# Large Reasoning Models

By Junqi Qu

# What is LLM Reasoning

- Input -> intermediate steps/tokens -> output

- Ling, Wang, et al. "Program induction by rationale generation: Learning to solve and explain algebraic word problems." *arXiv preprint arXiv:1705.04146* (2017).

# What is the output when concatenating the last letter of each word in "artificial intelligence"

- No reasoning: The answer is "le".
- Reasoning: The last letter of "atificial" is "l". The last letter of "intelligence" is "e". Concatenating "I" and  "e" leads to "le". So the answer is "le".

# Why “Intermediate Tokens” / “Reasoning” Matters?

- For any problems solvable by boolean circuits of size T, **constant-size transformers** can solve it by generating O(T) intermediate tokens
- If directly generating final answers, either requires a huge depth or cannot solve at all

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of Thought Empowers Transformers to Solve Inherently Serial Problems. ICLR 2024.

# Check more candidates

- I have 3 apples. My dad has 2 more apples than me. How many apples do we have in total?

5 apples. (Greedy Decoding)

I have 3 apples, my dad has 2 more than mee, so he has 5. 3 + 5 = 8.

We have 8 apples in total.

You have 3 apples, your dad has 2 more apples than you, so he has 5 apples. 3+5=8.

The answer is 5.

Top k = 5

$P(xn|x<n)$

Chain-of-Thought Reasoning Without Prompting, https://arxiv.org/abs/2402.10200

# Chain-of-Thought Decoding

1. Go beyond greedy decoding by checking more generation candidates
2. Choose candidates which have the highest confidence on the final answer

(t, t+n)

$$\Delta_{k,\text{answer}} = \frac{1}{|\,\text{answer}\,|} \sum_{x_t \in \text{answer}} \left[ p(x_t^{(1)} \mid x_{<t}) - p(x_t^{(2)} \mid x_{<t}) \right].$$

## Chain-of-Thought Prompting

Q: Elsa has 3 apples. Anna has 2 more apples than Elsa. How many apples do they have together?

A: Anna has 2 more apples than Elsa. So Anna has 2 + 3 = 5 apples. So Elsa and Anna have 3 + 5 = 8 apples together.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. NeurIPS 2022

## Let's Think Step by Step

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

Let's think step by step.

Kojima, T., Gu, S.S., Reid, M., Matsuo, Y. and Iwasawa, Y. Large language models are zero-shot reasoners. NeurIPS 2022.

# Why it works?

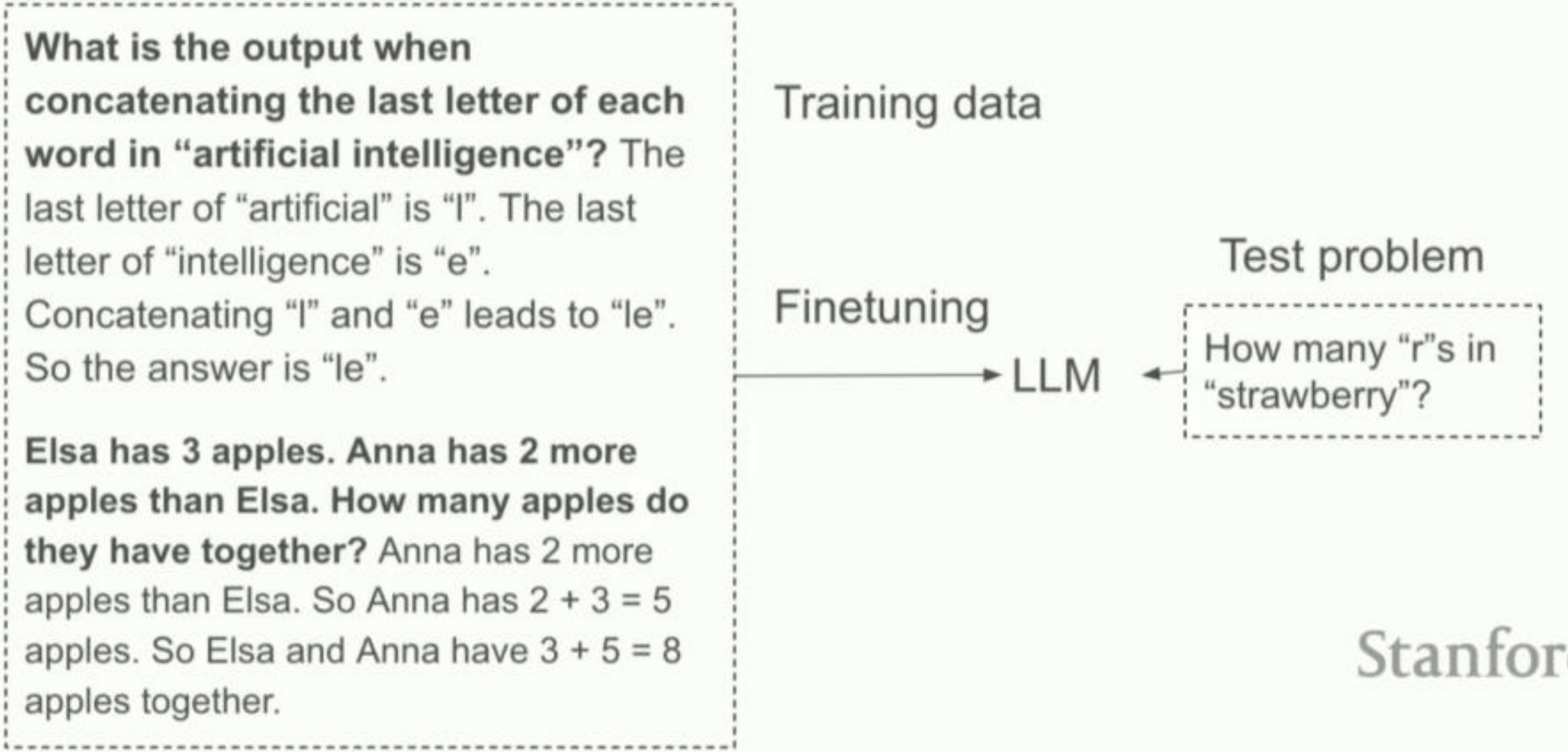
- Changes the LLM output distribution by adding more prompt
- $P(xt|x<t,c)$
- Simple and works

# Supervised Finetuning(SFT)

- Step 1:collect a set of problems and their step-by-step solutions from human annotators
- Step 2:maximize the likelihood of human solutions

- Program Induction by Rationale Generation : Learning to Solve and Explain Algebraic Word Problems, https://arxiv.org/abs/1705.04146
- Training Verifiers to Solve Math Word Problems, https://arxiv.org/abs/2110.14168

# Supervised Finetuning (SFT)



**What is the output when concatenating the last letter of each word in “artificial intelligence”?** The last letter of “artificial” is “l”. The last letter of “intelligence” is “e”. Concatenating “l” and “e” leads to “le”. So the answer is “le”.

**Elsa has 3 apples. Anna has 2 more apples than Elsa. How many apples do they have together?** Anna has 2 more apples than Elsa. So Anna has 2 + 3 = 5 apples. So Elsa and Anna have 3 + 5 = 8 apples together.

Training data

Finetuning

LLM

Test problem

How many “r”s in “strawberry”?

# Pros and cons

- Pros
  - Can be used to do on different tasks(datasets)
- Cons
  - Does not generalize well
  - Scaling does not help

# How to Fix the Generalization Failure from SFT?

**SFT procedure**

Step 1: collect a set of problems and their step-by-step solutions from ~~human annotators~~

Step 2: maximize the likelihood of ~~human solutions~~

# First Attempt: Self-Improve

Step 1: collect a set of problems and their step-by-step solutions generated from the model

Step 2: maximize the likelihood of correct solutions

Zelikman E, Wu Y, Mu J, Goodman N. Star: Bootstrapping reasoning with reasoning. NeurIPS 2022.

Huang J, Gu SS, Hou L, Wu Y, Wang X, Yu H, Han J. Large language models can self-improve. arXiv:2210.11610. 2022

# RL Finetuning

- Repeat this process:
- Step 1: collect a set of problems and their step-by-step solutions **generated from the model**
- Step 2: maximize the likelihood of **correct solutions**, minimize the likelihood of wrong solutions

- ReFT: Reasoning with Reinforced Fine-Tuning, arXiv:2401.08967

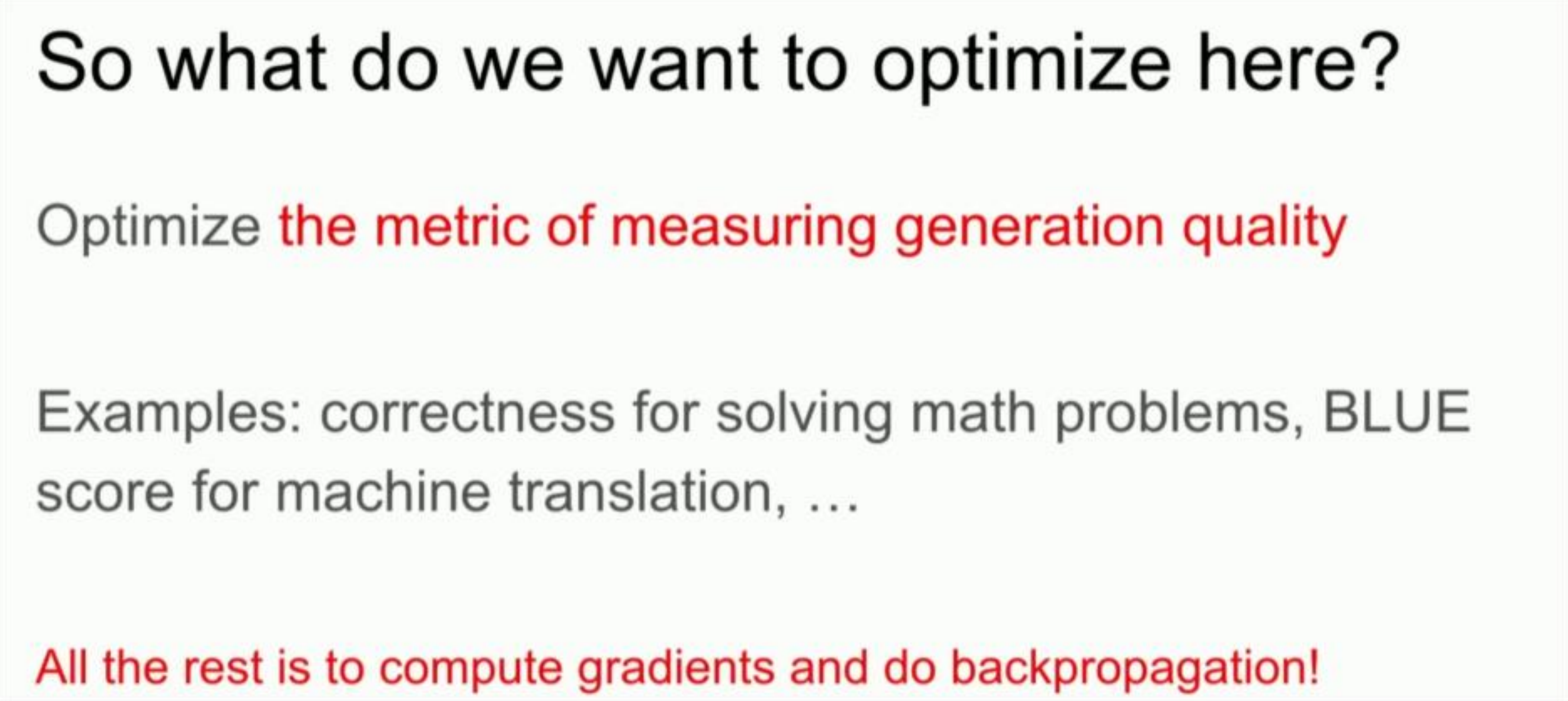
# A reliable verifier is the most crucial in RL finetuning, not the RL algorithm

> “Verification, the key to AI”
>
> — Rich Sutton, 11/15/2001

# Why “generated from the model” instead of “from humans”?

First principle in ML:

Directly optimize what we want!

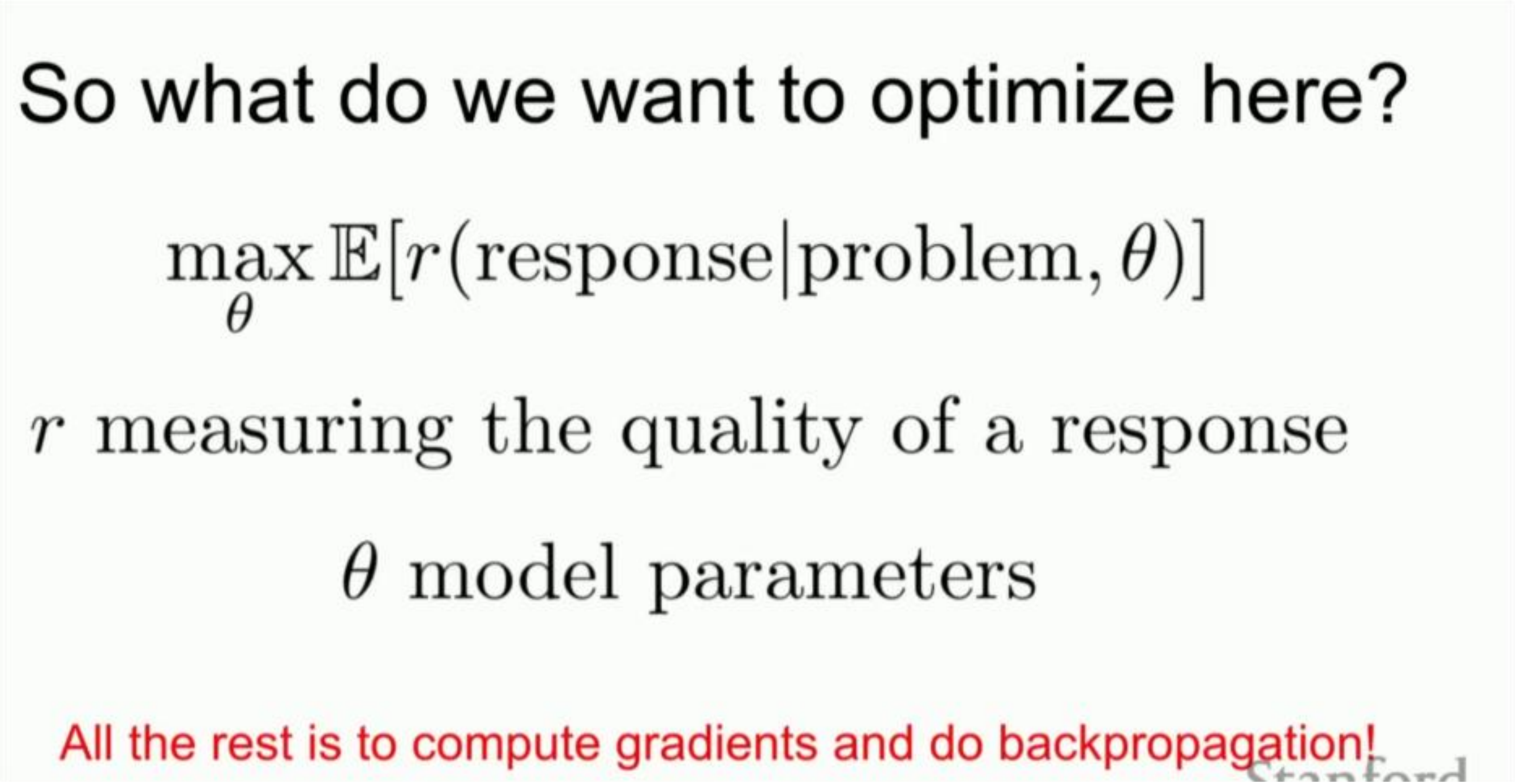


# So what do we want to optimize here?

Optimize the metric of measuring generation quality

Examples: correctness for solving math problems, BLUE score for machine translation, …

All the rest is to compute gradients and do backpropagation!



# So what do we want to optimize here?

$$\max_{\theta} \mathbb{E}[r(\text{response}|\text{problem}, \theta)]$$

$r$ measuring the quality of a response

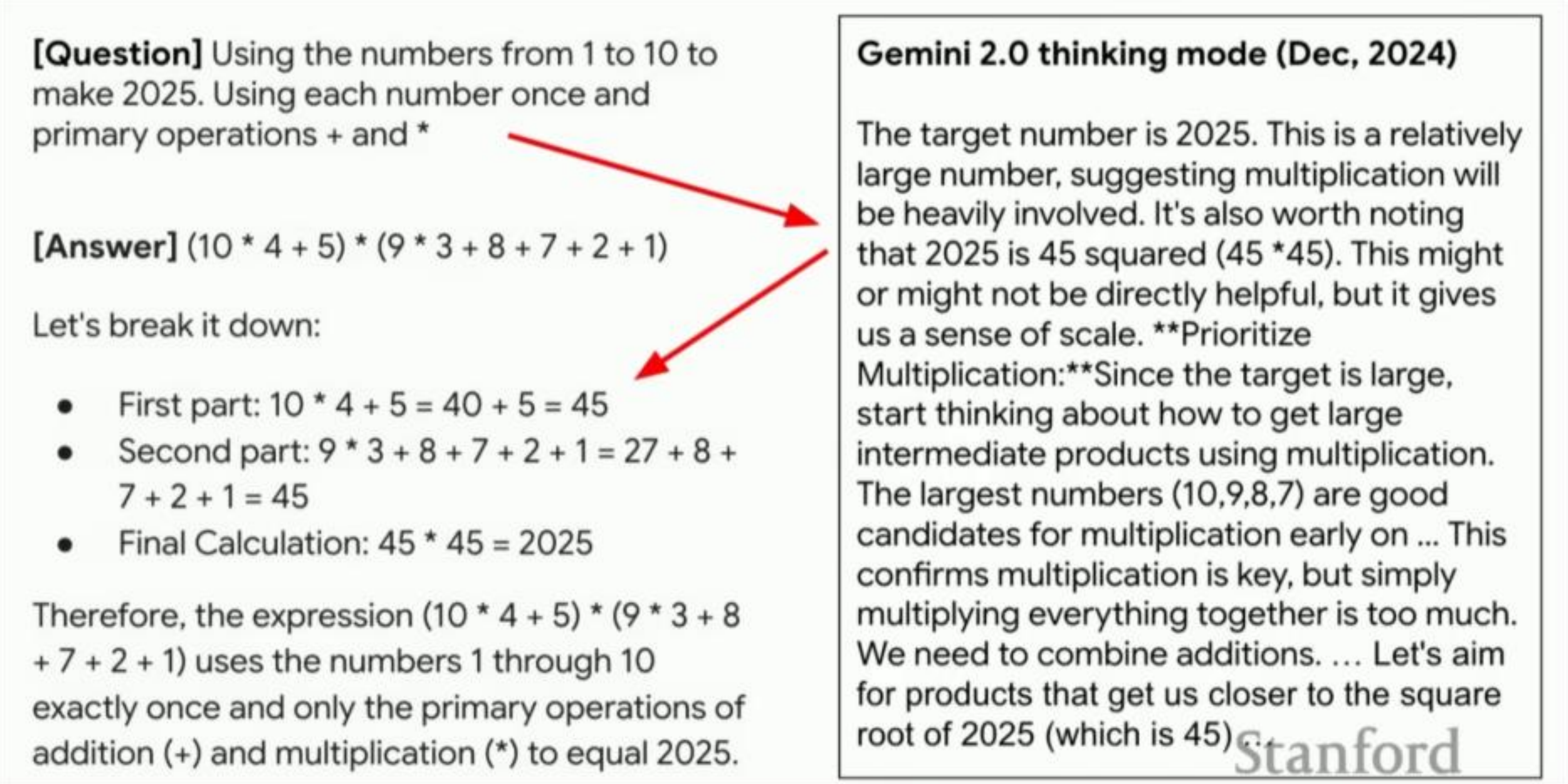
$\theta$ model parameters

All the rest is to compute gradients and do backpropagation!

# Scaling Reasoning Training: What to Scale?

- For any problems solvable by boolean circuits of size T, **constant-size transformers** can solve it by generating O(T) intermediate tokens — Scaling output length!
- If directly generating final answers, either requires a huge depth to solve or cannot solve at all — Scaling model depth!

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of Thought Empowers Transformers to Solve Inherently Serial Problems. ICLR 2024.



**[Question]** Using the numbers from 1 to 10 to make 2025. Using each number once and primary operations + and *

**[Answer]** (10 * 4 + 5) * (9 * 3 + 8 + 7 + 2 + 1)

Let's break it down:

- First part: 10 * 4 + 5 = 40 + 5 = 45
- Second part: 9 * 3 + 8 + 7 + 2 + 1 = 27 + 8 + 7 + 2 + 1 = 45
- Final Calculation: 45 * 45 = 2025

Therefore, the expression (10 * 4 + 5) * (9 * 3 + 8 + 7 + 2 + 1) uses the numbers 1 through 10 exactly once and only the primary operations of addition (+) and multiplication (*) to equal 2025.

**Gemini 2.0 thinking mode (Dec, 2024)**

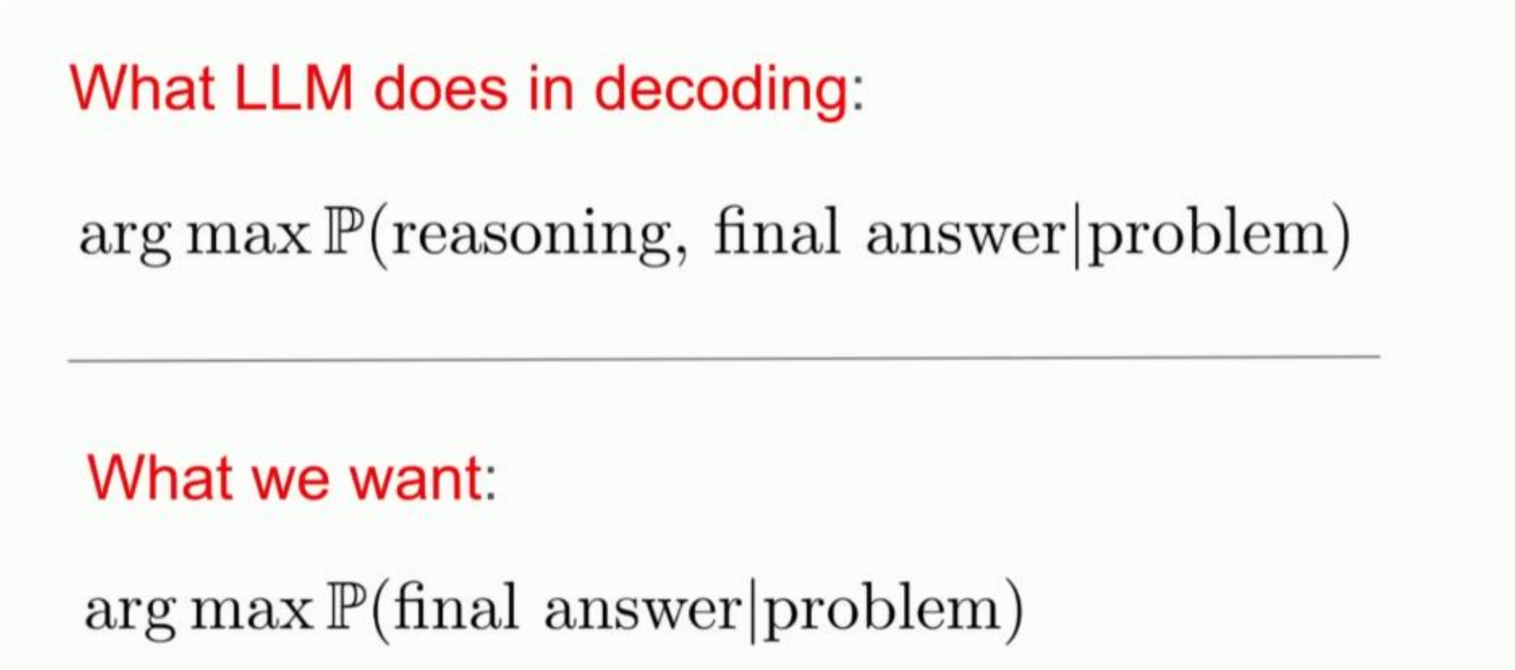
The target number is 2025. This is a relatively large number, suggesting multiplication will be heavily involved. It's also worth noting that 2025 is 45 squared (45 *45). This might or might not be directly helpful, but it gives us a sense of scale. **Prioritize Multiplication:**Since the target is large, start thinking about how to get large intermediate products using multiplication. The largest numbers (10,9,8,7) are good candidates for multiplication early on ... This confirms multiplication is key, but simply multiplying everything together is too much. We need to combine additions. … Let's aim for products that get us closer to the square root of 2025 (which is 45)...

# Pros and Cons of RL finetuning

- Pros
  - Generalize well for automatically verifiable tasks
- Cons
  - Not all tasks are automatically verifiable
- RLFTs: Direct Preference Optimization, Proximal Policy Optimization, Group-relative policy optimization

# Furthermore, on LLMs

# Aggregation and Retrieval



What LLM does in decoding:

$$\arg\max \mathbb{P}(\text{reasoning},\ \text{final answer}|\text{problem})$$

What we want:

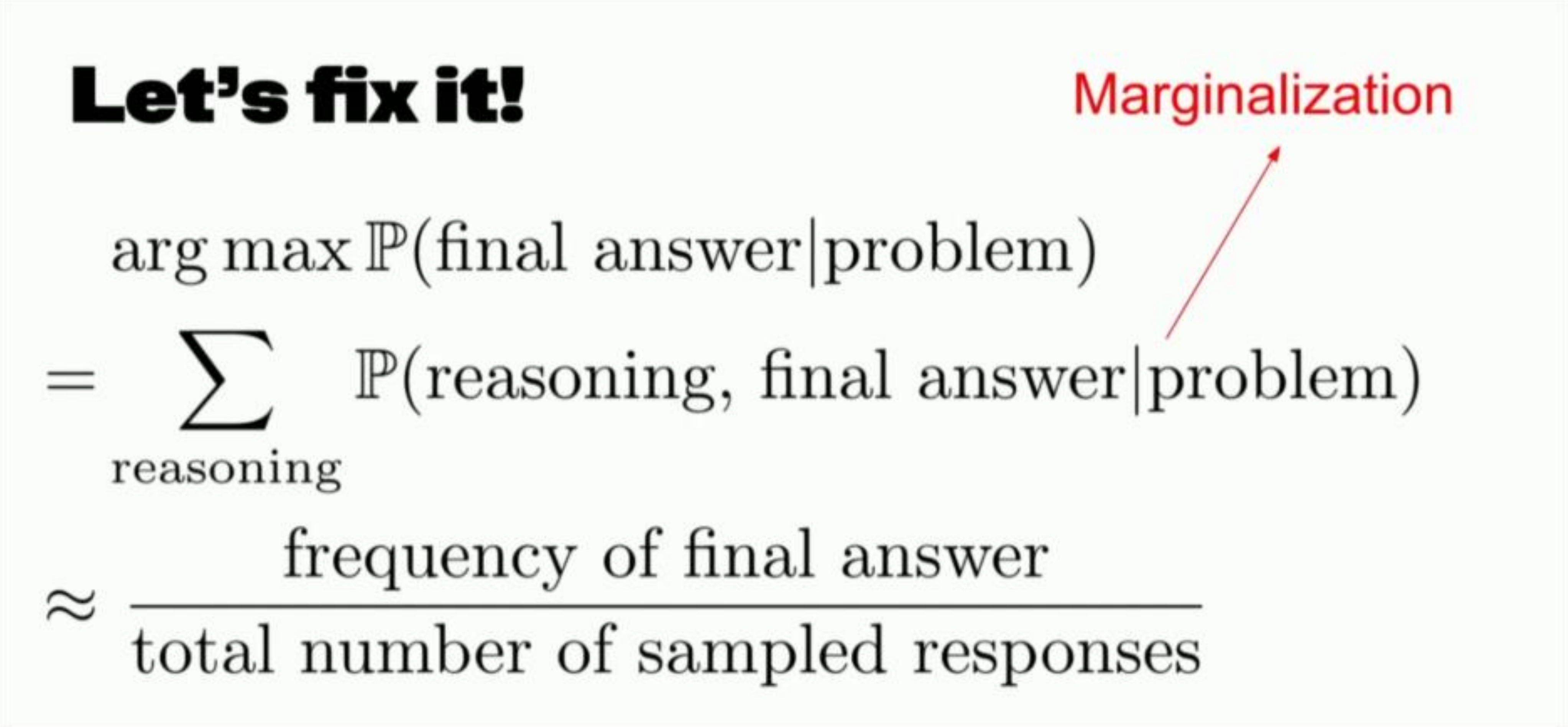
$$\arg\max \mathbb{P}(\text{final answer}|\text{problem})$$

**What is the area of the square with the four vertices at (-2, 2), (2, -2), (-2, -6), and (-6, -2)?** Recall a related problem, and then solve this one.

**The related problem is finding the distance between two points on a coordinate plane**. To find the distance between two points, we use the distance formula: d = √((x2-x1)^2 + (y2-y1)^2).

In this problem, we have four points: (-2, 2), (2, -2), (-2, -6), and (-6, -2). Let's choose (-2, 2) and (2, -2) to find the distance d between them: d = √((2-(-2))^2 + (-2-2)^2) = √(4^2 + (-4)^2) = √(16 + 16) = √32. We know that the distance between two opposite sides of a square is equal. So, the length of each side of the square is √32. To find the area of the square, we square the length of one side: Area = (√32)^2 = 32.

Michihiro Yasunaga, Xinyun Chen, Yujia Li, Panupong Pasupat, Jure Leskovec, Percy Liang, Ed H. Chi, Denny Zhou. Large Language Models as Analogical Reasoners. ICLR 2024.



## Let's fix it!

Marginalization

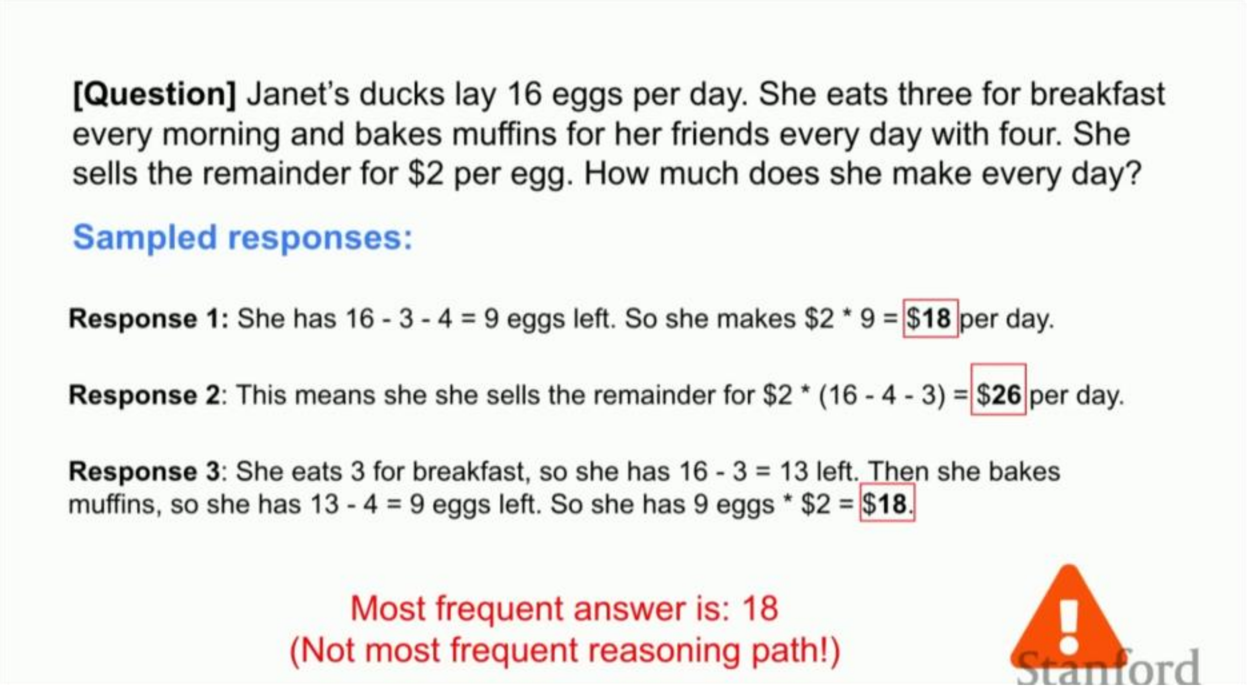
$$\arg\max \mathbb{P}(\text{final answer}|\text{problem})$$

$$= \sum_{\text{reasoning}} \mathbb{P}(\text{reasoning, final answer}|\text{problem})$$

$$\approx \frac{\text{frequency of final answer}}{\text{total number of sampled responses}}$$

# Self-Consistency

1. Generate multiple responses by randomly sampling
2. Choose the answer that appears most frequently



**[Question]** 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?

**Sampled responses:**

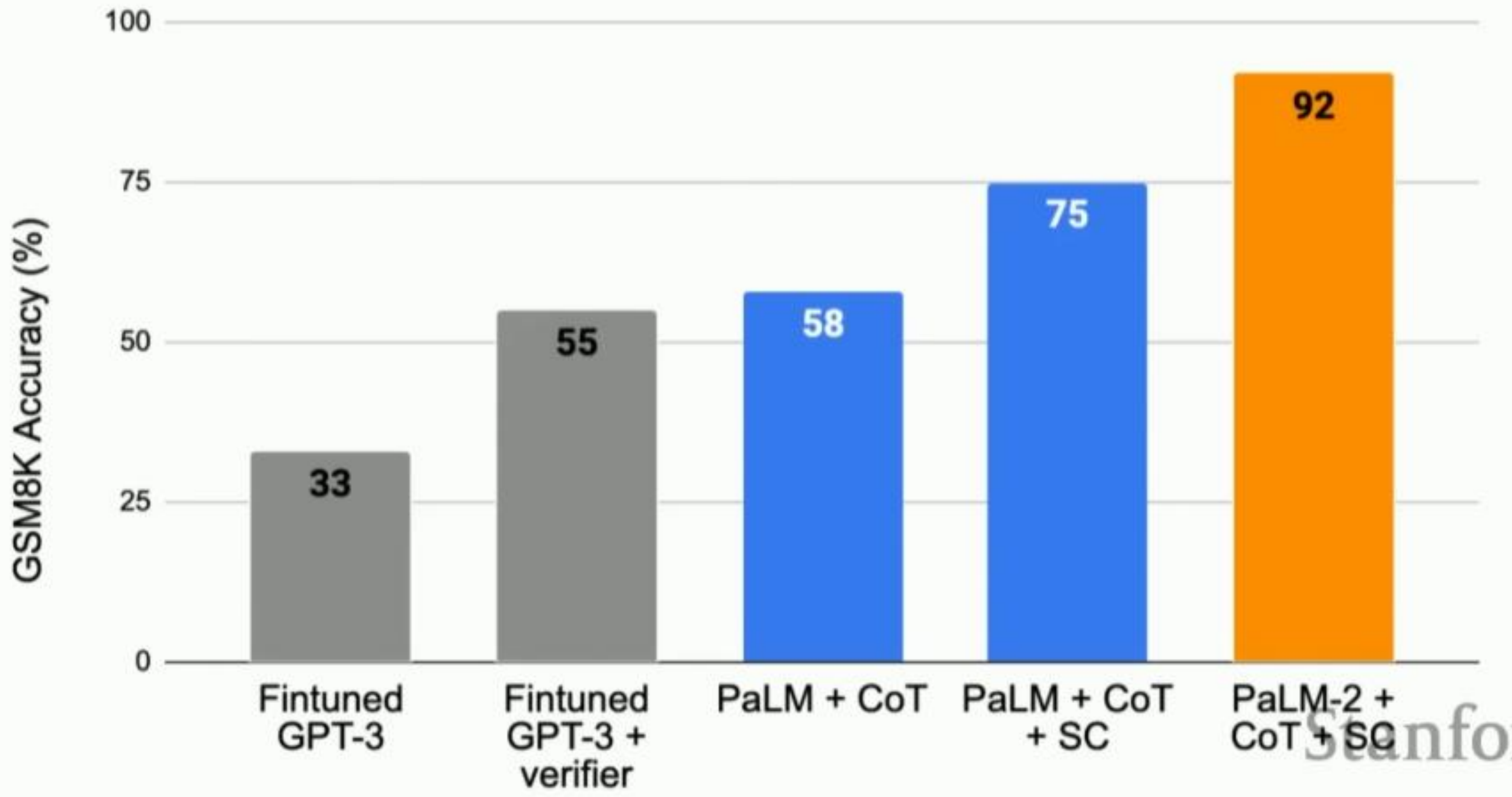
**Response 1:** She has 16 - 3 - 4 = 9 eggs left. So she makes $2 * 9 = **$18** per day.

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

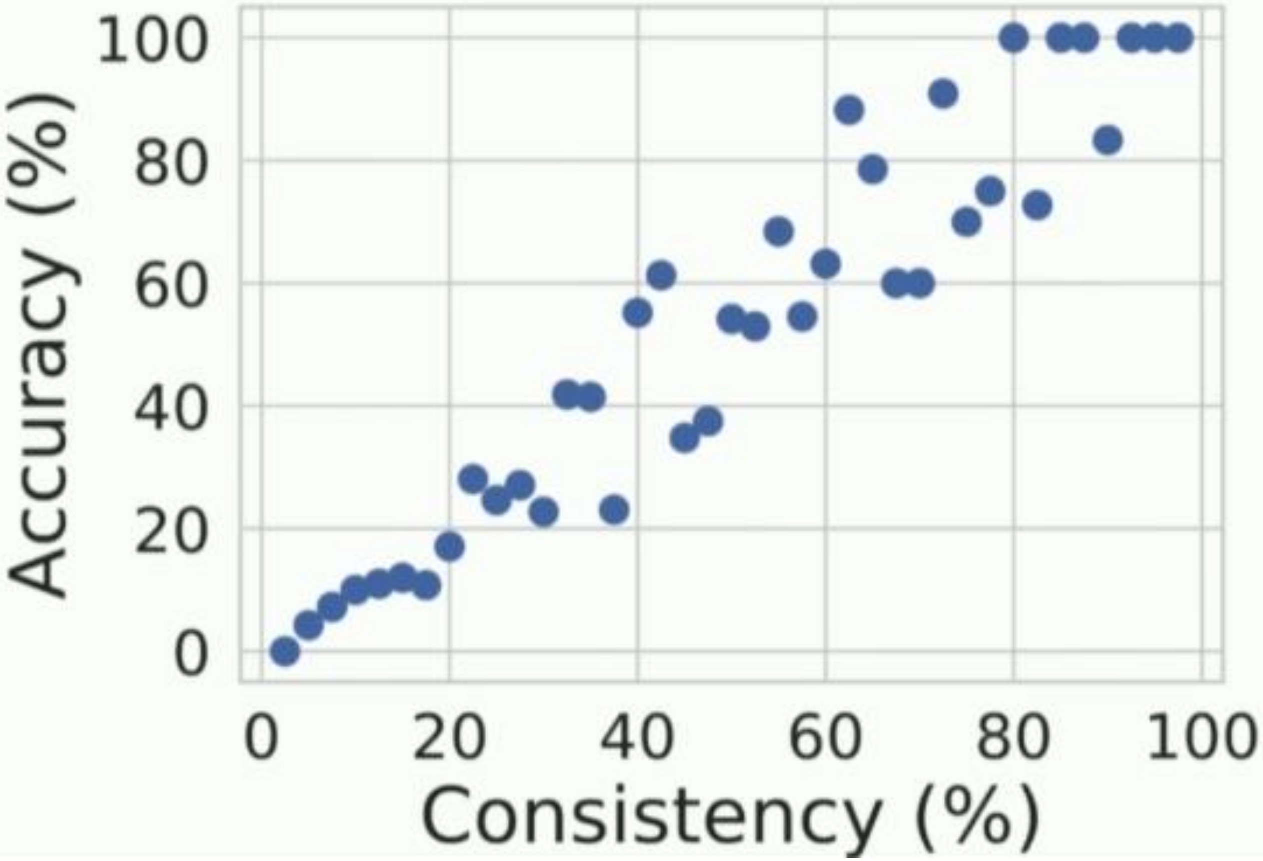
**Response 3**: She eats 3 for breakfast, so she has 16 - 3 = 13 left. Then she bakes muffins, so she has 13 - 4 = 9 eggs left. So she has 9 eggs * $2 = **$18**.

Most frequent answer is: 18
(Not most frequent reasoning path!)

# Results on GSM8K (8 shots, Jan 2022/3)



# Higher Consistency Indicates Higher Accuracy



Stanford

# Tokenization

- Language model places a probability over sequence of tokens(ints)
- https://tiktokenizer.vercel.app/?encoder=gpt2

# Types of tokenization

- Character, byte, word
  - Why they are bad
  - Dict(apple, 200)

# Byte Pair Encoding(BPE)

- https://en.wikipedia.org/wiki/Byte_pair_encoding
- The BPE algorithm was introduced by Philip Gage in 1994 for data compression.