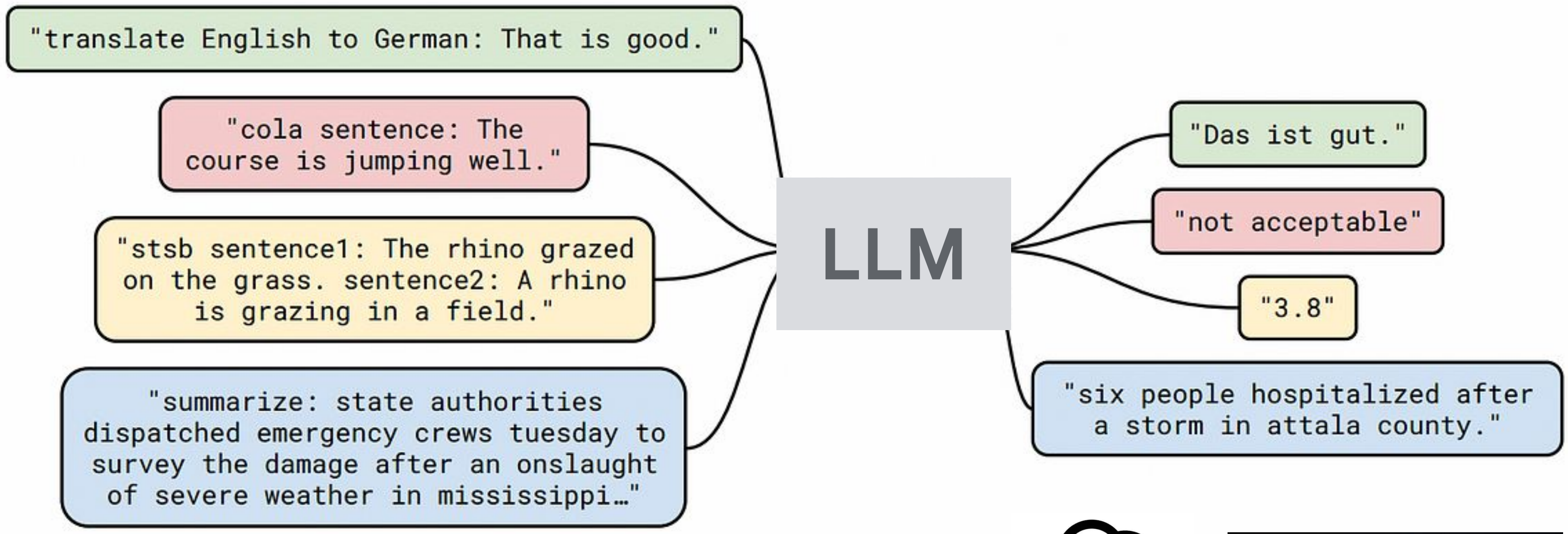
Google DeepMind

Improving large language models using self-generated data

Rishabh Agarwal
Research Scientist, Google DeepMind

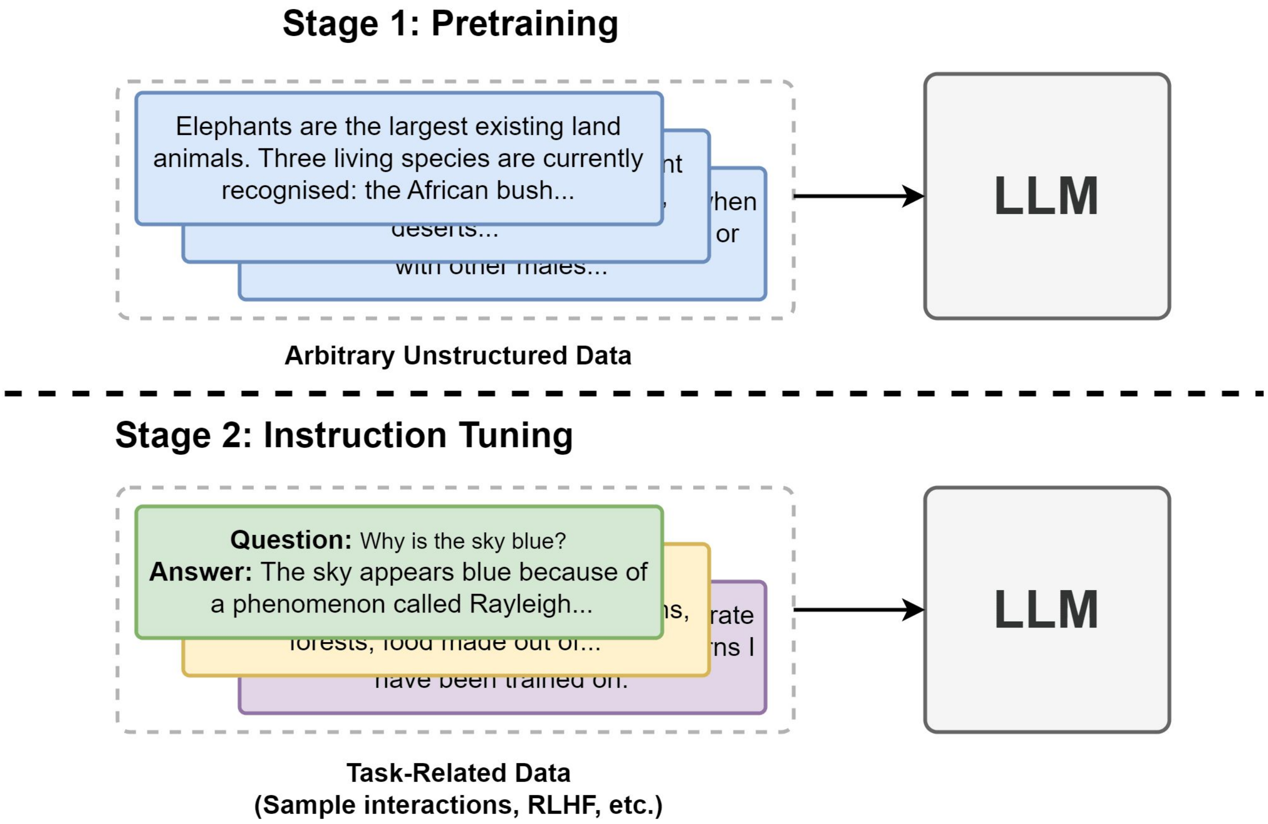
Large Language Models (LLMs)



GPT - 4

Gemini

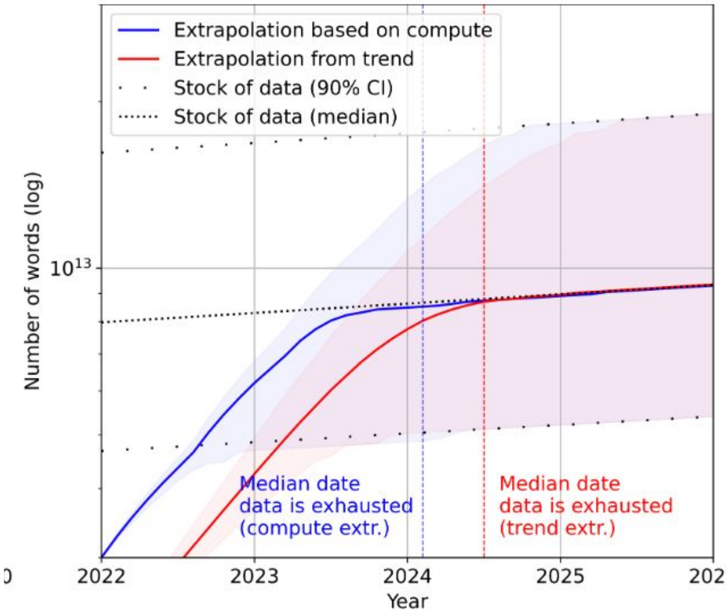
Training LLMs needs high quality data



High quality data, scraped from web or collected from humans.

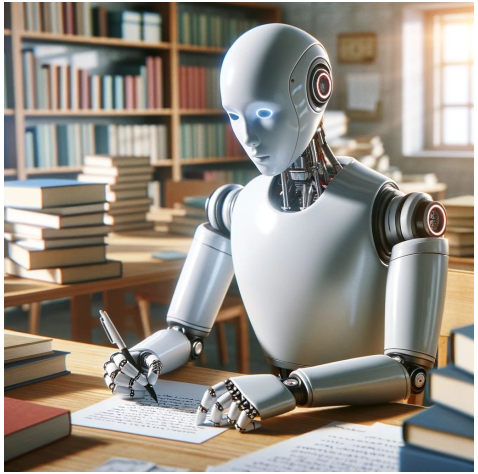
Google DeepMind

Are we running out of high-quality data?



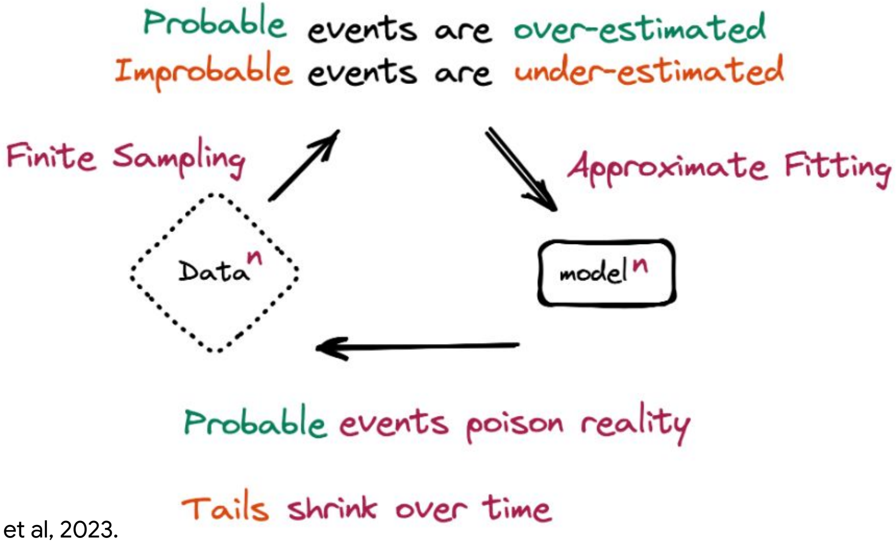
epochai.org

Synthetic data to the rescue?



What if the models could generate their own training data?

Naively doing so can result in model collapse!



The Curse of Recursion: Training on Generated Data Makes Models Forget. Shumailov et al, 2023.

Synthetic data to the rescue?

Verification can often be easier than Generation!

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Solving sudoku puzzles is harder than checking one!

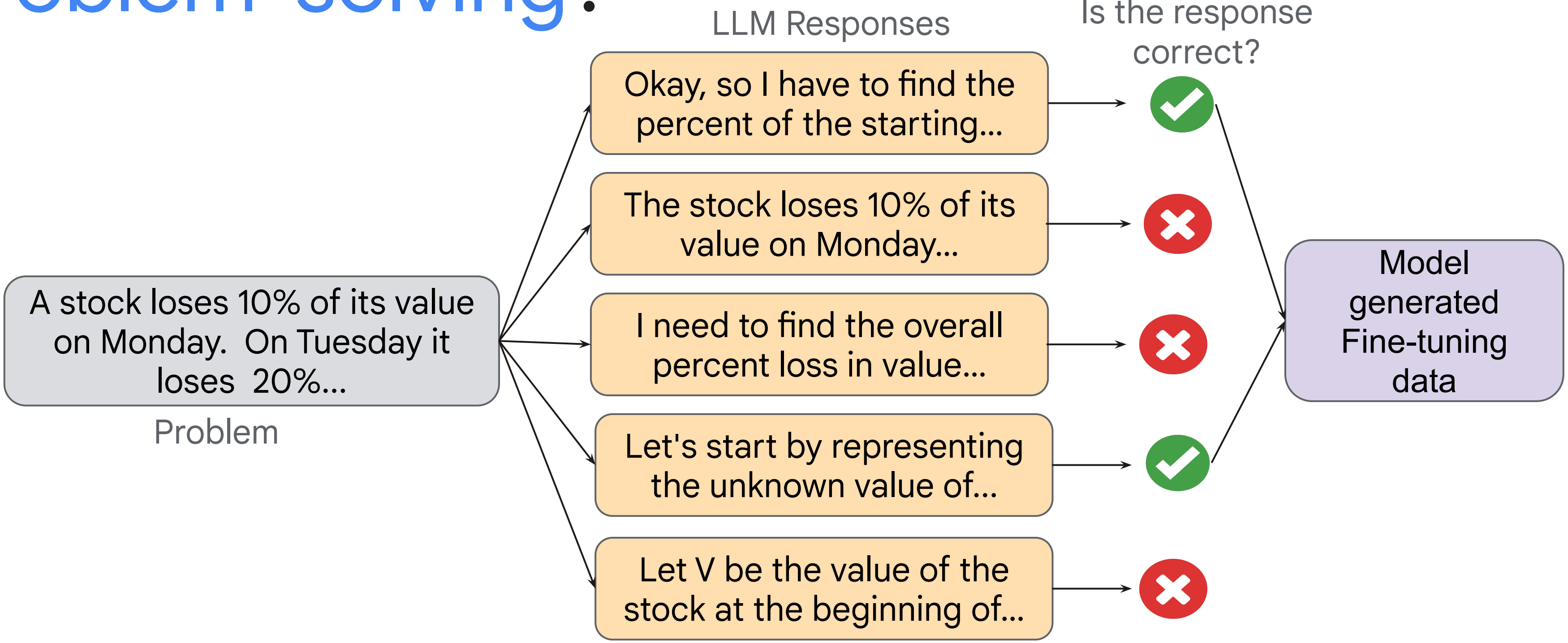
GIVEN A STRING, FIND THE LENGTH OF THE LONGEST SUBSTRING WITHOUT REPEATING CHARACTERS.

Generating code can be harder than verifying it via test case execution.

Can we use model-generated data for training given access to some form of feedback?

Google DeepMind

How do we self-generate data for problem-solving?



Google DeepMind

A simple recipe for self-training (ReST^{EM})

Repeat this process a few times:

1. **Generate samples from the model and filter them using binary feedback. (E-step)**
2. **Fine-tune the model on these samples (M-step)**

This process corresponds to **expectation-maximization based RL!** Check the math in the paper.

Beyond Human Data: Scaling Self-Training for Problem-Solving with Language Models (TMLR) 2023. Singh*, Co-reyes*, Agarwal* et al

Google DeepMind

Problem-Solving tasks: Math & Coding

Hendrycks MATH

Problem: The equation $x^2 + 2x = i$ has two complex solutions. Determine the product of their real parts.
Solution: Complete the square by adding 1 to each side. Then $(x+1)^2 = 1 + i = e^{\frac{i\pi}{4}}\sqrt{2}$, so $x + 1 = \pm e^{\frac{i\pi}{8}}\sqrt[4]{2}$. The desired product is then

$$\left(-1+\cos\left(\frac{\pi}{8}\right)\sqrt[4]{2}\right)\left(-1-\cos\left(\frac{\pi}{8}\right)\sqrt[4]{2}\right) =$$

$$1-\cos^2\left(\frac{\pi}{8}\right)\sqrt{2} = 1 - \frac{\left(1+\cos\left(\frac{\pi}{4}\right)\right)}{2}\sqrt{2} = \boxed{\frac{1-\sqrt{2}}{2}}.$$

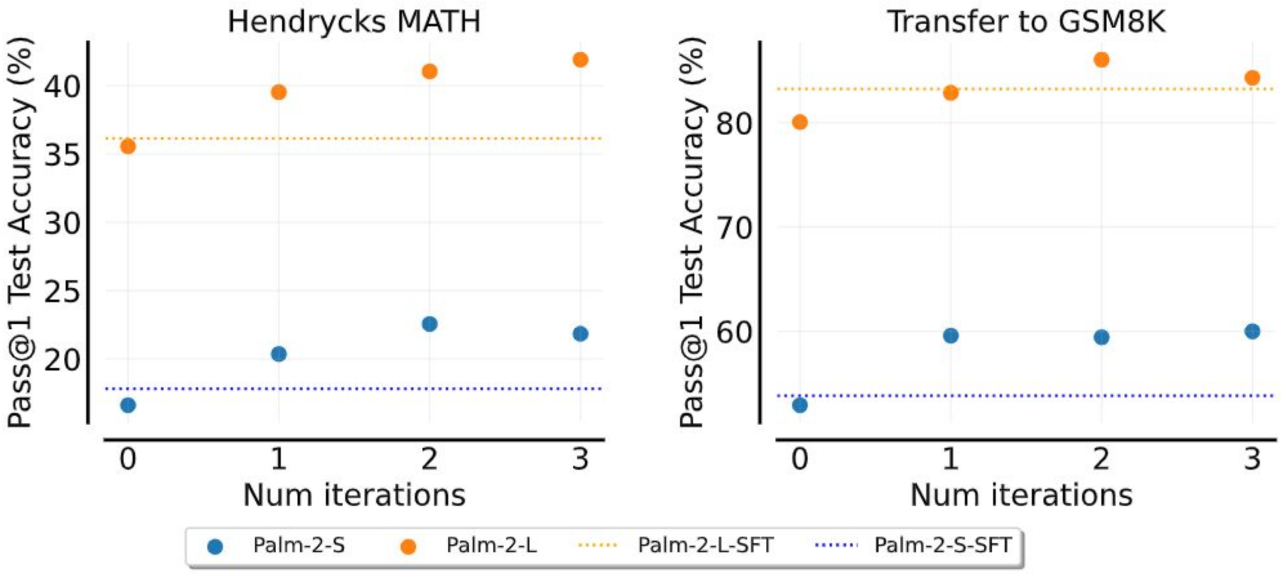
APPS Coding (Intro)

We will buy a product for N yen (the currency of Japan) at a shop. If we use only 1000-yen bills to pay the price, how much change will we receive? Assume we use the minimum number of bills required.
-----Constraints----- - 1 \leq N \leq 10000 - N is an integer.
-----Input----- Input is given from Standard Input in the following format: N
-----Output----- Print the amount of change as an integer.
-----Sample Input-----
1900
-----Sample Output-----
100
We will use two 1000-yen bills to pay the price and receive 100 yen in change.

Beyond Human Data: Scaling Self-Training for Problem-Solving with Language Models (TMLR) 2023. Singh*, Co-reyes*, Agarwal* et al

Google DeepMind

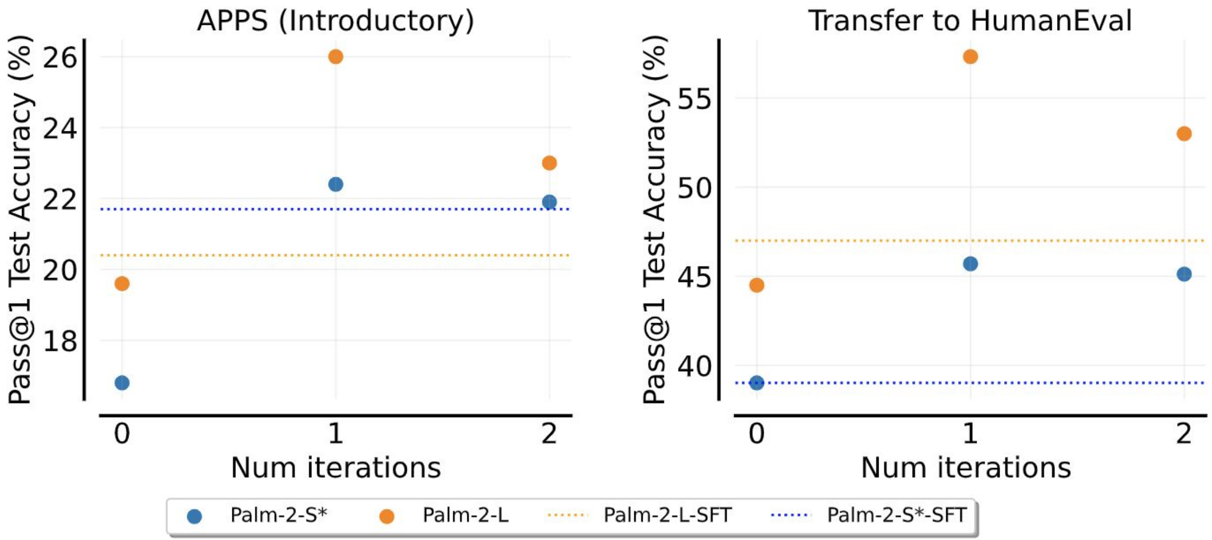
This... beats human data!



Beyond Human Data: Scaling Self-Training for Problem-Solving with Language Models. 2023. Singh*, Co-reyes*, Agarwal* et al

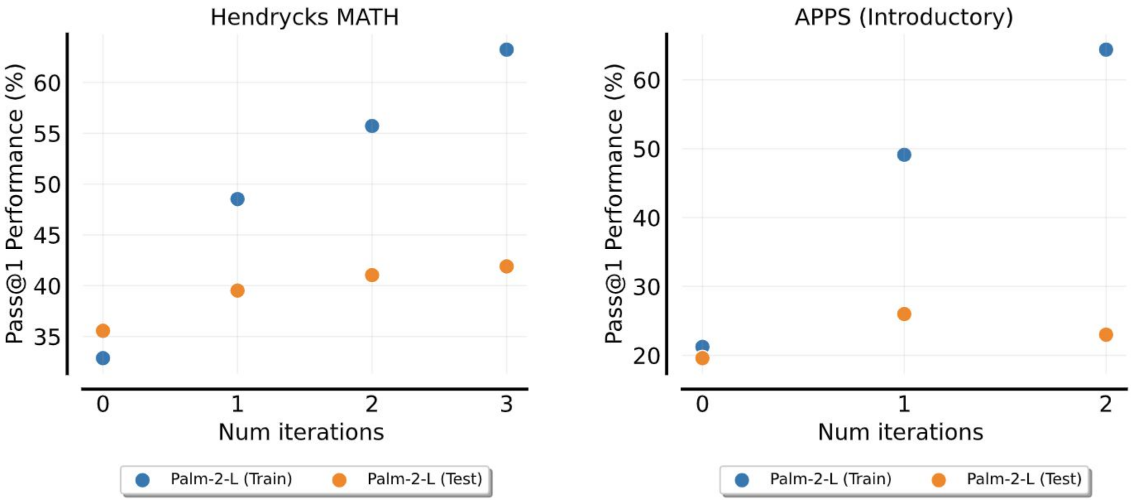
Google DeepMind

ReST^{EM} works on coding too.



Beyond Human Data: Scaling Self-Training for Problem-Solving with Language Models. 2023. Singh*, Co-reyes*, Agarwal* et al

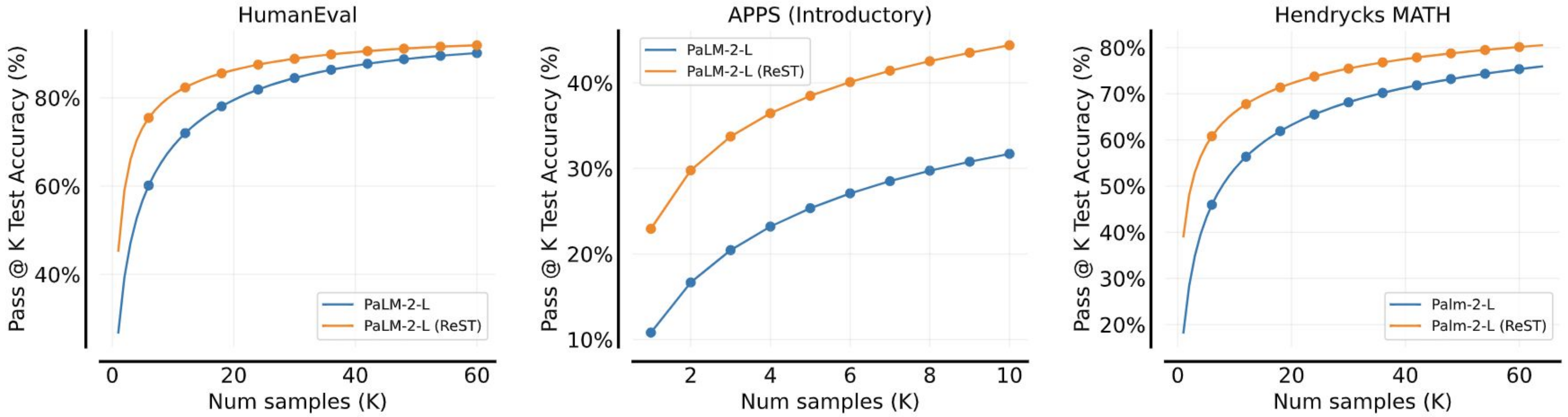
Overfitting is an issue



Beyond Human Data: Scaling Self-Training for Problem-Solving with Language Models. 2023. Singh*, Co-reyes*, Agarwal* et al

Google DeepMind

Pass@K performance improves as well

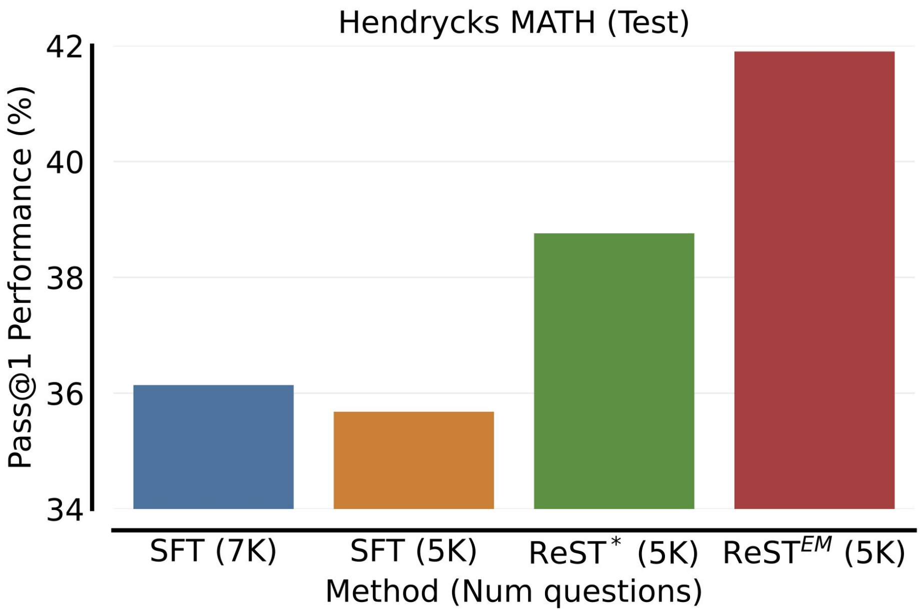


Pass@K measures the probability that at least one of the top k-generated solution for a problem is correct.

Beyond Human Data: Scaling Self-Training for Problem-Solving with Language Models. 2023. Singh*, Co-reyes*, Agarwal* et al

Google DeepMind

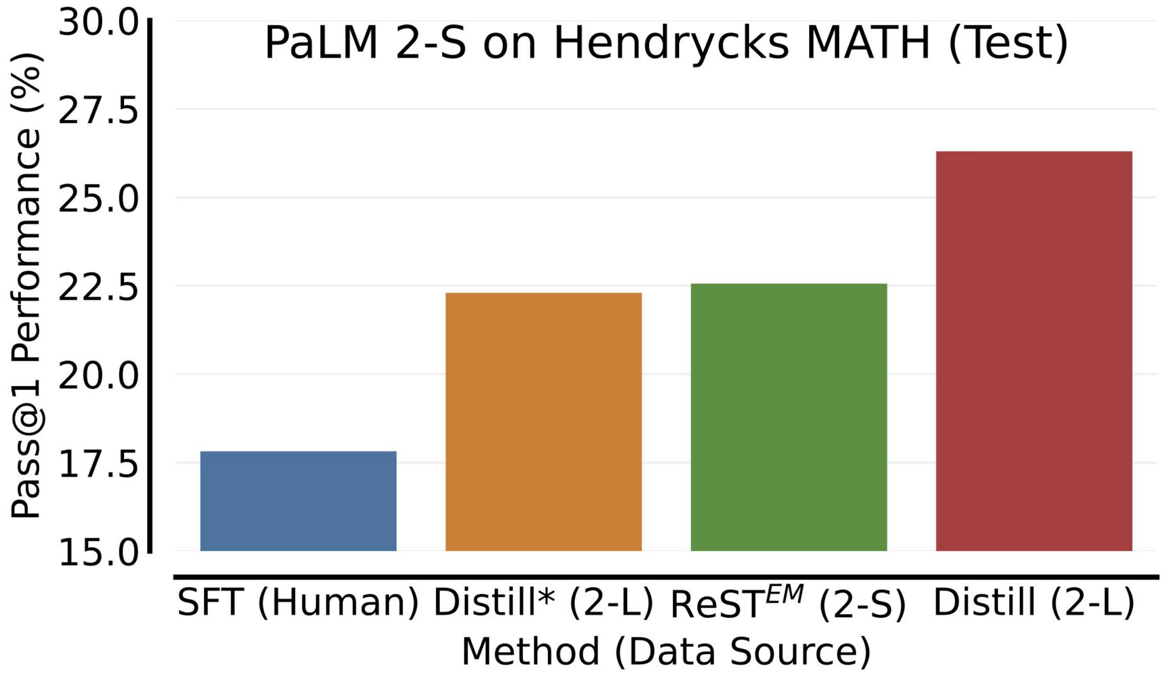
Apples-to-Apples Comparison



Beyond Human Data: Scaling Self-Training for Problem-Solving with Language Models. 2023. Singh*, Co-reyes*, Agarwal* et al

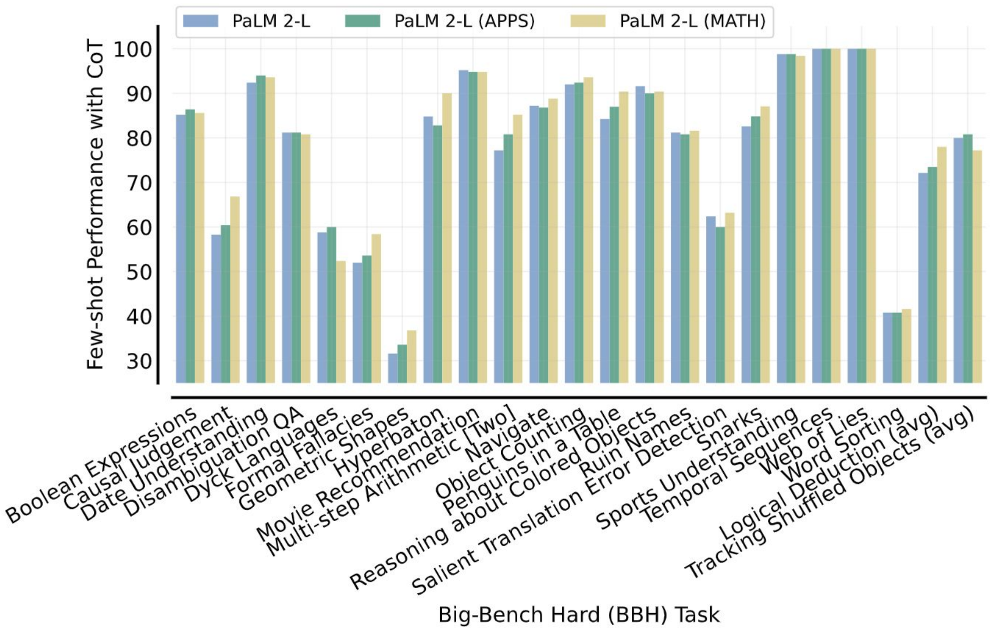
Google DeepMind

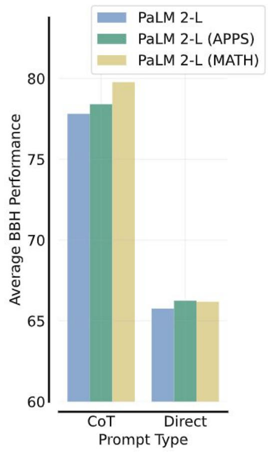
Distilling Palm-2-S using L



Google DeepMind

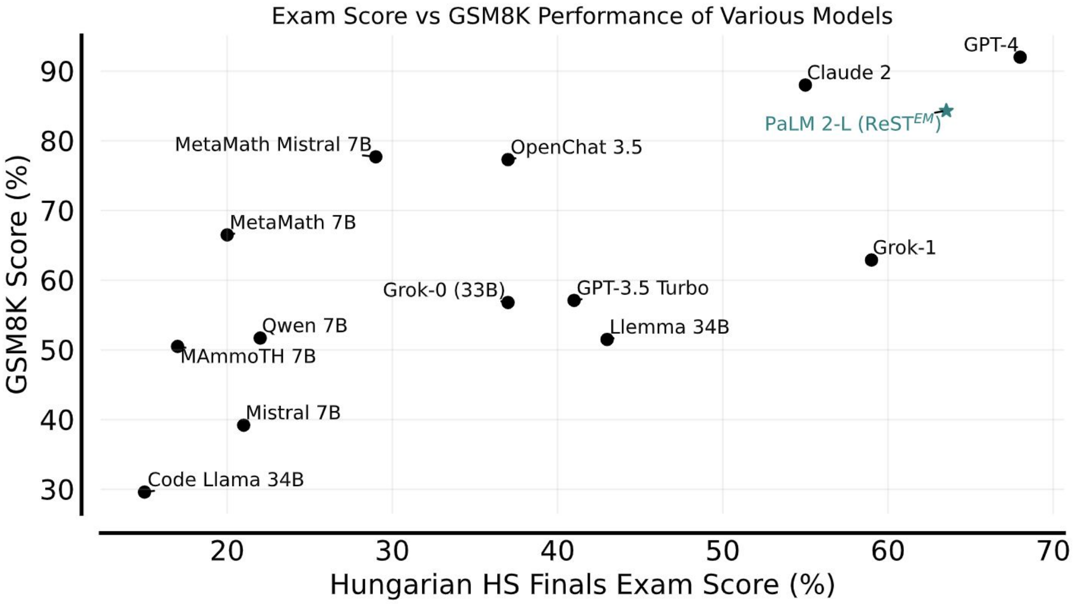
Impact on reasoning tasks





Google DeepMind

Held-Out Eval: 2023 Hungarian HS Exam



Google DeepMind

Things we learned so far:

- Self-generated data improves performance, given reliable reward.
- Self-generated data can often outperform human data – it’s more in-distribution!

Google DeepMind

Revisiting ReST^{EM}

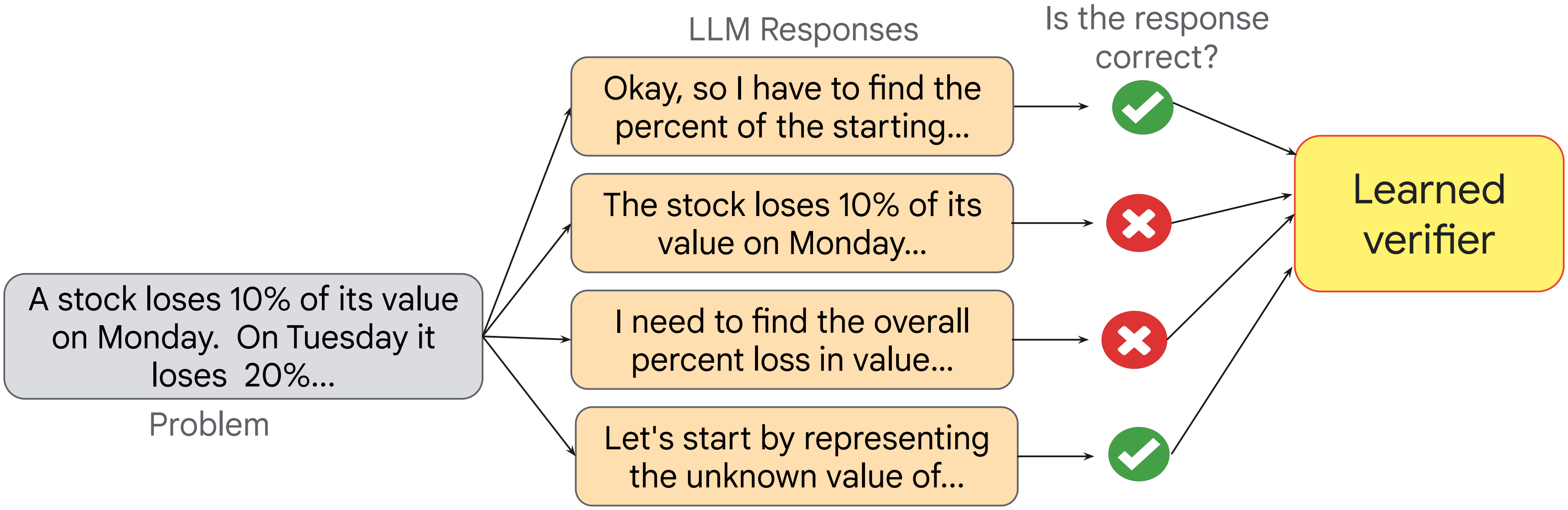
Repeat this process a few times:

1. Generate samples from the model and **filter them using binary feedback.**
2. Fine-tune the model on these samples

Discard the large amounts of incorrect solutions generated during this process, potentially neglecting valuable information!

Google DeepMind

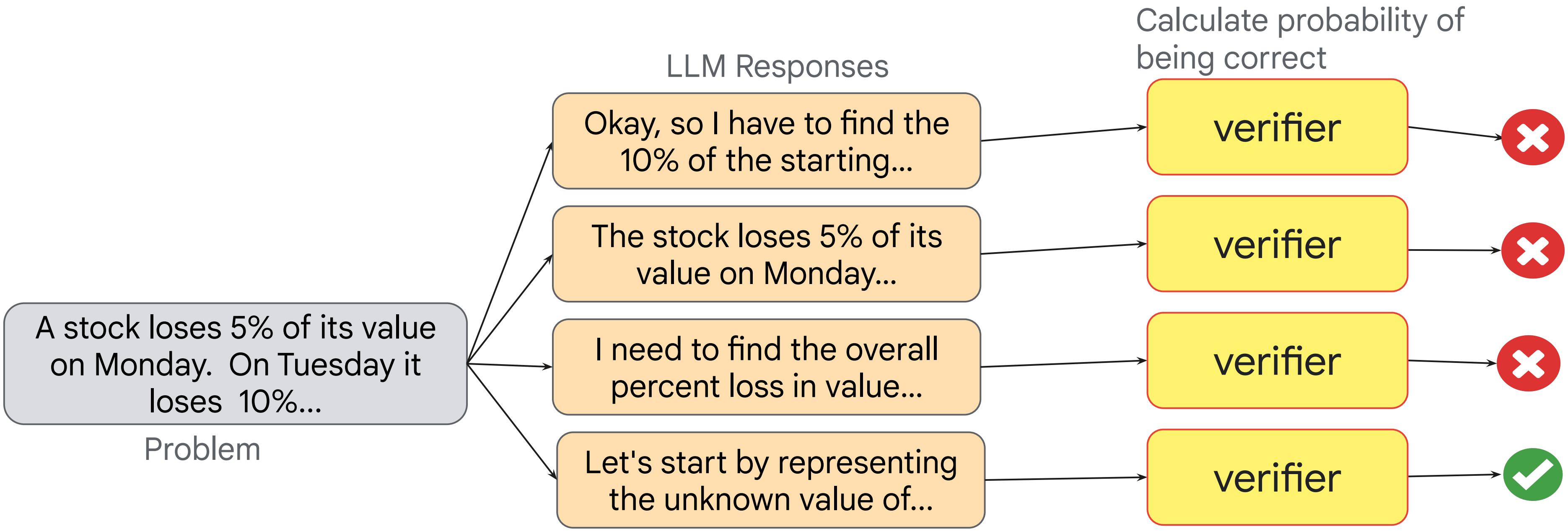
Incorrect solutions for training verifiers



Let's Verify Step by Step. OpenAI, 2023.

Google DeepMind

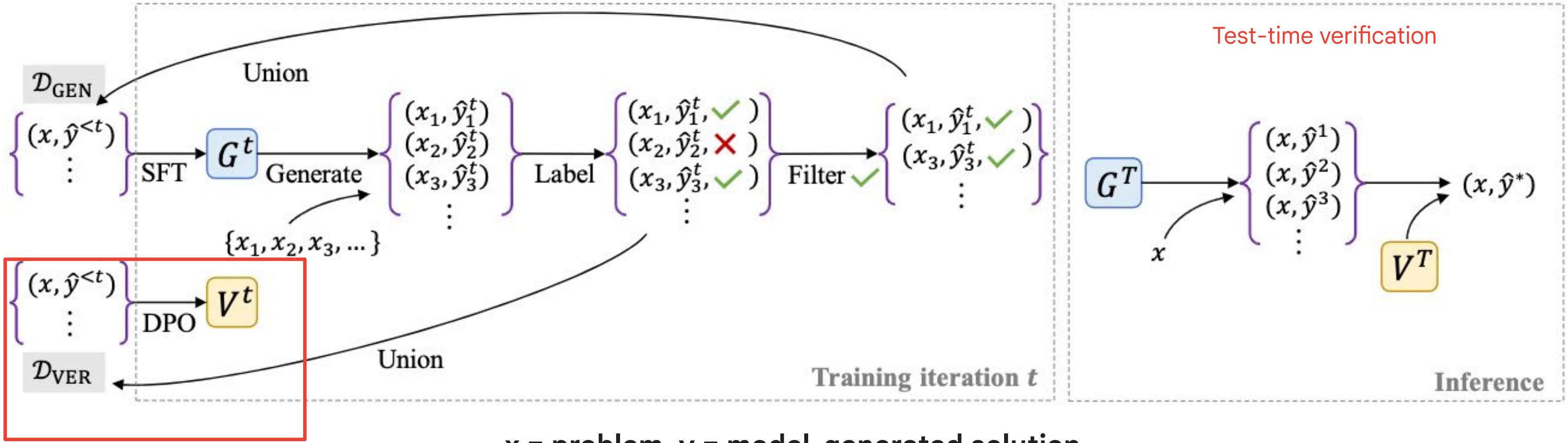
How to use a verifier?



Let's Verify Step by Step. OpenAI, 2023.

Google DeepMind

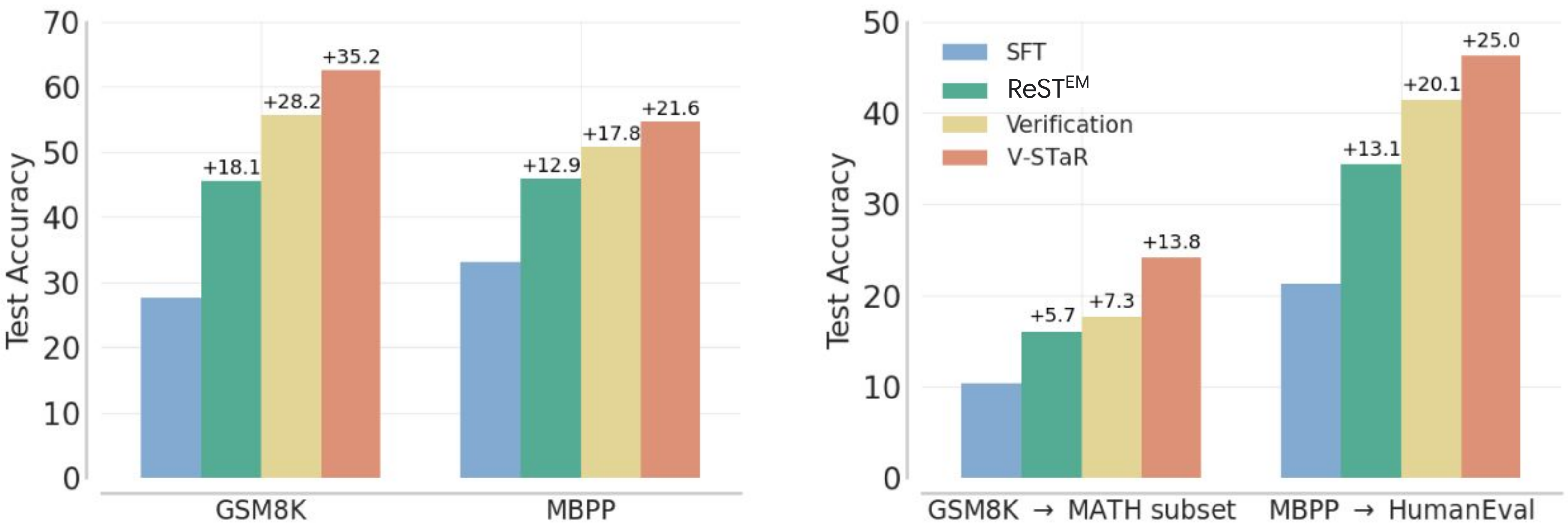
Idea: Augmenting ReST^{EM} with a verifier



x = problem, y = model-generated solution

V-STaR: Training Verifiers for Self-Taught Reasoners. Hosseini et al. 2024

V-STaR: ReST^{EM} + verifier works quite well!



Large gains on math and code reasoning with LLaMA2 7B and 13B models.

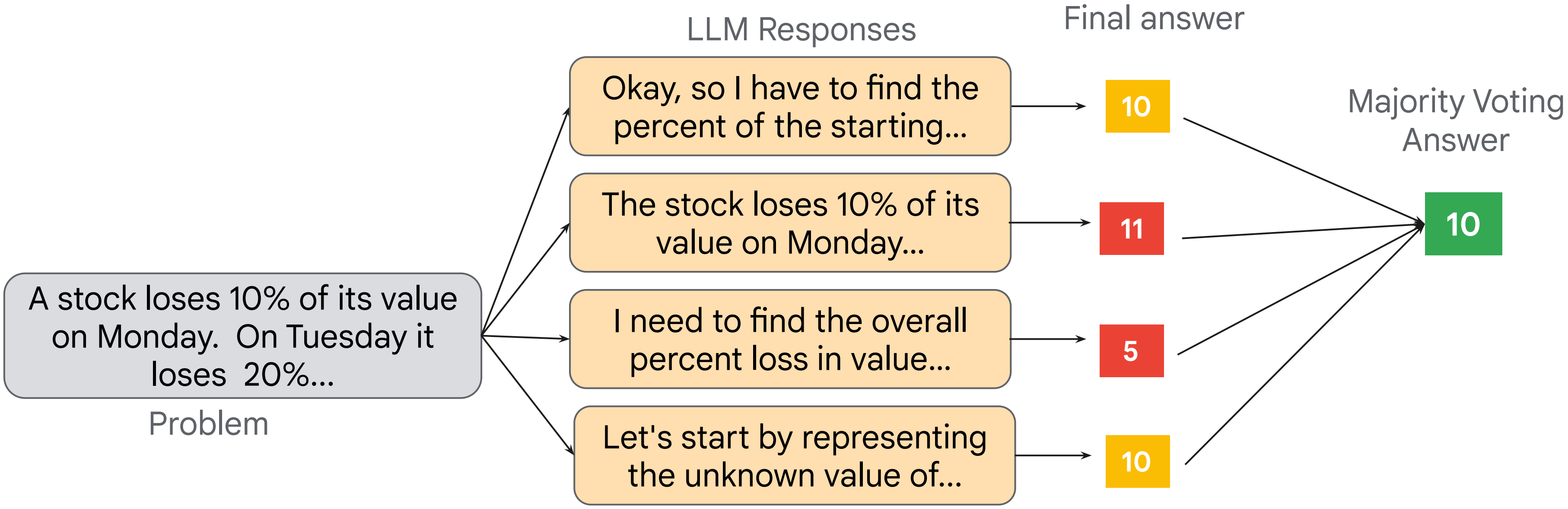
V-STaR: Training Verifiers for Self-Taught Reasoners. Hosseini et al. 2024

V-STaR: Performance across iterations

V-STaR: Training Verifiers for Self-Taught Reasoners. Hosseini et al. 2024

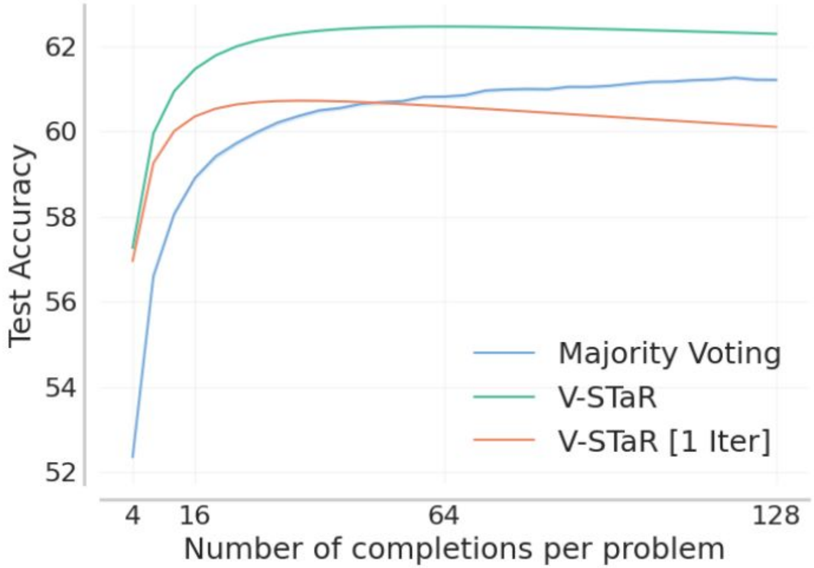
Google DeepMind

A Strong Baseline: Majority Voting



Let's Verify Step by Step. OpenAI, 2023.

V-STaR Outperforms Majority Voting.



V-STaR: Training Verifiers for Self-Taught Reasoners. Hosseini et al. 2024

Google DeepMind

Things we learned so far:

- Self-generated data improves performance, given reliable reward.
- Self-generated data can often outperform human data – it's more in-distribution!
- We can train a verifier, using both correct and incorrect solutions.

Revisiting ReST^{EM} (yet again!)

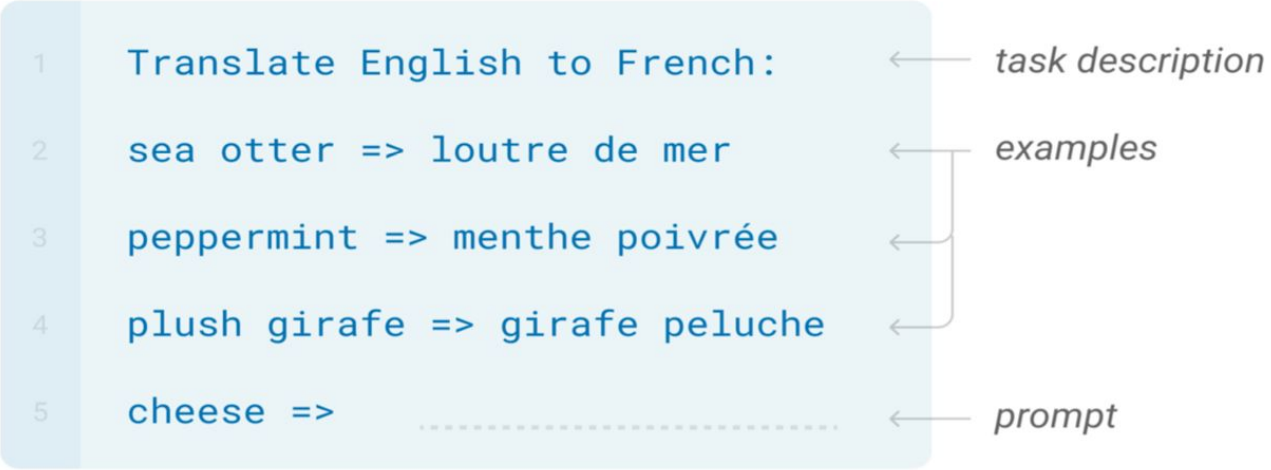
Repeat this process a few times:

1. Generate samples from the model and filter them using binary feedback.
2. Fine-tune the model on these samples

Is fine-tuning necessary? Wait, what?

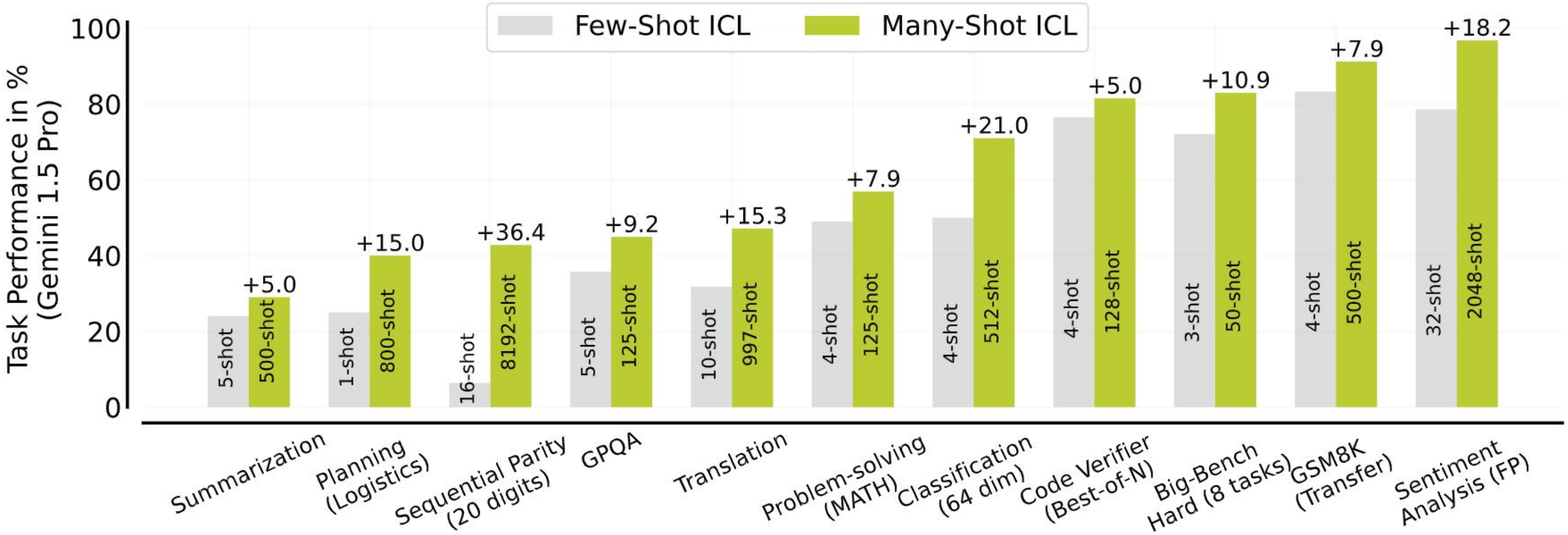
Background: In-Context Learning

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



Google DeepMind

Many-Shot In-Context Learning



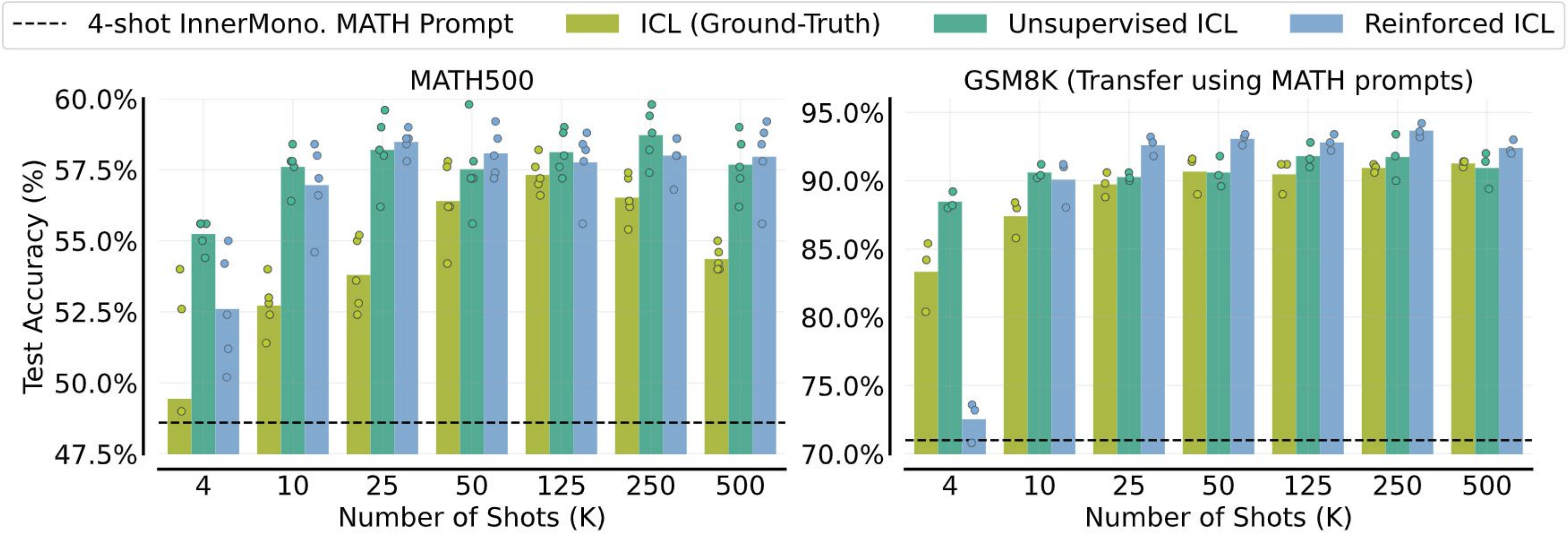
Many-Shot In-Context Learning. Agarwal et al, 2024

Google DeepMind

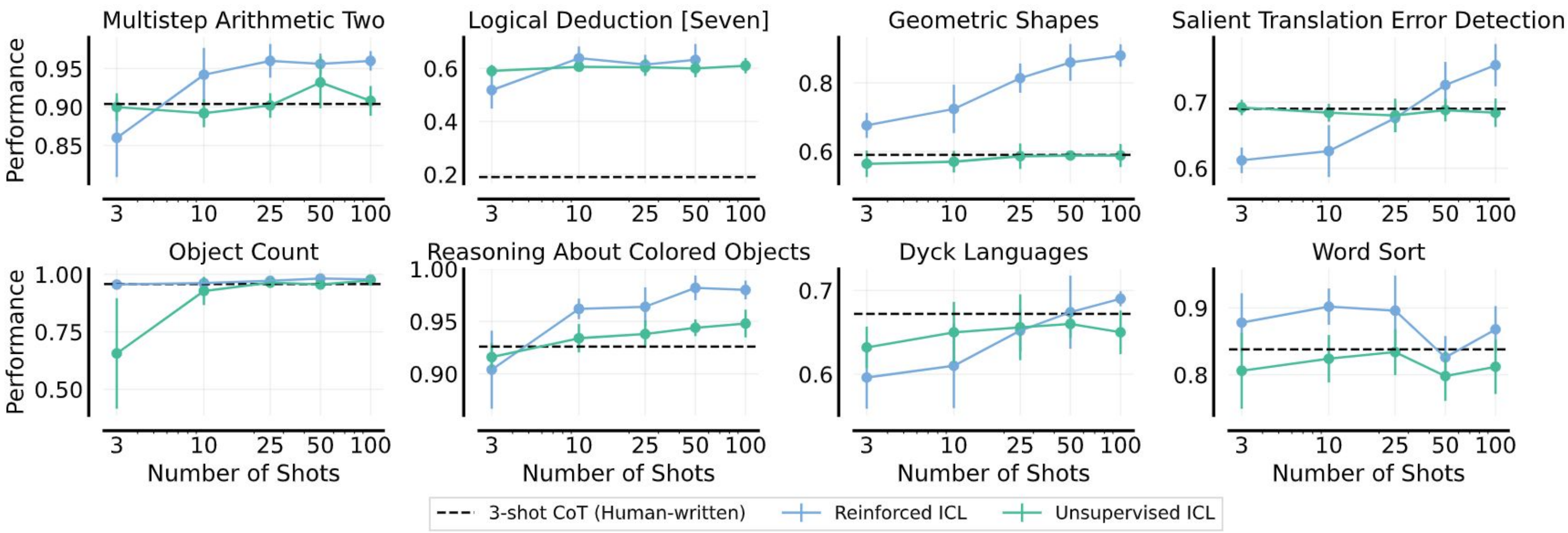
In-Context ReST^{EM} : Reinforced ICL

1. Generate samples from the model and filter them using binary feedback.
2. Put these (problem, solution) pairs in-context for the model.

Reinforced ICL on MATH

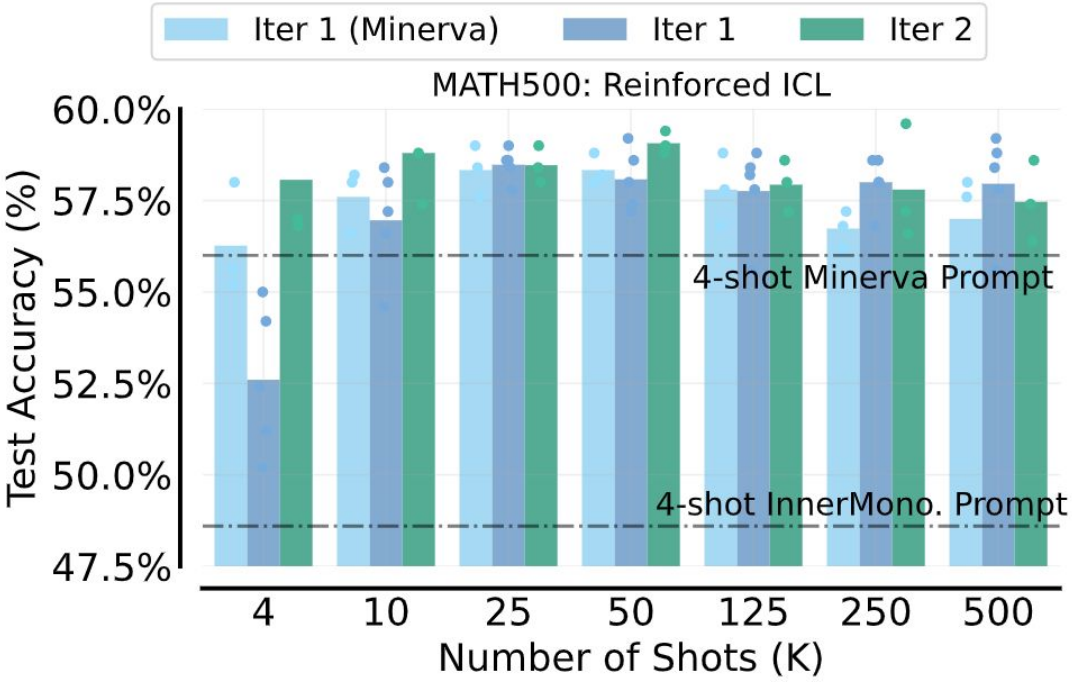


Reinforced ICL on Big-Bench Hard



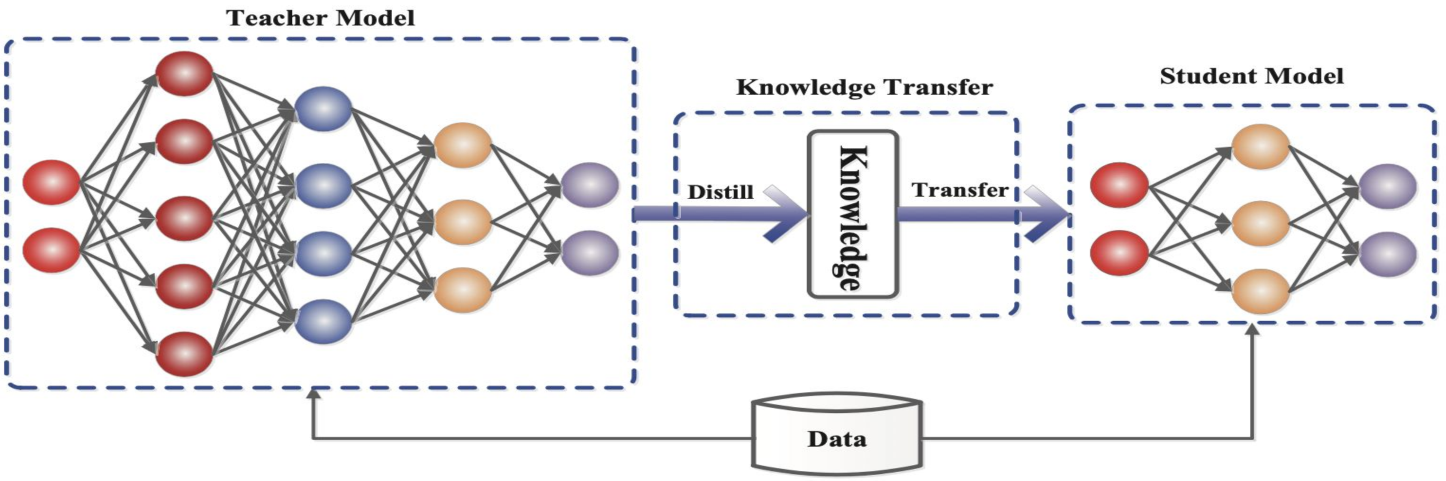
Google DeepMind

Reinforced ICL: Iteration 2



Google DeepMind

On-policy Distillation of LLMs: Learning from Self-Generated Mistakes

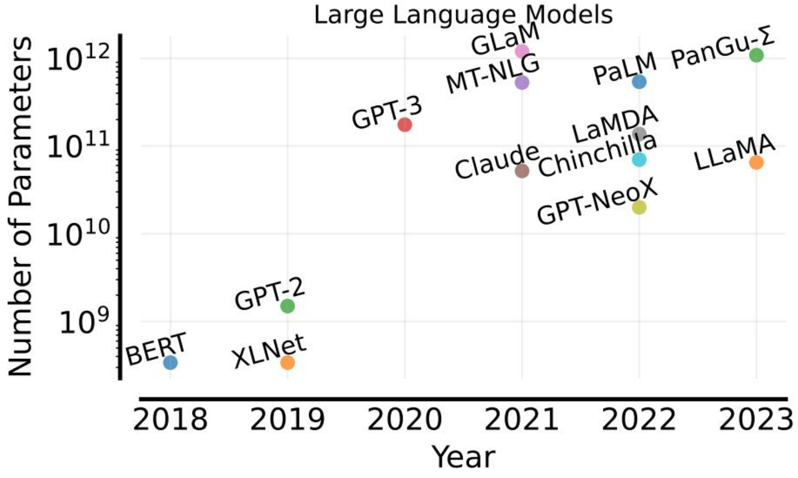


The generic framework of teacher-student knowledge distillation training. (Image source: Gou et al. 2020)

Google DeepMind

Why Distill: Aren’t bigger LLMs better?

- Deployment of “large” models limited by either their **inference cost** or **memory footprint**.
    - You can’t put PaLM 540B on your smartphone.
    - You don’t want to typically wait several minutes for an ML model to generate an output.



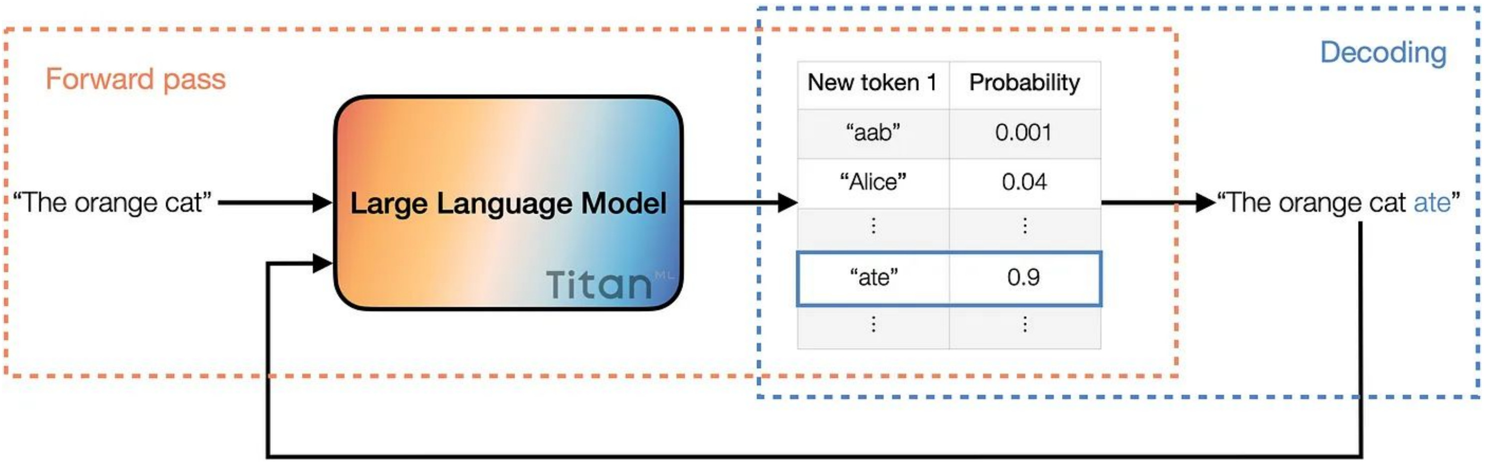
What is Model Compression?

The main idea is to simplify the model without diminishing accuracy. A simplified model means reduced in size and/or latency from the original.

- ➢ Size reduction can be achieved by reducing the model parameters and thus using less RAM.
- ➢ Latency reduction can be achieved by decreasing the time it takes for the model to make a prediction, and thus lowering energy consumption at runtime (and carbon footprint).

Google DeepMind

Language models generate text auto-regressively!

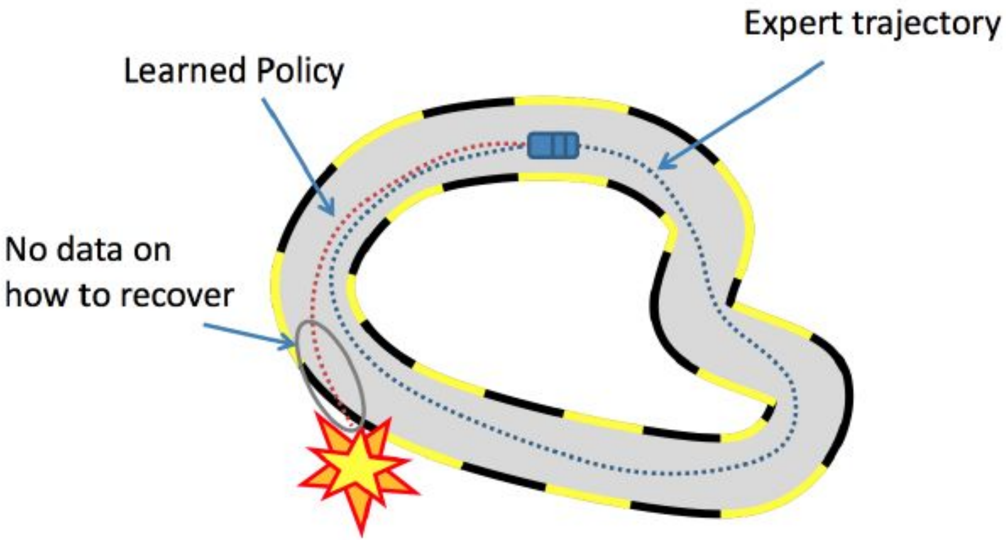


Language models (LMs) generate outputs sequentially token by token – later output tokens depend on past tokens!

Google DeepMind

Distribution Mismatch (Exposure Bias)

Existing methods typically train on a fixed dataset of output sequences. This results in a mismatch with the sequences generated by the student auto-regressively during inference.



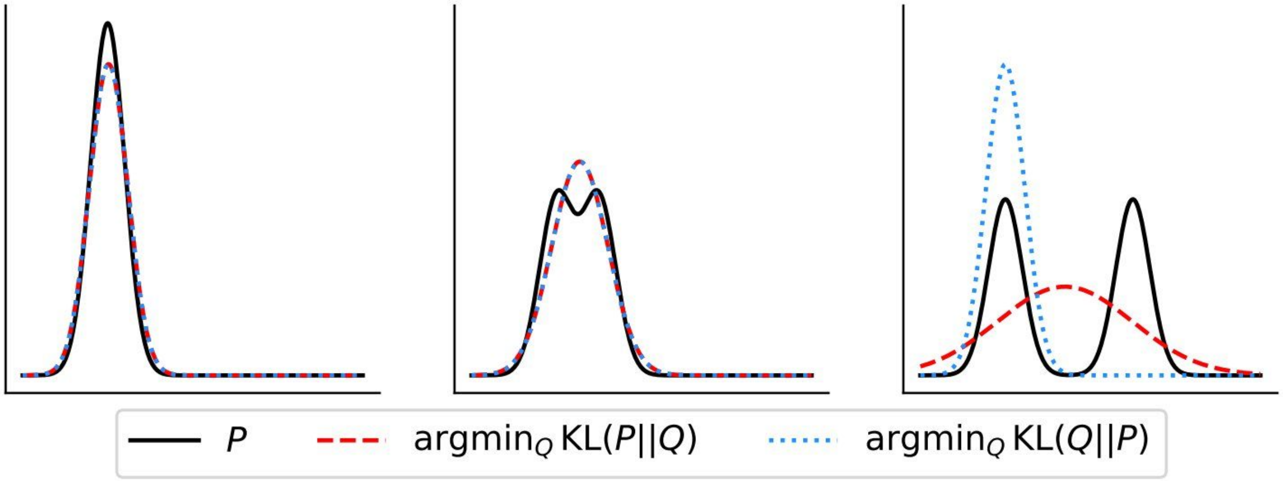
Well-known in the Imitation learning community.

On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes. ICLR 2024.

Google DeepMind

Model Underspecification

If student is often not expressive enough to fit the teacher's distribution, standard KD objective can lead to unnatural student-generated samples. MLE = KL(P||Q).



Google DeepMind

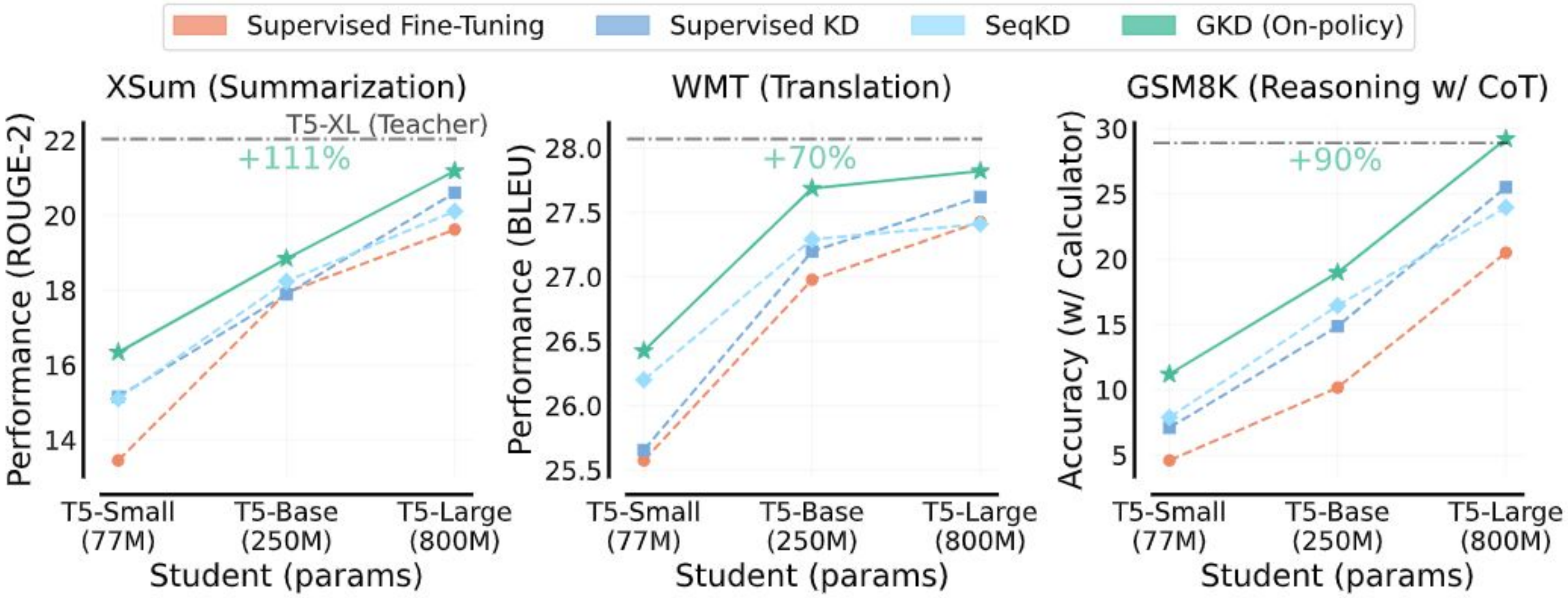
Generalized Knowledge Distillation (GKD)

- ➢ Sample self-generated output sequences from the student model.
- ➢ Run inference on the teacher to get logits on these sequences – (what the teacher would do in this situation)
- ➢ Minimize the mismatch between the student and teacher logits for each token.

On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes. ICLR 2024.

Google DeepMind

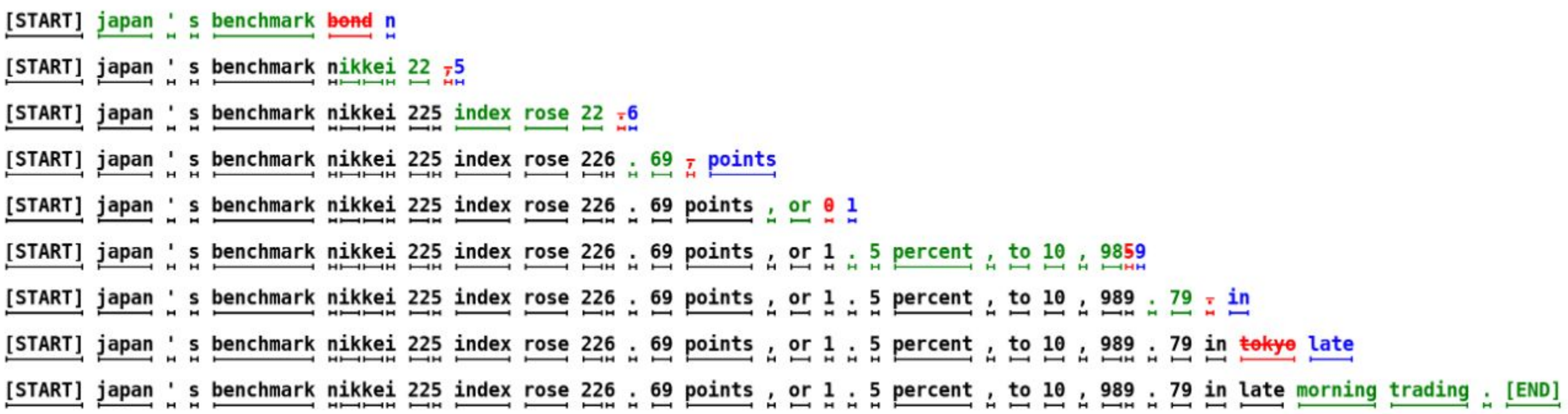
Task-specific GKD Results



On-Policy Distillation of Language Models: Learning from Self-Generated Mistakes. ICLR 2024.

Google DeepMind

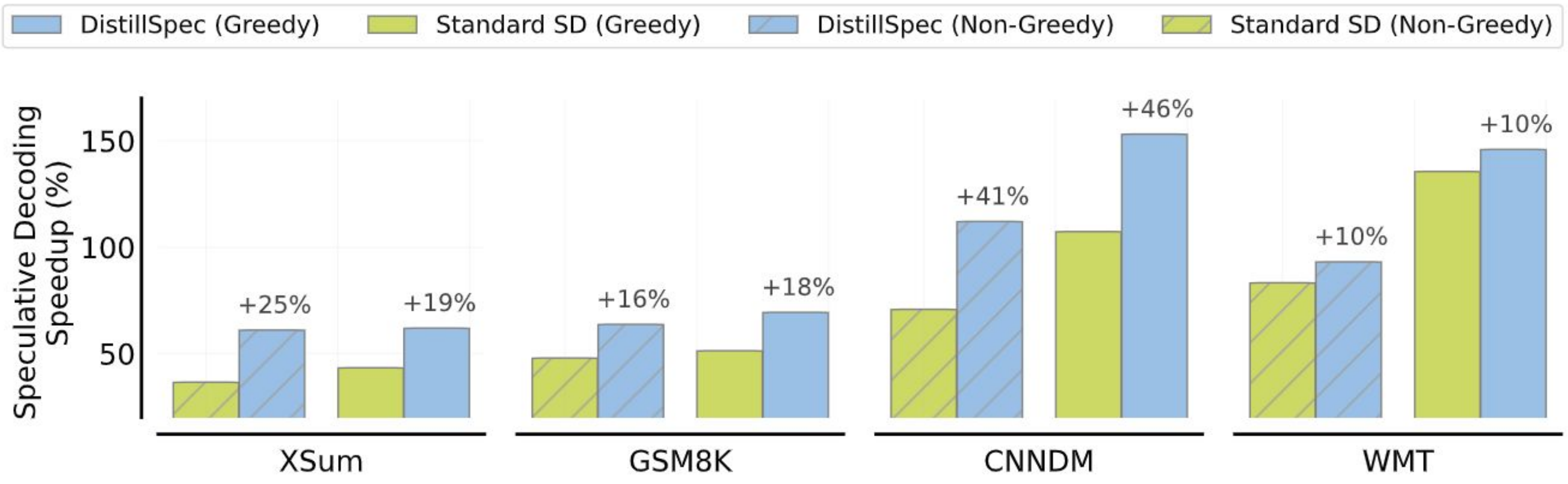
DistillSpec: KD for Speculative Decoding



Fast Inference from Transformers via Speculative Decoding. ICML 2023.

Google DeepMind

DistillSpec: KD for Speculative Decoding



DistillSpec: Improving Speculative Decoding via Knowledge Distillation. ICLR 2024.

Google DeepMind

Thank you!
Questions?