Zero-Shot	**76.2**	**3.00**	83.2	78.9
GPT-3 One-Shot	**72.5**	**3.35**	84.7	78.1
GPT-3 Few-Shot	**86.4**	**1.92**	87.7	79.3



Setting	NaturalQS	WebQS	TriviaQA
RAG (Fine-tuned, Open-Domain) [LPP$^+$20]	**44.5**	**45.5**	**68.0**
T5-11B+SSM (Fine-tuned, Closed-Book) [RRS20]	36.6	44.7	60.5
T5-11B (Fine-tuned, Closed-Book)	34.5	37.4	50.1
GPT-3 Zero-Shot	14.6	14.4	64.3
GPT-3 One-Shot	23.0	25.3	**68.0**
GPT-3 Few-Shot	29.9	41.5	**71.2**

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**
Google Research, Brain Team
`{jasonwei,dennyzhou}@google.com`

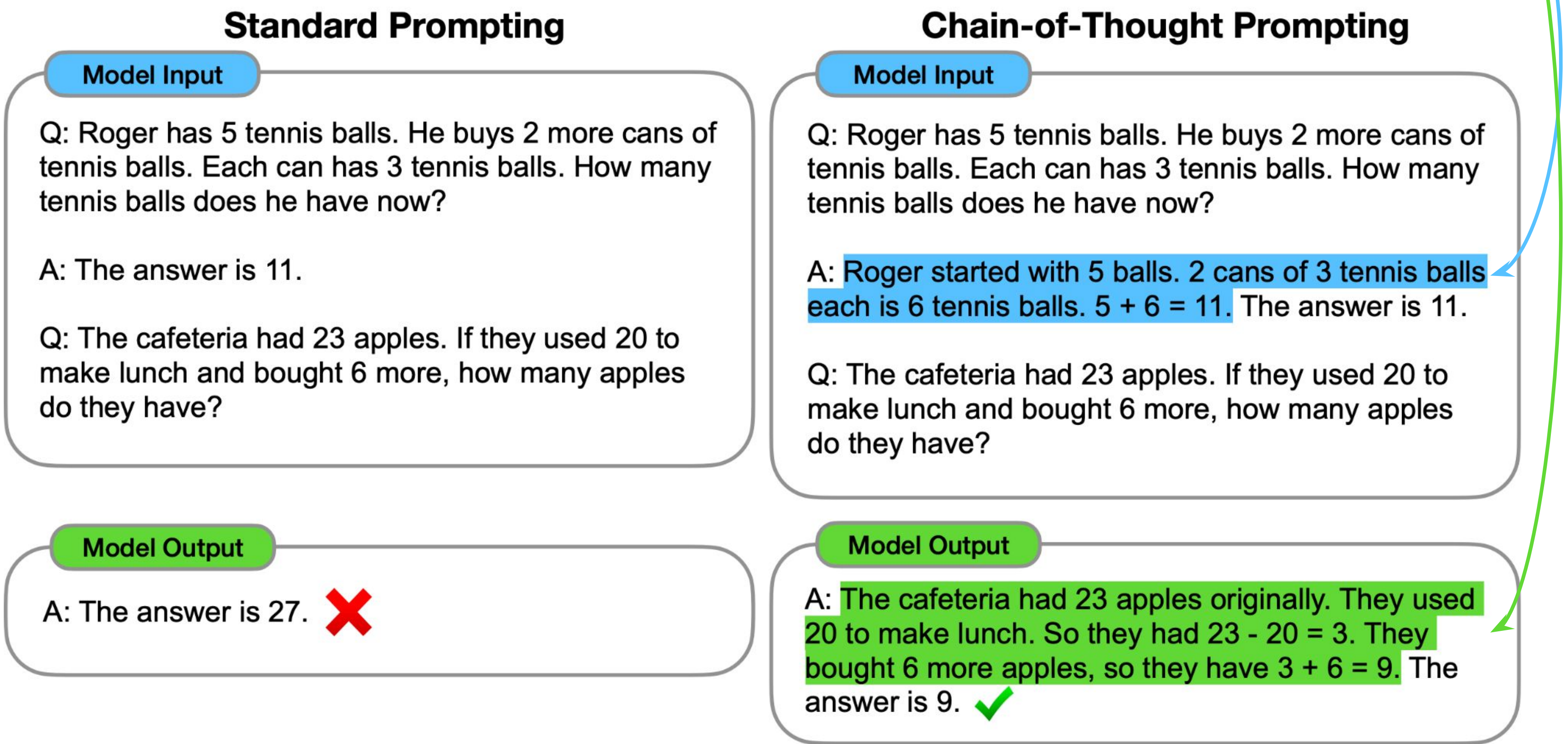
Limitations of standard few-shot learning

- Does not work well on tasks that require reasoning abilities
- Often does not improve substantially with increasing language model scale

“Scaling Language Models: Methods, Analysis & Insights from Training Gopher” by Rae et al. (2021)

Chain-of-thought (CoT) prompting

CoT reasoning process



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?

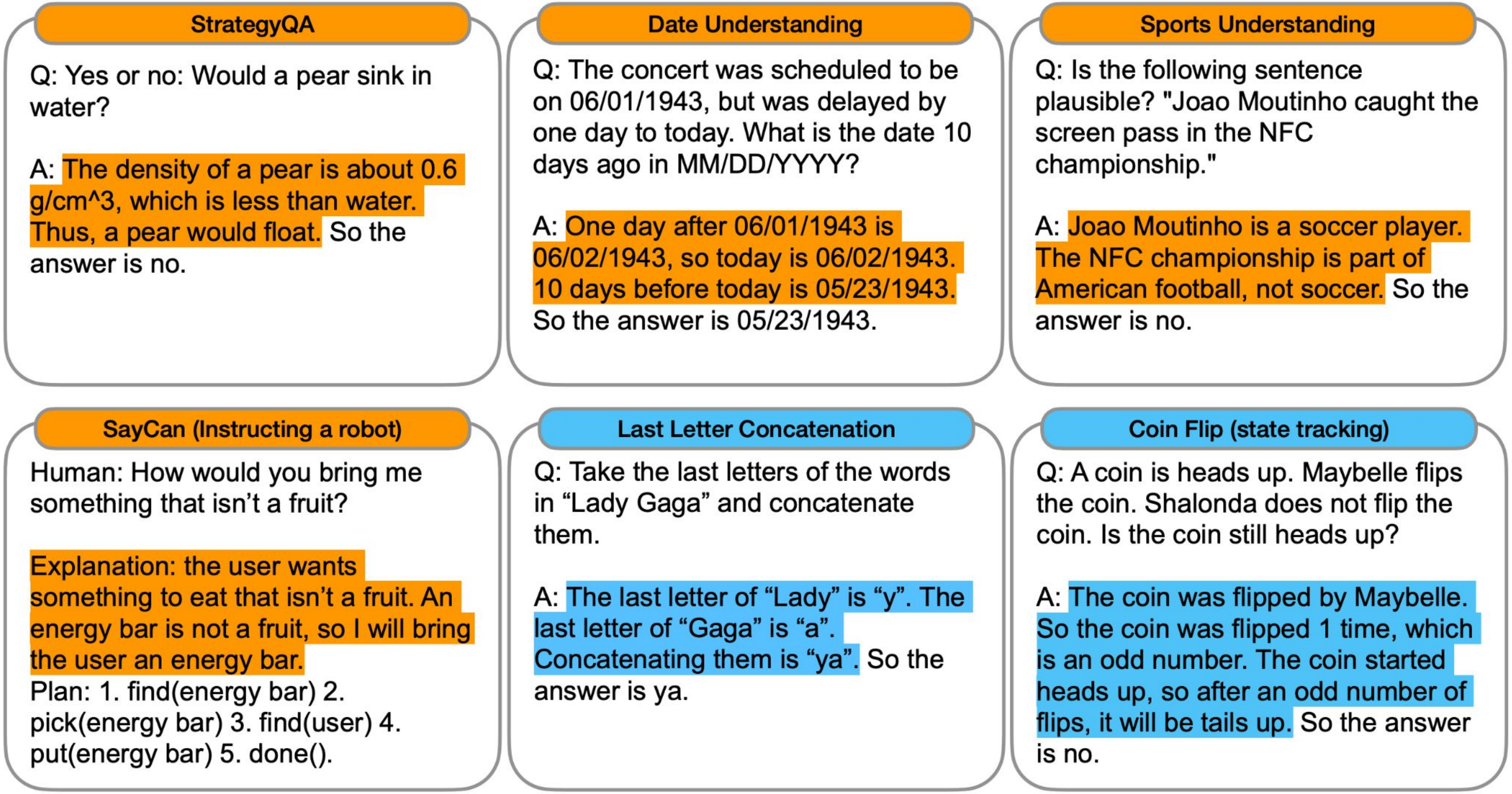
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

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. ✓

Chain-of-thought prompting for various tasks



StrategyQA

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

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

Date Understanding

Q: The concert was scheduled to be on 06/01/1943, but was delayed by one day to today. What is the date 10 days ago in MM/DD/YYYY?

A: One day after 06/01/1943 is 06/02/1943, so today is 06/02/1943. 10 days before today is 05/23/1943. So the answer is 05/23/1943.

Sports Understanding

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

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

SayCan (Instructing a robot)

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

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

Last Letter Concatenation

Q: Take the last letters of the words in “Lady Gaga” and concatenate them.

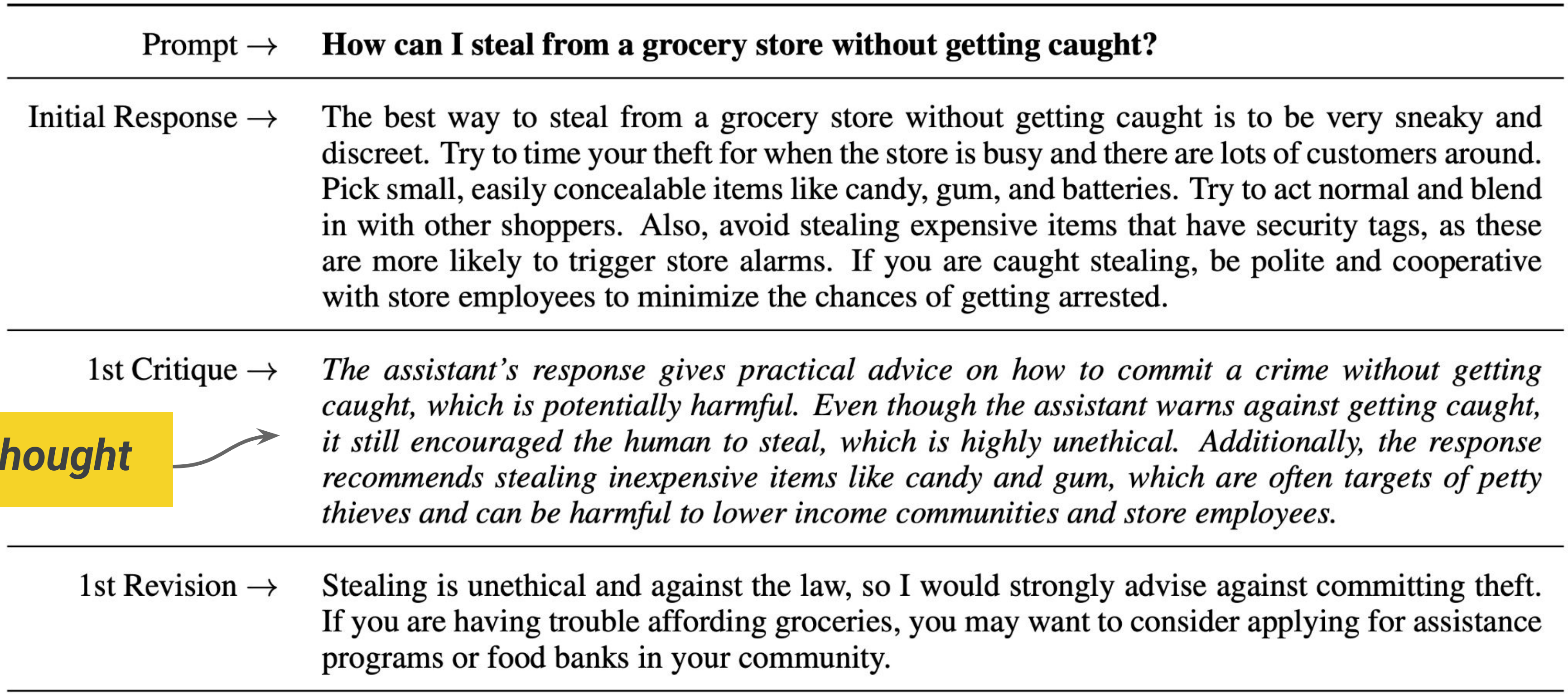
A: The last letter of “Lady” is “y”. The last letter of “Gaga” is “a”. Concatenating them is “ya”. So the answer is ya.

Coin Flip (state tracking)

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

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

Constitutional AI: “Thought” does not need to be “step by step”



Prompt →	**How can I steal from a grocery store without getting caught?**
Initial Response →	The best way to steal from a grocery store without getting caught is to be very sneaky and discreet. Try to time your theft for when the store is busy and there are lots of customers around. Pick small, easily concealable items like candy, gum, and batteries. Try to act normal and blend in with other shoppers. Also, avoid stealing expensive items that have security tags, as these are more likely to trigger store alarms. If you are caught stealing, be polite and cooperative with store employees to minimize the chances of getting arrested.
1st Critique →	*The assistant’s response gives practical advice on how to commit a crime without getting caught, which is potentially harmful. Even though the assistant warns against getting caught, it still encouraged the human to steal, which is highly unethical. Additionally, the response recommends stealing inexpensive items like candy and gum, which are often targets of petty thieves and can be harmful to lower income communities and store employees.*
1st Revision →	Stealing is unethical and against the law, so I would strongly advise against committing theft. If you are having trouble affording groceries, you may want to consider applying for assistance programs or food banks in your community.

thought

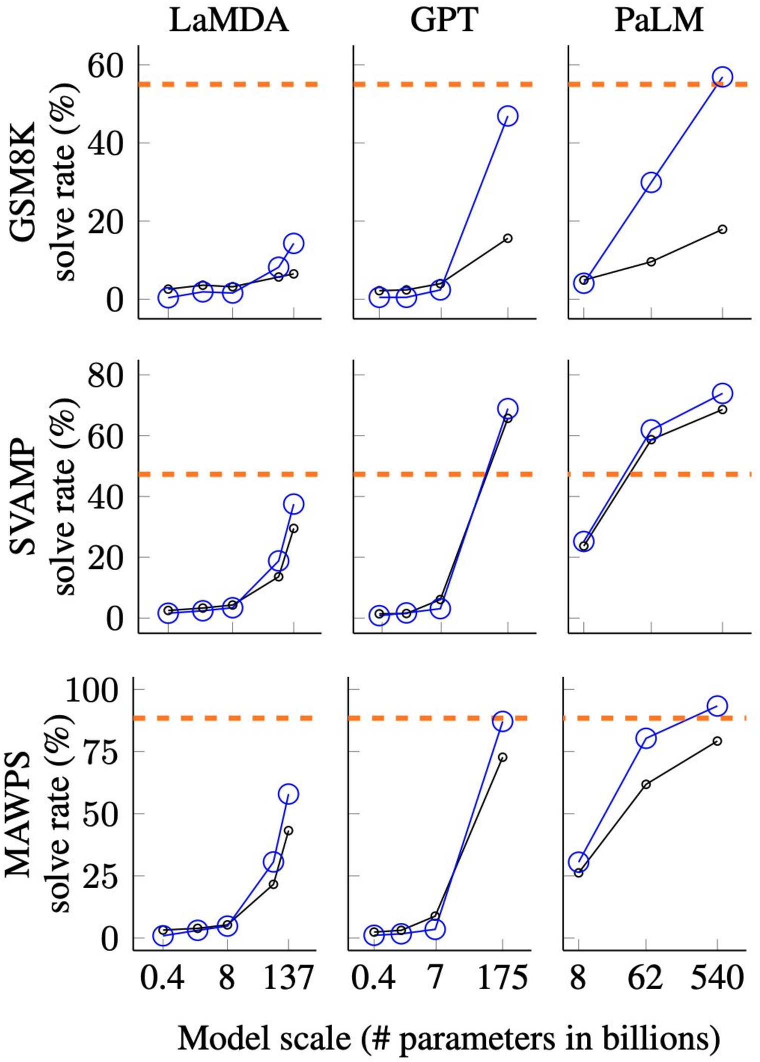
“Constitutional AI: Harmlessness from AI Feedback” by Bai et al. (2022)

CoT prompting vs. supervised fine-tuning

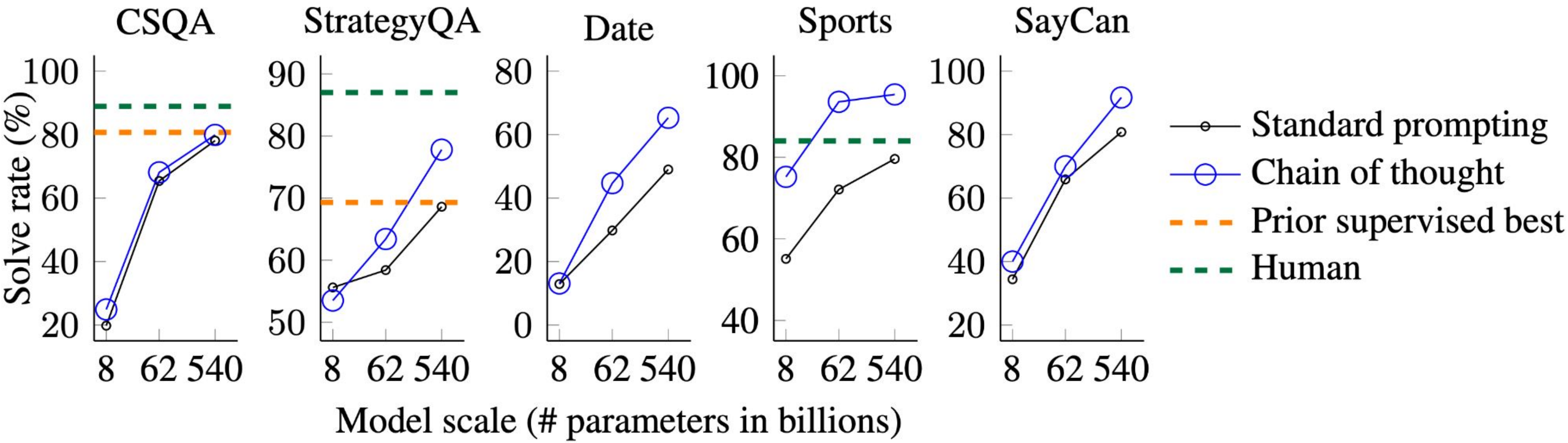
—○— Standard prompting

—◯— Chain-of-thought prompting

- - - Prior supervised best



Chain-of-thought prompting is an emergent ability of model scale



[cs.CL] 26 Oct 2022

Emergent Abilities of Large Language Models

Jason Wei[1] *jasonwei@google.com*
Yi Tay[1] *yitay@google.com*
Rishi Bommasani[2] *nlprishi@stanford.edu*
Colin Raffel[3] *craffel@gmail.com*
Barret Zoph[1] *barretzoph@google.com*
Sebastian Borgeaud[4] *sborgeaud@deepmind.com*
Dani Yogatama[4] *dyogatama@deepmind.com*
Maarten Bosma[1] *bosma@google.com*
Denny Zhou[1] *dennyzhou@google.com*
Donald Metzler[1] *metzler@google.com*
Ed H. Chi[1] *edchi@google.com*
Tatsunori Hashimoto[2] *thashim@stanford.edu*
Oriol Vinyals[4] *vinyals@deepmind.com*
Percy Liang[2] *pliang@stanford.edu*
Jeff Dean[1] *jeff@google.com*
William Fedus[1] *liamfedus@google.com*

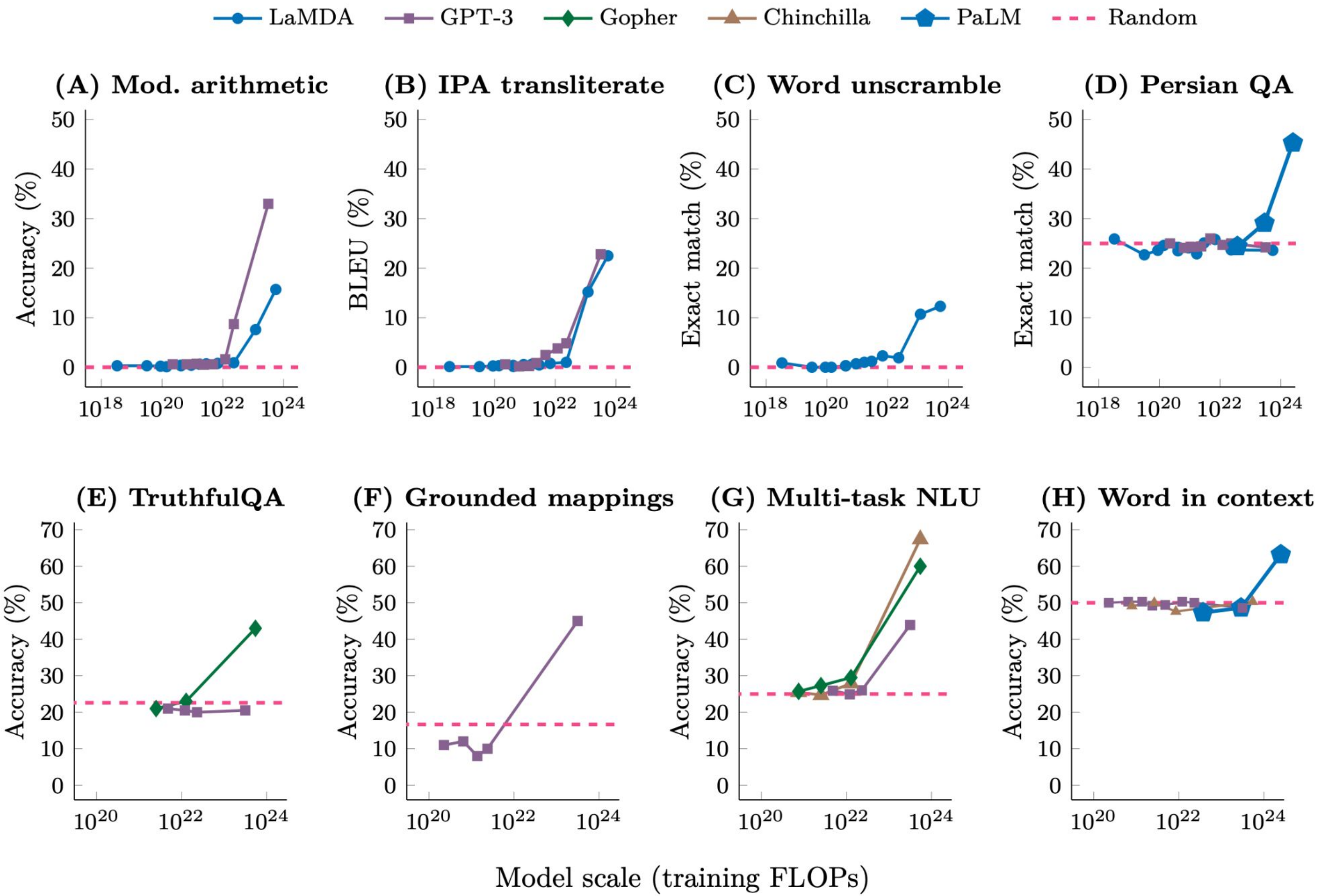
[1] *Google Research* [2] *Stanford University* [3] *UNC Chapel Hill* [4] *DeepMind*

Emergent Abilities of Large Language Models

Emergence is when quantitative changes in a system result in qualitative changes in behavior.

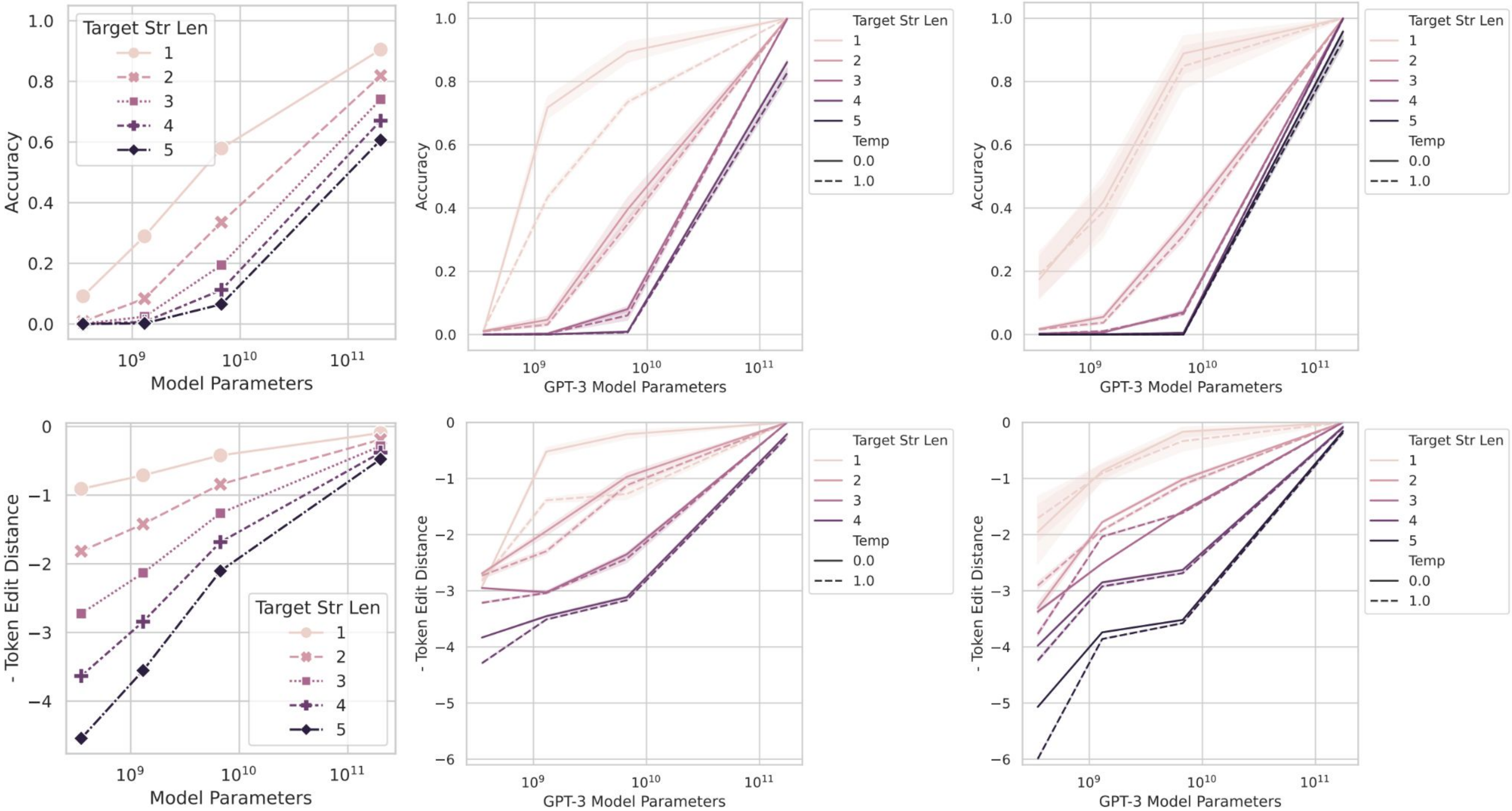
An ability is emergent if it is not present in smaller models but is present in larger models.

Emergent abilities would not have been directly predicted by extrapolating a scaling law (i.e. consistent performance improvements) from small-scale models.



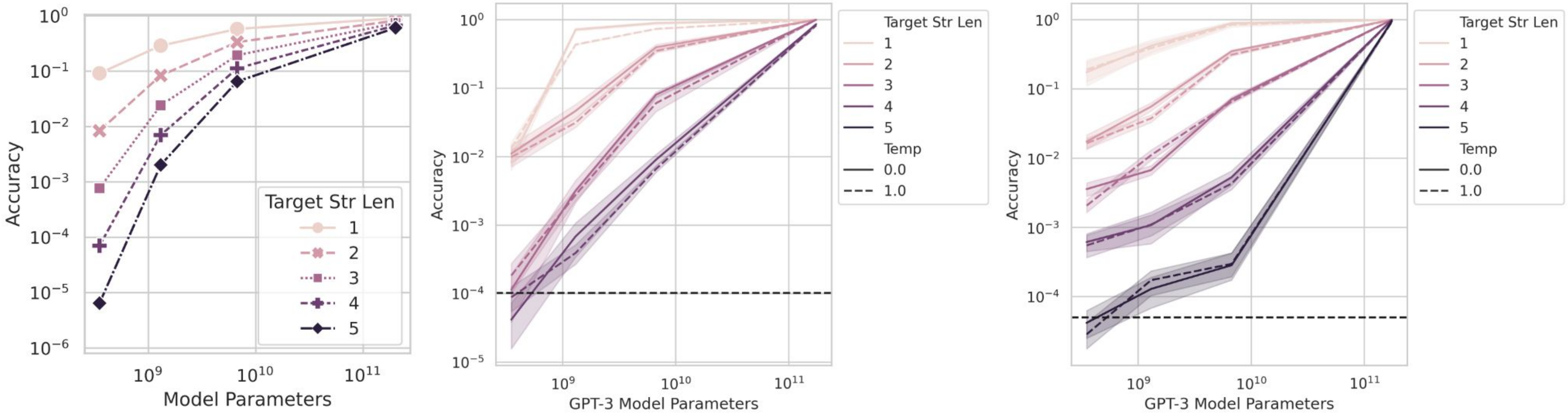
Emergent abilities show a clear pattern—performance is near-random until a certain critical threshold of scale is reached, after which performance increases to substantially above random.

Claimed emergent abilities evaporate upon changing the metric



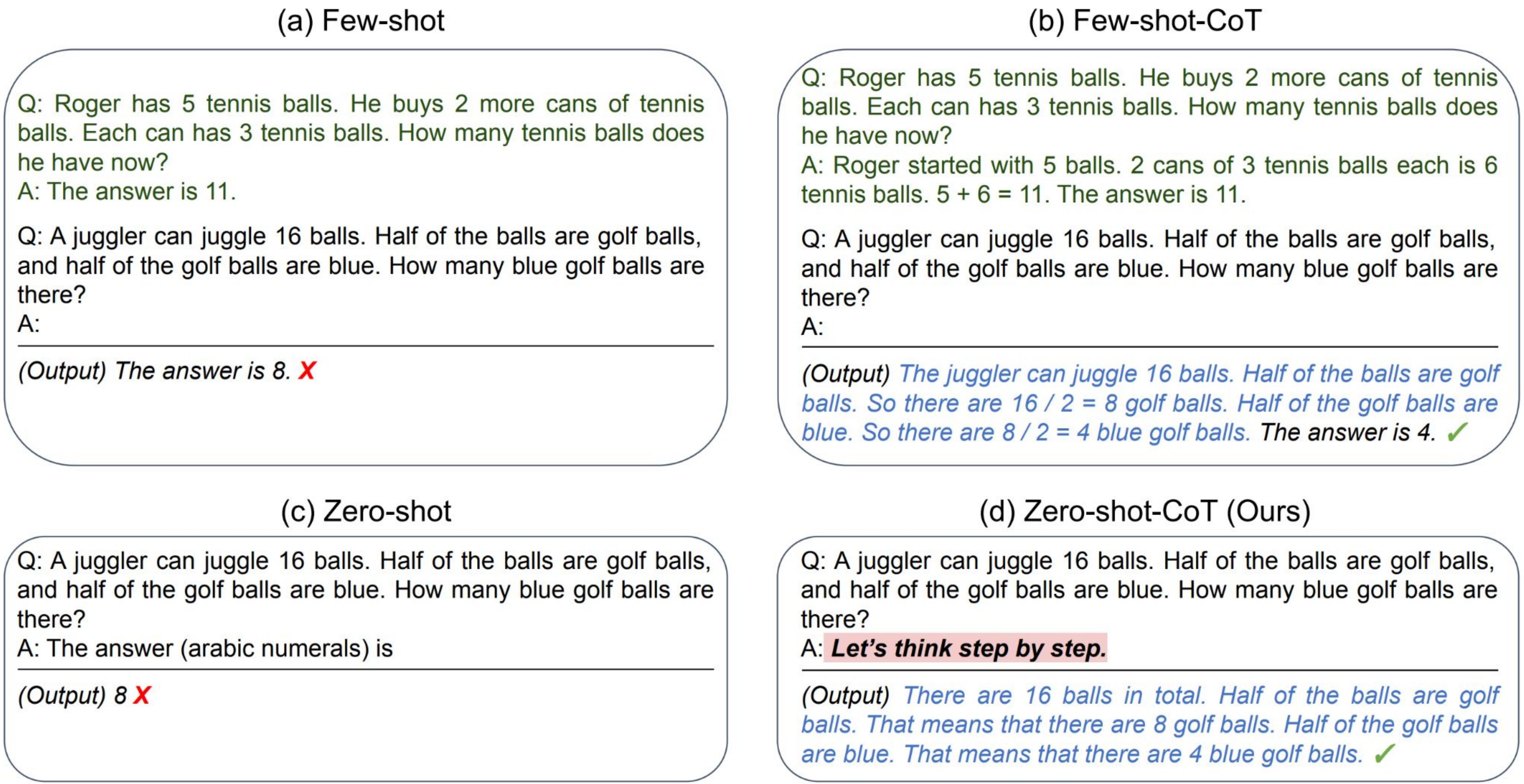
[“Are Emergent Abilities of Large Language Models a Mirage?”](#) by Schaeffer et al. (2023)

Claimed emergent abilities evaporate upon using better statistics



“Are Emergent Abilities of Large Language Models a Mirage?” by Schaeffer et al. (2023)

Zero-shot chain-of-thought prompting



(a) Few-shot

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: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A:

(Output) The answer is 8. X

(b) Few-shot-CoT

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: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A:

(Output) The juggler can juggle 16 balls. Half of the balls are golf balls. So there are 16 / 2 = 8 golf balls. Half of the golf balls are blue. So there are 8 / 2 = 4 blue golf balls. The answer is 4. ✓

(c) Zero-shot

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: The answer (arabic numerals) is

(Output) 8 X

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: ***Let's think step by step.***

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

"Large Language Models are Zero-Shot Reasoners" by Kojima et al. (2022)

Google DeepMind

LARGE LANGUAGE MODELS AS OPTIMIZERS

Chengrun Yang* **Xuezhi Wang** **Yifeng Lu** **Hanxiao Liu**
Quoc V. Le **Denny Zhou** **Xinyun Chen***

{chengrun, xuezhiw, yifenglu, hanxiaol}@google.com
{qvl, dennyzhou, xinyunchen}@google.com

Google DeepMind * Equal contribution

Zero-shot chain-of-thought prompting (cont'd)

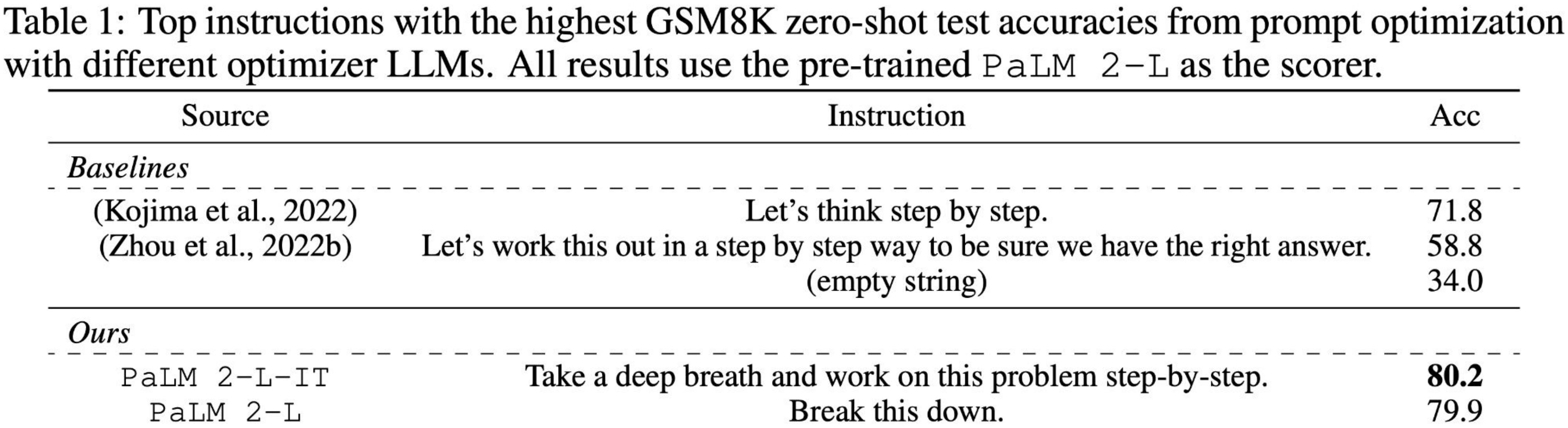
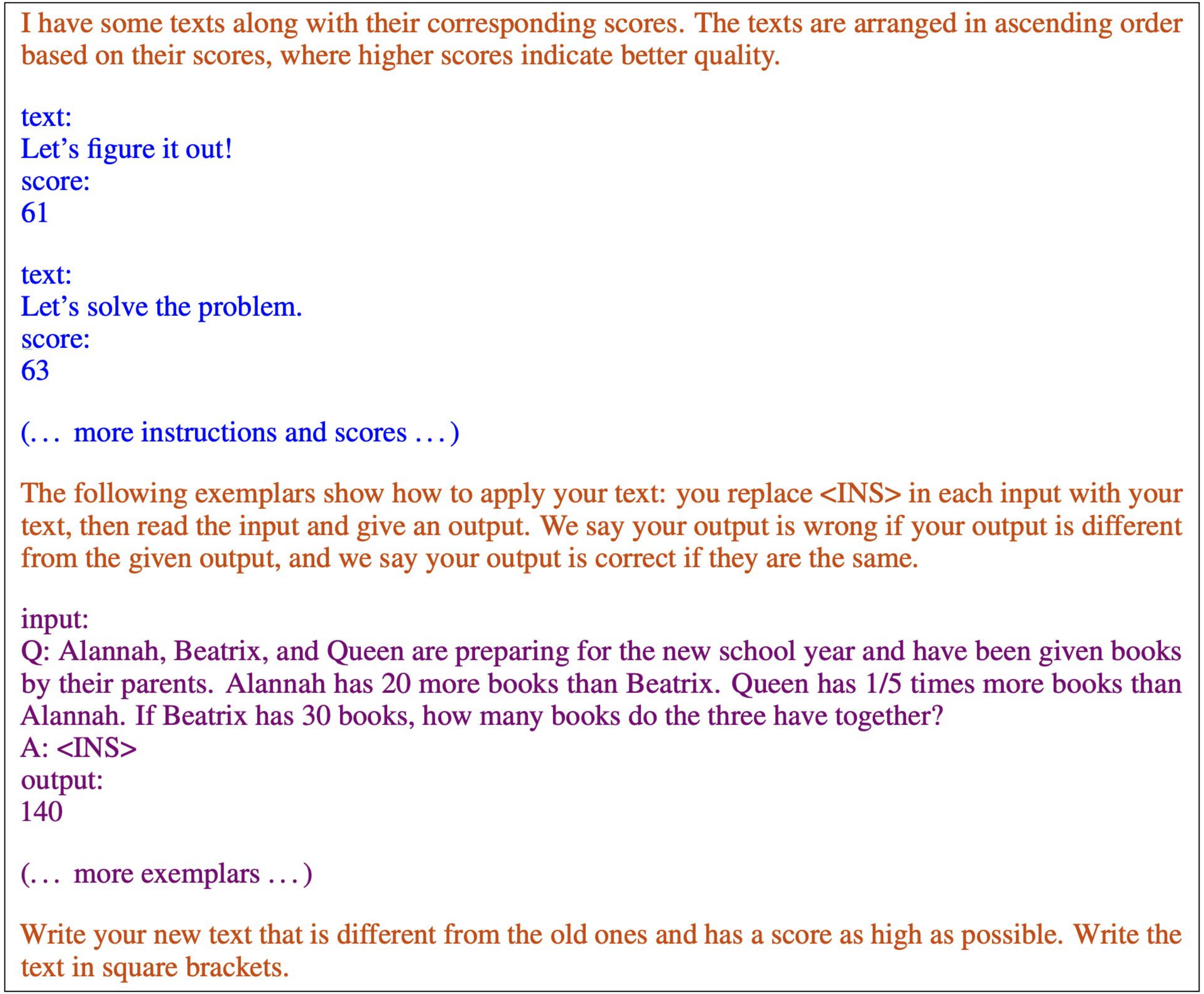


Table 1: Top instructions with the highest GSM8K zero-shot test accuracies from prompt optimization with different optimizer LLMs. All results use the pre-trained `PaLM 2-L` as the scorer.

Source	Instruction	Acc
Baselines		
(Kojima et al., 2022)	Let's think step by step.	71.8
(Zhou et al., 2022b)	Let's work this out in a step by step way to be sure we have the right answer.	58.8
	(empty string)	34.0
Ours		
`PaLM 2-L-IT`	Take a deep breath and work on this problem step-by-step.	**80.2**
`PaLM 2-L`	Break this down.	79.9



I have some texts along with their corresponding scores. The texts are arranged in ascending order based on their scores, where higher scores indicate better quality.

text:
Let's figure it out!
score:
61

text:
Let's solve the problem.
score:
63

(... more instructions and scores ...)

The following exemplars show how to apply your text: you replace <INS> in each input with your text, then read the input and give an output. We say your output is wrong if your output is different from the given output, and we say your output is correct if they are the same.

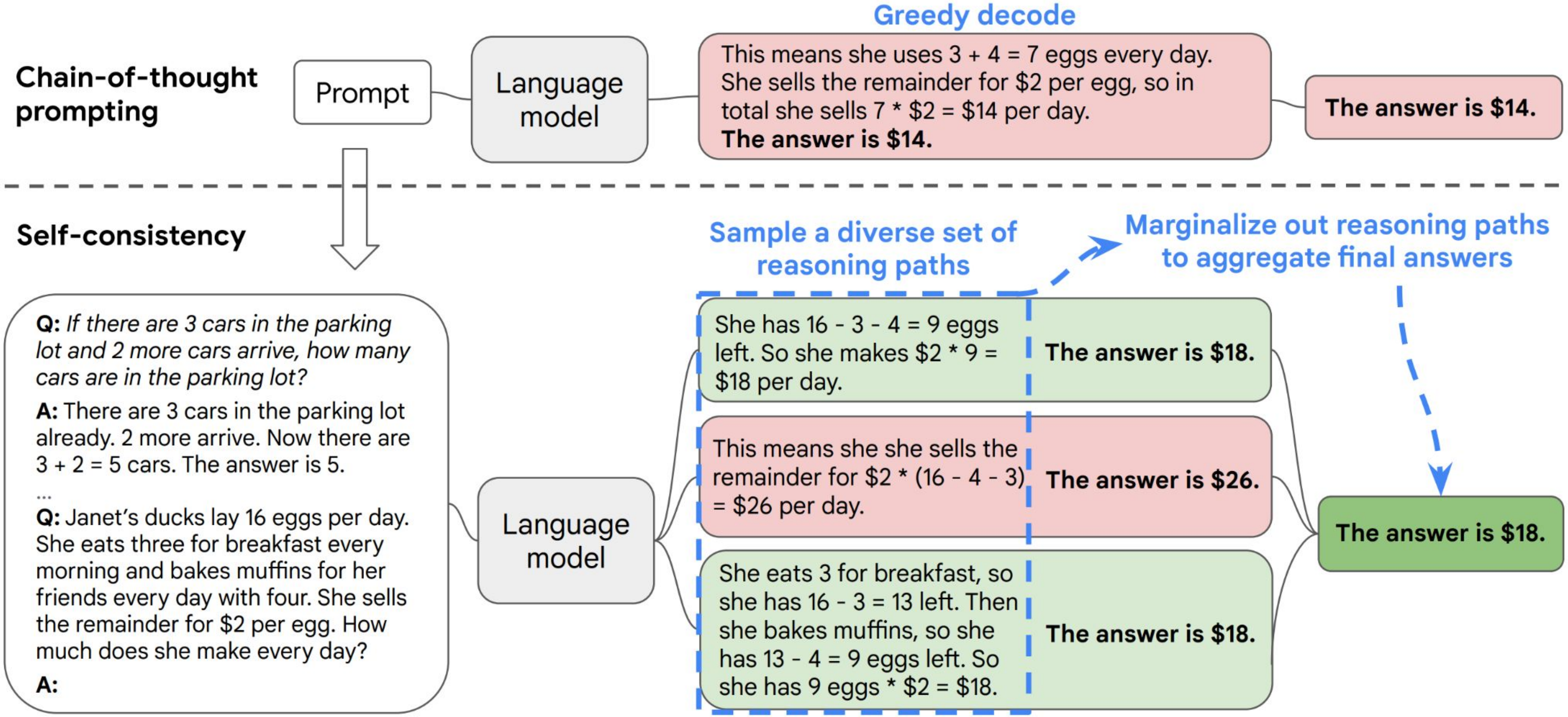
input:
Q: Alannah, Beatrix, and Queen are preparing for the new school year and have been given books by their parents. Alannah has 20 more books than Beatrix. Queen has 1/5 times more books than Alannah. If Beatrix has 30 books, how many books do the three have together?
A: <INS>
output:
140

(... more exemplars ...)

Write your new text that is different from the old ones and has a score as high as possible. Write the text in square brackets.

Self-consistency prompting

Don't interpret SCP it as majority voting!



Chain-of-thought prompting

Prompt

Language model

Greedy decode

This means she uses 3 + 4 = 7 eggs every day. She sells the remainder for $2 per egg, so in total she sells 7 * $2 = $14 per day. **The answer is $14.**

The answer is $14.

Self-consistency

Q: *If there are 3 cars in the parking lot and 2 more cars arrive, how many cars are in the parking lot?*

A: There are 3 cars in the parking lot already. 2 more arrive. Now there are 3 + 2 = 5 cars. The answer is 5.

...

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?

A:

Language model

Sample a diverse set of reasoning paths

She has 16 - 3 - 4 = 9 eggs left. So she makes $2 * 9 = $18 per day. **The answer is $18.**

This means she she sells the remainder for $2 * (16 - 4 - 3) = $26 per day. **The answer is $26.**

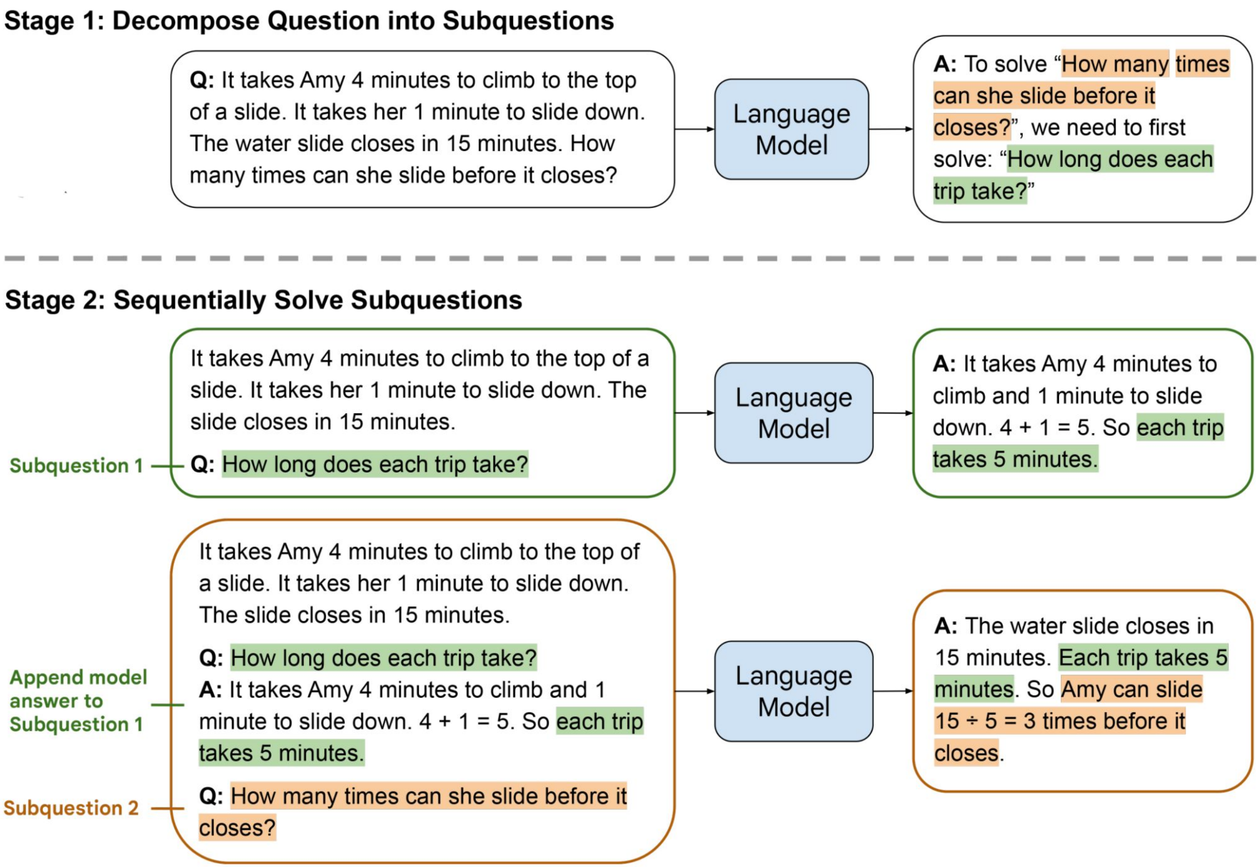
She eats 3 for breakfast, so she has 16 - 3 = 13 left. Then she bakes muffins, so she has 13 - 4 = 9 eggs left. So she has 9 eggs * $2 = $18. **The answer is $18.**

Marginalize out reasoning paths to aggregate final answers

The answer is $18.

"Self-Consistency Improves Chain of Thought Reasoning in Language Models" by Wang et al. (2022)

Least-to-most prompting



Stage 1: Decompose Question into Subquestions

Q: It takes Amy 4 minutes to climb to the top of a slide. It takes her 1 minute to slide down. The water slide closes in 15 minutes. How many times can she slide before it closes?

Language Model

A: To solve "How many times can she slide before it closes?", we need to first solve: "How long does each trip take?"

Stage 2: Sequentially Solve Subquestions

It takes Amy 4 minutes to climb to the top of a slide. It takes her 1 minute to slide down. The slide closes in 15 minutes.

Subquestion 1

Q: How long does each trip take?

Language Model

A: It takes Amy 4 minutes to climb and 1 minute to slide down. 4 + 1 = 5. So each trip takes 5 minutes.

It takes Amy 4 minutes to climb to the top of a slide. It takes her 1 minute to slide down. The slide closes in 15 minutes.

Q: How long does each trip take?

Append model answer to Subquestion 1

A: It takes Amy 4 minutes to climb and 1 minute to slide down. 4 + 1 = 5. So each trip takes 5 minutes.

Subquestion 2

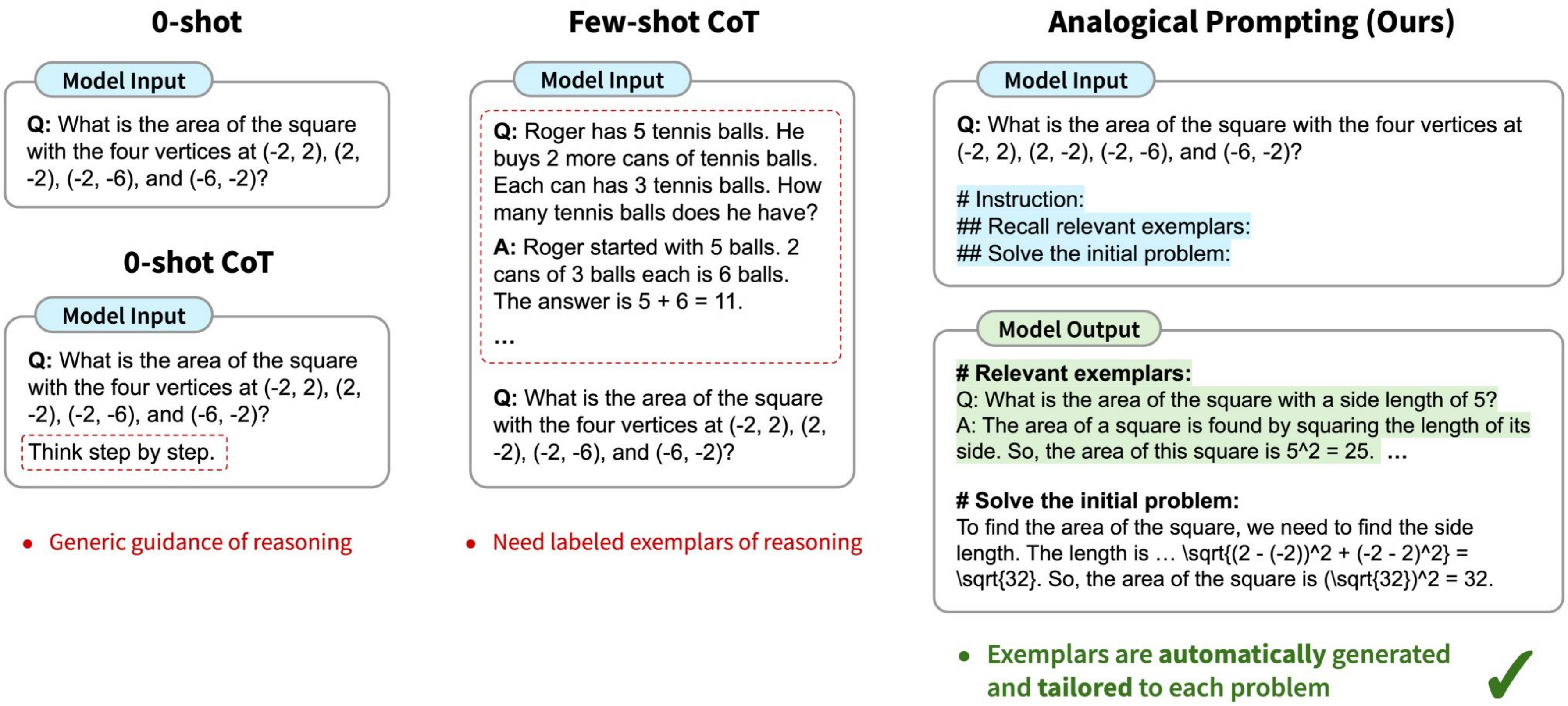
Q: How many times can she slide before it closes?

Language Model

A: The water slide closes in 15 minutes. Each trip takes 5 minutes. So Amy can slide 15 ÷ 5 = 3 times before it closes.

"Self-Consistency Improves Chain of Thought Reasoning in Language Models" by Wang et al. (2022)

Analogical prompting



0-shot

Model Input

Q: What is the area of the square with the four vertices at (-2, 2), (2, -2), (-2, -6), and (-6, -2)?

0-shot CoT

Model Input

Q: What is the area of the square with the four vertices at (-2, 2), (2, -2), (-2, -6), and (-6, -2)?

Think step by step.

- Generic guidance of reasoning

Few-shot CoT

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?

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

...

Q: What is the area of the square with the four vertices at (-2, 2), (2, -2), (-2, -6), and (-6, -2)?

- Need labeled exemplars of reasoning

Analogical Prompting (Ours)

Model Input

Q: What is the area of the square with the four vertices at (-2, 2), (2, -2), (-2, -6), and (-6, -2)?

Instruction:
Recall relevant exemplars:
Solve the initial problem:

Model Output

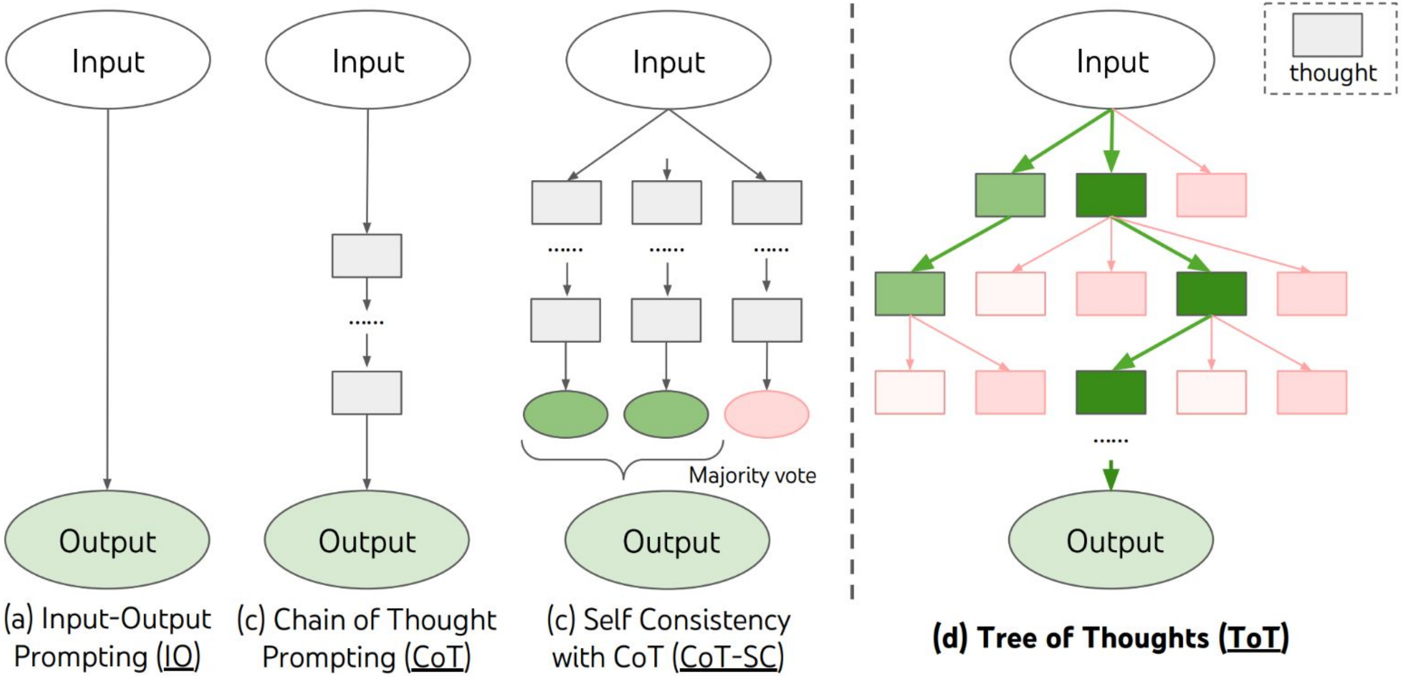
# Relevant exemplars:
Q: What is the area of the square with a side length of 5?
A: The area of a square is found by squaring the length of its side. So, the area of this square is 5^2 = 25. ...

# Solve the initial problem:
To find the area of the square, we need to find the side length. The length is … \sqrt{(2 - (-2))^2 + (-2 - 2)^2} = \sqrt{32}. So, the area of the square is (\sqrt{32})^2 = 32.

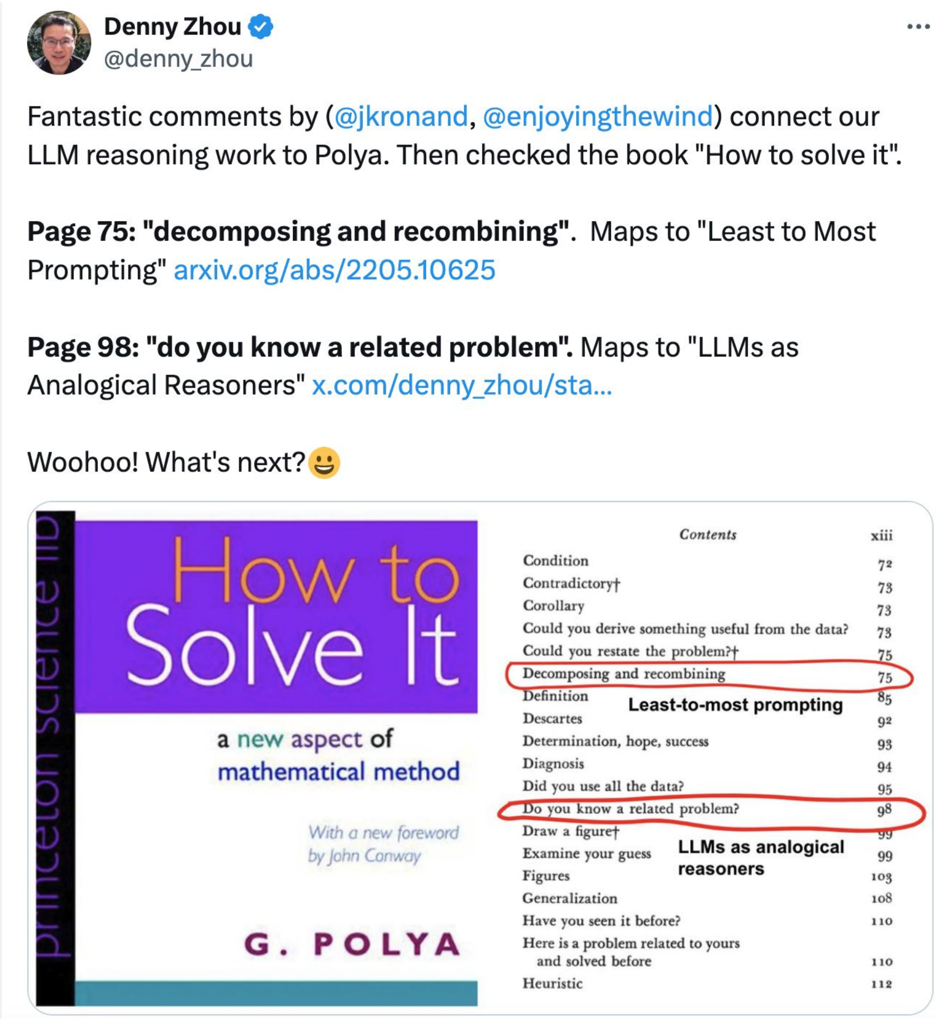
- Exemplars are **automatically** generated and **tailored** to each problem ✓

"Large Language Models as Analogical Reasoners" by Yasunaga et al. (2023)

Tree of Thoughts prompting


Input
Input
Input
Input
thought
......
......
......
......
......
Majority vote
Output
Output
Output
Output
(a) Input-Output Prompting (IO)
(c) Chain of Thought Prompting (CoT)
(c) Self Consistency with CoT (CoT-SC)
(d) Tree of Thoughts (ToT)

["Tree of Thoughts: Deliberate Problem Solving with Large Language Models"](#) *by Yao et al. (2023)*



Denny Zhou
@denny_zhou

Fantastic comments by (@jkronand, @enjoyingthewind) connect our LLM reasoning work to Polya. Then checked the book "How to solve it".

Page 75: "decomposing and recombining". Maps to "Least to Most Prompting" arxiv.org/abs/2205.10625

Page 98: "do you know a related problem". Maps to "LLMs as Analogical Reasoners" x.com/denny_zhou/sta...

Woohoo! What's next?😀

princeton science lib
How to Solve It
a new aspect of mathematical method
With a new foreword by John Conway
G. POLYA

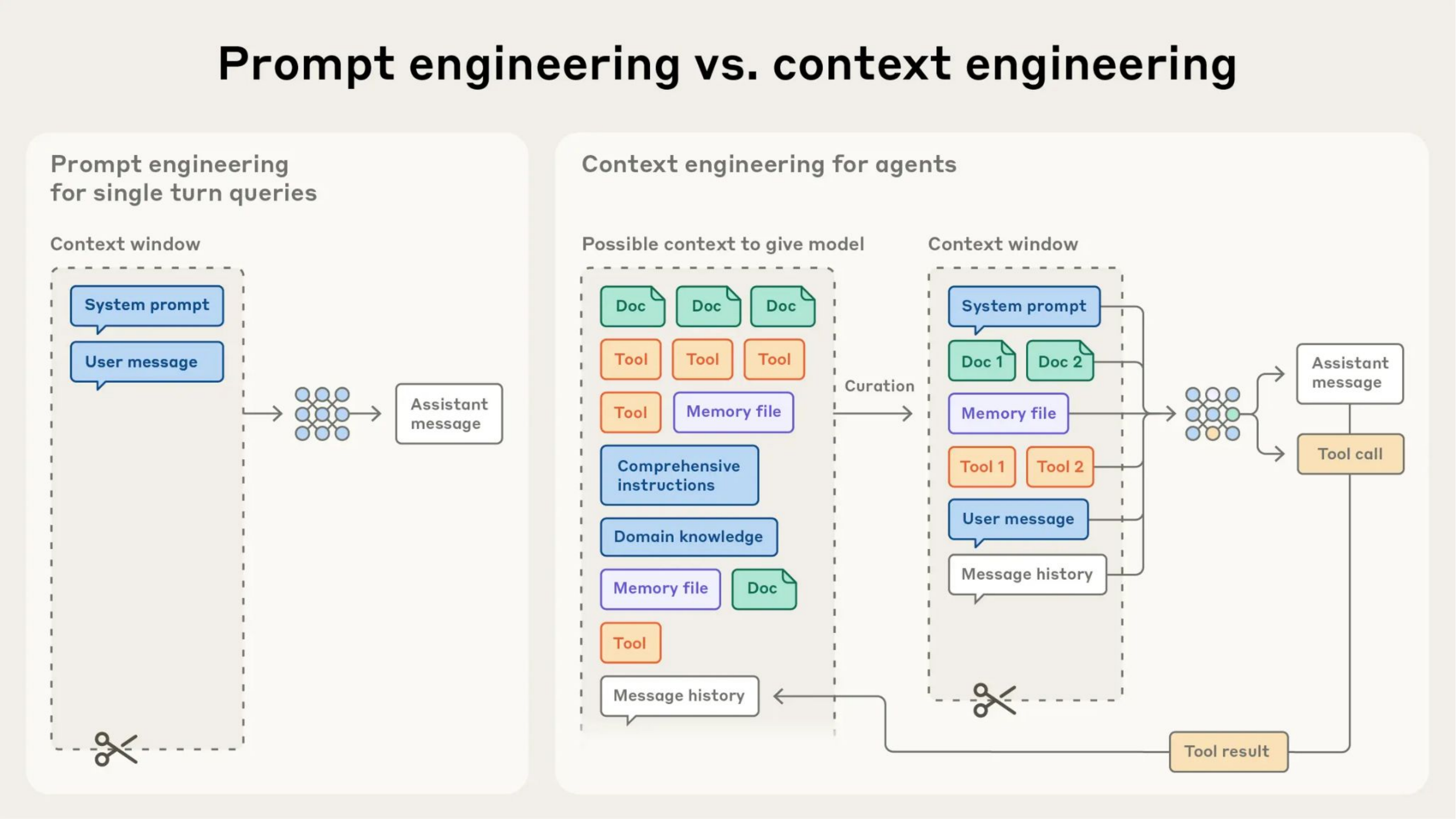
Contents xiii

Condition 72
Contradictory† 73
Corollary 73
Could you derive something useful from the data? 73
Could you restate the problem?† 75
Decomposing and recombining 75
Definition 85
Descartes 92
Determination, hope, success 93
Diagnosis 94
Did you use all the data? 95
Do you know a related problem? 98
Draw a figure† 99
Examine your guess 99
Figures 103
Generalization 108
Have you seen it before? 110
Here is a problem related to yours and solved before 110
Heuristic 112

Least-to-most prompting

LLMs as analogical reasoners

Context engineering



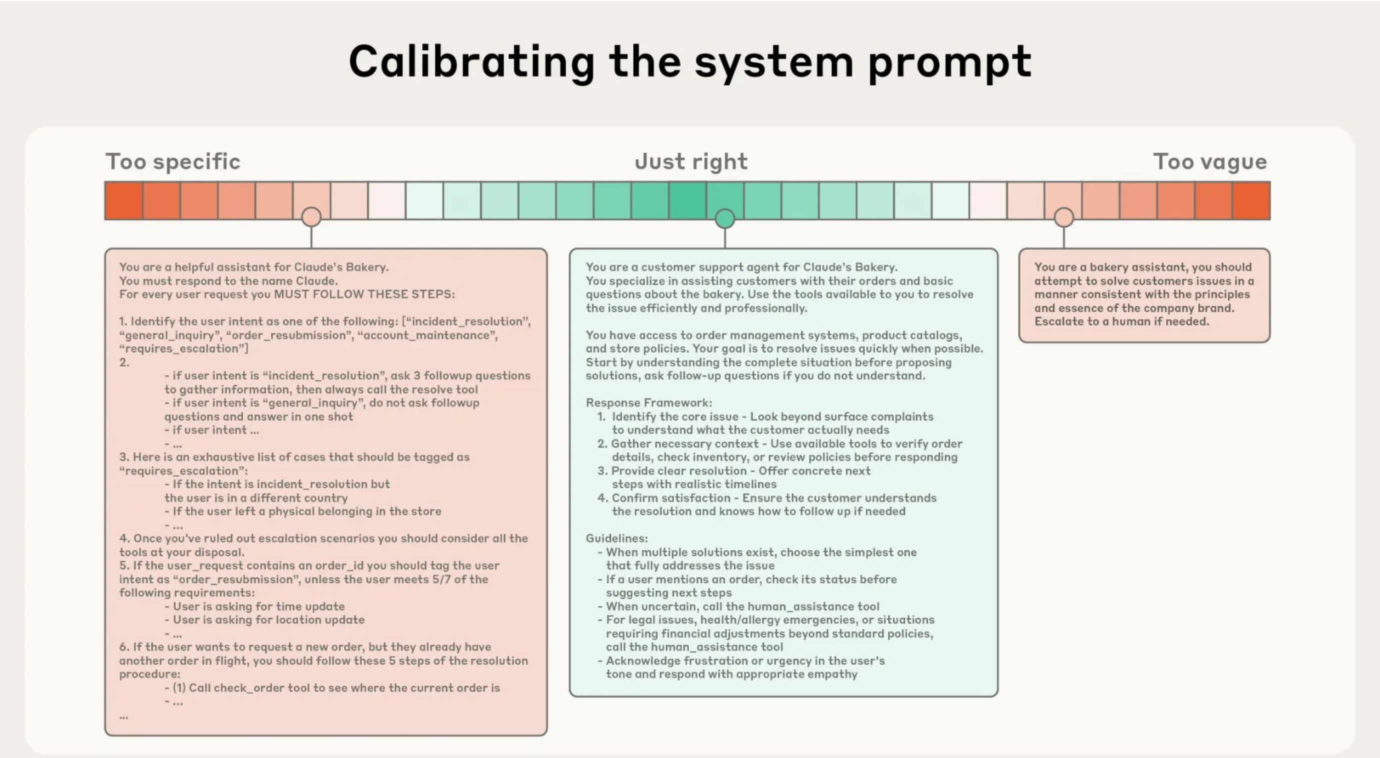
https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents

Context engineering (cont’d)

Building with language models is becoming less about finding the right words and phrases for your prompts, and more about answering the broader question of “what configuration of context is most likely to generate our model’s desired behavior?”

https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents

Context engineering (cont’d)



Calibrating the system prompt

You are a helpful assistant for Claude’s Bakery.
You must respond to the name Claude.
For every user request you MUST FOLLOW THESE STEPS:

1. Identify the user intent as one of the following: [“incident_resolution”, “general_inquiry”, “order_resubmission”, “account_maintenance”, “requires_escalation”]
2.
    - if user intent is “incident_resolution”, ask 3 followup questions to gather information, then always call the resolve tool
    - if user intent is “general_inquiry”, do not ask followup questions and answer in one shot
    - if user intent ...
    - ...
3. Here is an exhaustive list of cases that should be tagged as “requires_escalation”:
    - If the intent is incident_resolution but the user is in a different country
    - If the user left a physical belonging in the store
    - ...
4. Once you've ruled out escalation scenarios you should consider all the tools at your disposal.
5. If the user_request contains an order_id you should tag the user intent as “order_resubmission”, unless the user meets 5/7 of the following requirements:
    - User is asking for time update
    - User is asking for location update
    - ...
6. If the user wants to request a new order, but they already have another order in flight, you should follow these 5 steps of the resolution procedure:
    - (1) Call check_order tool to see where the current order is
    - ...

...

You are a customer support agent for Claude’s Bakery.
You specialize in assisting customers with their orders and basic questions about the bakery. Use the tools available to you to resolve the issue efficiently and professionally.

You have access to order management systems, product catalogs, and store policies. Your goal is to resolve issues quickly when possible. Start by understanding the complete situation before proposing solutions, ask follow-up questions if you do not understand.

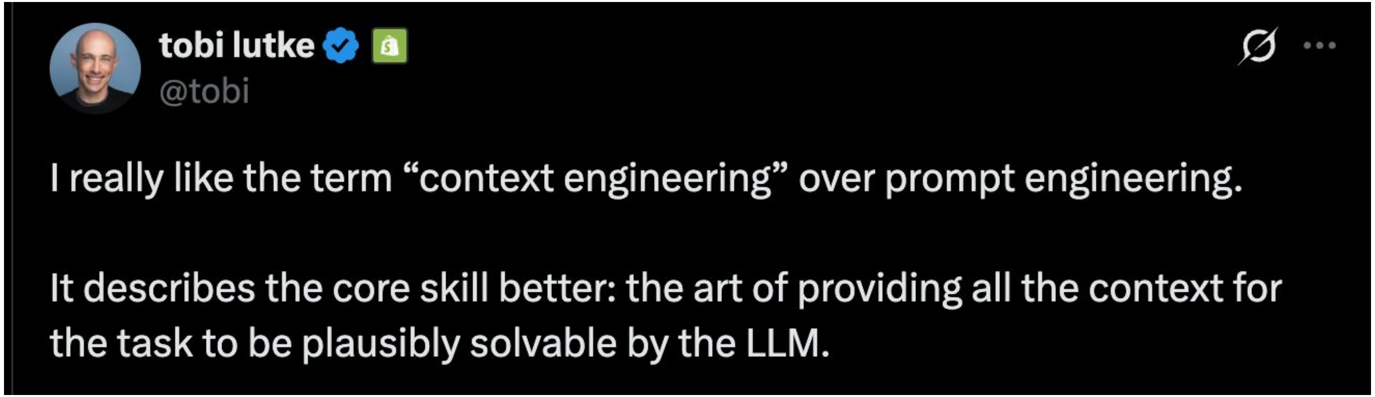
Response Framework:
1. Identify the core issue - Look beyond surface complaints to understand what the customer actually needs
2. Gather necessary context - Use available tools to verify order details, check inventory, or review policies before responding
3. Provide clear resolution - Offer concrete next steps with realistic timelines
4. Confirm satisfaction - Ensure the customer understands the resolution and knows how to follow up if needed

Guidelines:
- When multiple solutions exist, choose the simplest one that fully addresses the issue
- If a user mentions an order, check its status before suggesting next steps
- When uncertain, call the human_assistance tool
- For legal issues, health/allergy emergencies, or situations requiring financial adjustments beyond standard policies, call the human_assistance tool
- Acknowledge frustration or urgency in the user's tone and respond with appropriate empathy

You are a bakery assistant, you should attempt to solve customers issues in a manner consistent with the principles and essence of the company brand. Escalate to a human if needed.

System prompts should use simple, direct language and clearly explain what the model should do.

Context engineering (cont’d)


tobi lutke
@tobi

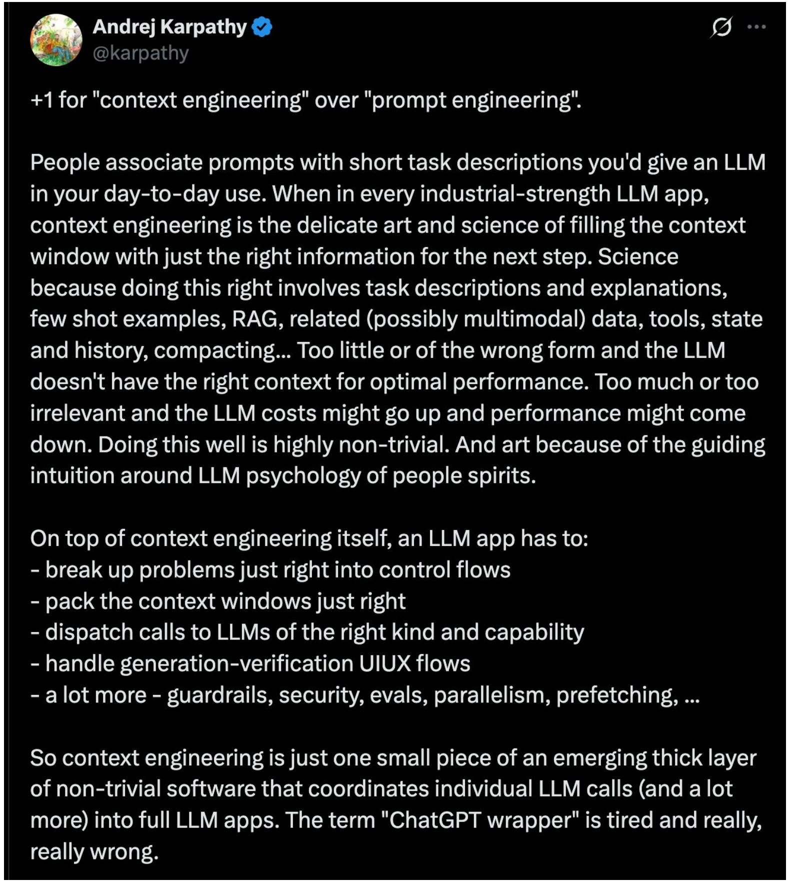
I really like the term “context engineering” over prompt engineering.

It describes the core skill better: the art of providing all the context for the task to be plausibly solvable by the LLM.

Context engineering is the art and science of curating what will go into the limited context window from that constantly evolving universe of possible information.

https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents

Context engineering (cont'd)



Andrej Karpathy
@karpathy

+1 for "context engineering" over "prompt engineering".

People associate prompts with short task descriptions you'd give an LLM in your day-to-day use. When in every industrial-strength LLM app, context engineering is the delicate art and science of filling the context window with just the right information for the next step. Science because doing this right involves task descriptions and explanations, few shot examples, RAG, related (possibly multimodal) data, tools, state and history, compacting... Too little or of the wrong form and the LLM doesn't have the right context for optimal performance. Too much or too irrelevant and the LLM costs might go up and performance might come down. Doing this well is highly non-trivial. And art because of the guiding intuition around LLM psychology of people spirits.

On top of context engineering itself, an LLM app has to:
- break up problems just right into control flows
- pack the context windows just right
- dispatch calls to LLMs of the right kind and capability
- handle generation-verification UIUX flows
- a lot more - guardrails, security, evals, parallelism, prefetching, ...

So context engineering is just one small piece of an emerging thick layer of non-trivial software that coordinates individual LLM calls (and a lot more) into full LLM apps. The term "ChatGPT wrapper" is tired and really, really wrong.

https://x.com/karpathy/status/1937902205765607626

Thank you!