Continually self-improving AI

Zitong Yang
Stanford Statistics

Dissertation defense, March 3rd, 2026

A continually self-improving AI is one that, once created, can autonomously and continually improve itself better than its human creators can improve it.

Definition

We consider AI systems satisfying the following two assumptions.

- **(A1) Parametric:** The AI system is based on one or more neural networks, so that its knowledge is encoded in a well-defined set of parametric weights
- **(A2) Pretrained:** The AI system went through a resource-intensive pretraining phase:

```
ai_system = learning_algorithm(training_signal)
```

  where `training_signal` encompasses much of human knowledge.

Definition

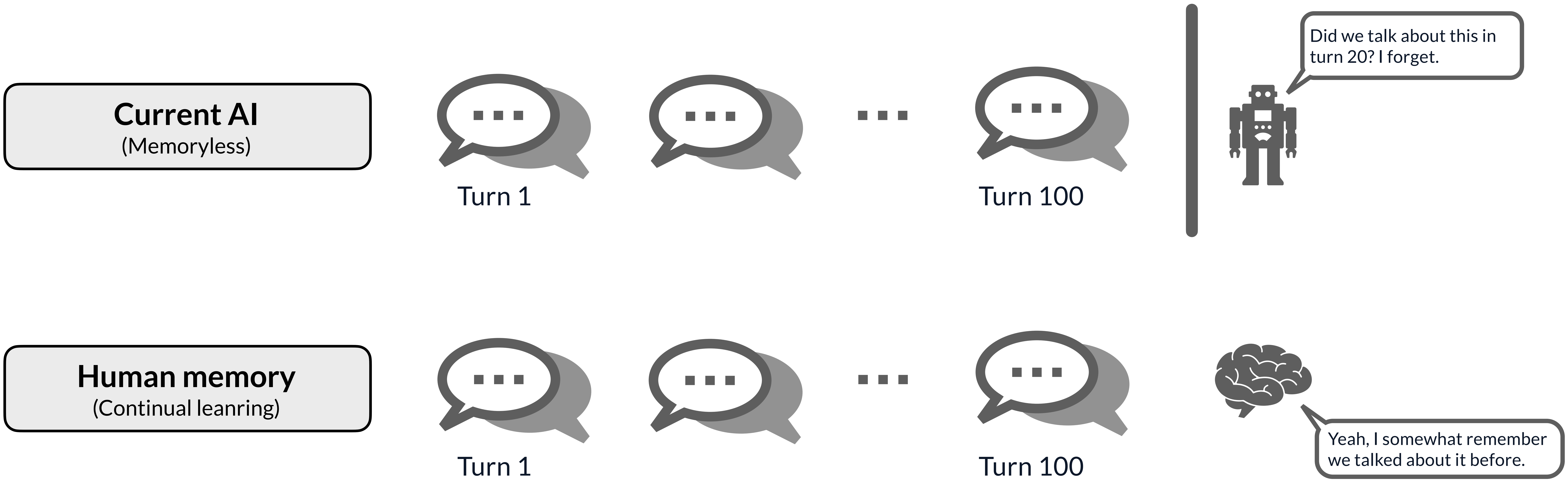
Under parametric and pretrained assumption, a *continually self-improving AI system* satisfies

- **(P1)** After the initial pretraining phase, the system continues to acquire new knowledge into its parametric weights without catastrophically forgetting existing capabilities.
- **(P2)** The system generates its own `training_signal`, and learning from these self-generated signals yields improvement beyond what human-generated signals provide.
- **(P3)** The system can autonomously design what `learning_algorithm` to use to learn from its training signals.

Why continually self-imporving AI?

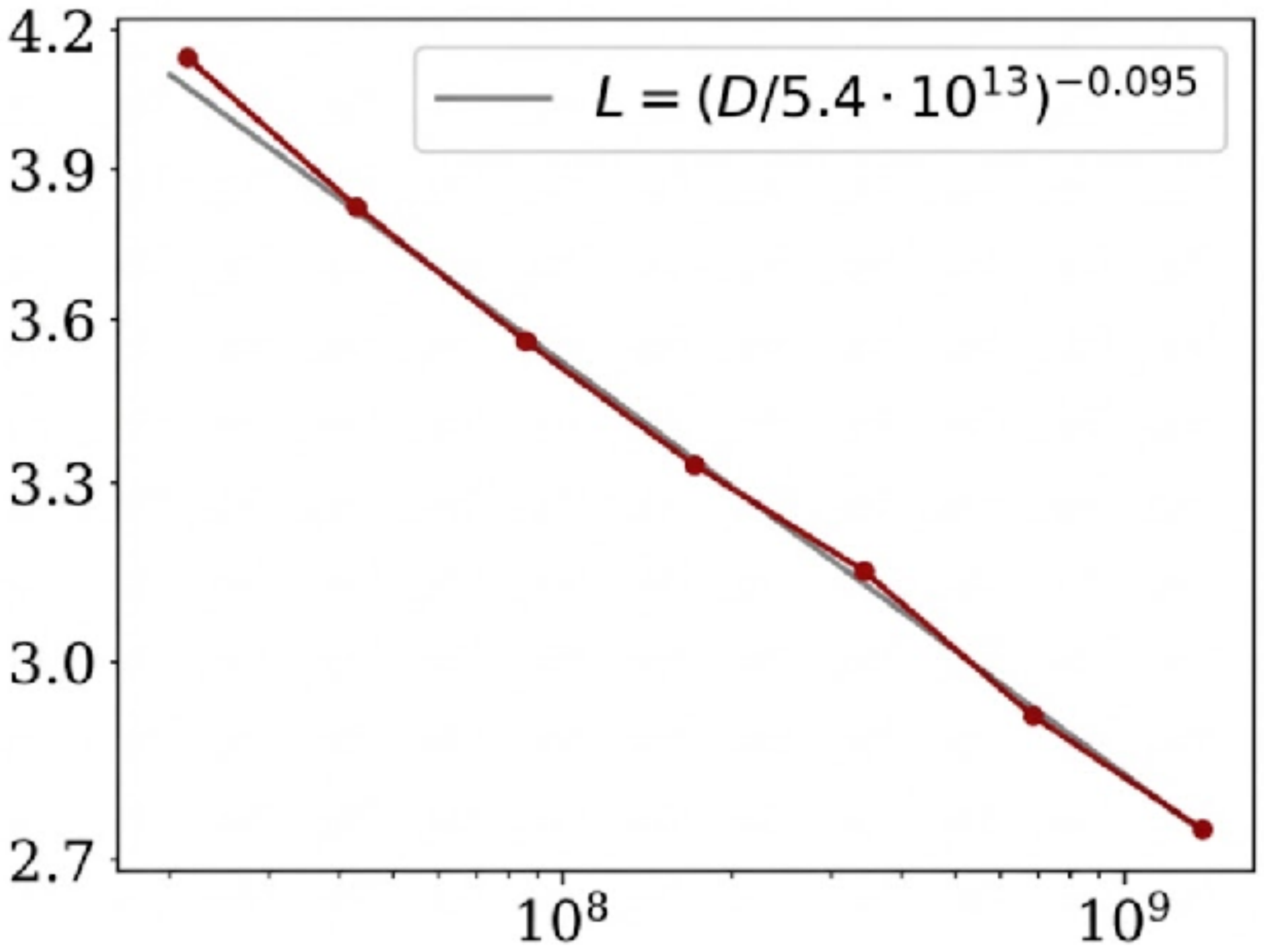
Static weights after *human* creation

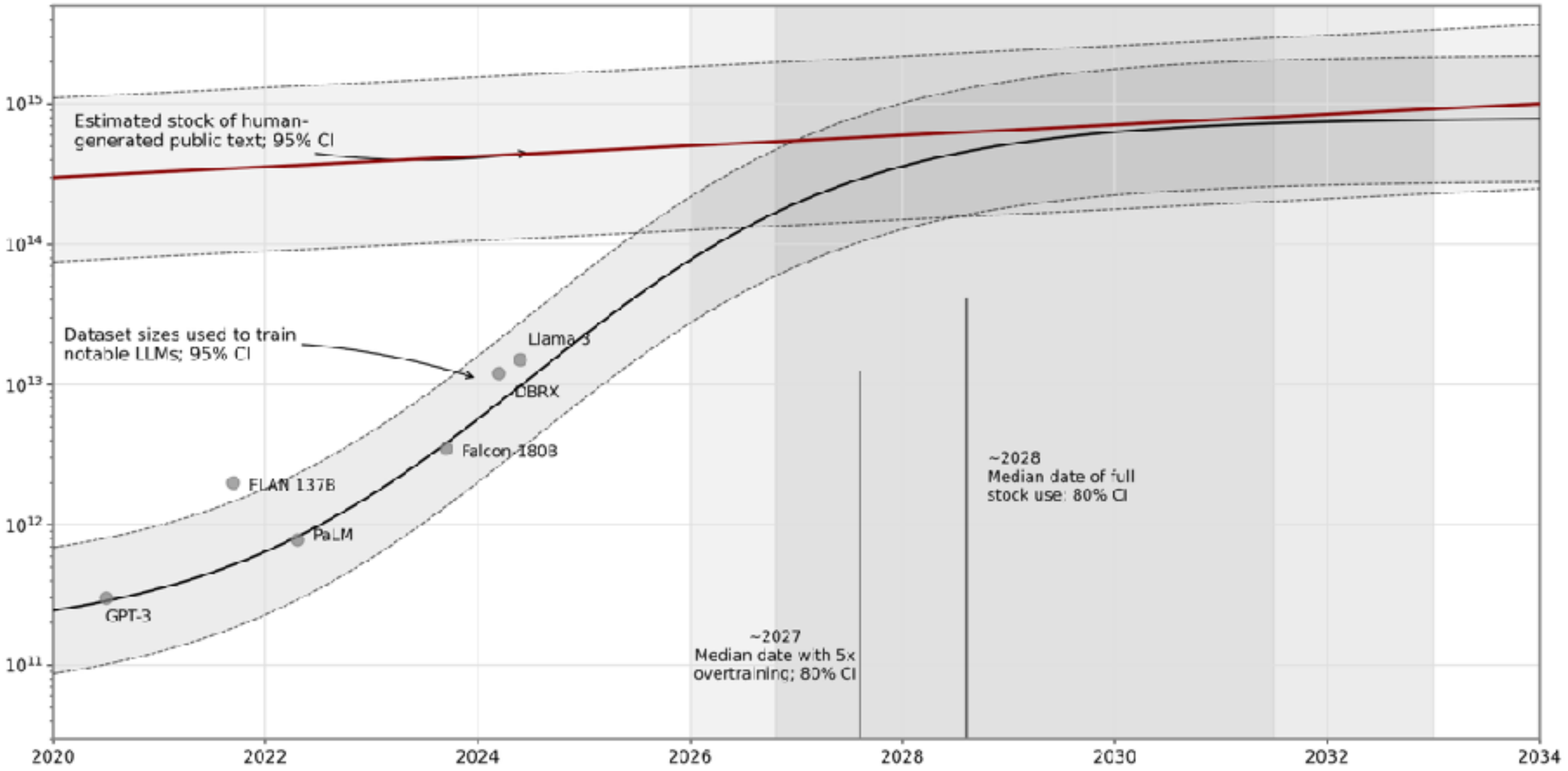
context compactification



Why continually self-imporving AI?

Scaling under finite *human* data





Why continually self-imporving AI?

Limited by algorithms *human* can find

All posible algorithms

Human discovery

Generating ideas

- Maximum likelihood estimation
- Gradient descent
- Transformers

Experimentation

- `print(f'L={loss}, t={steps}')`
- `RuntimeError: CUDA out of memory.`
- $$x^T - x^1 = \sum_{t=1}^{T} -\gamma \nabla f(x^t)$$

Research artifact

- We regret to … due to high volume …
- ⚙️⚙️⚙️ Excited to share …
- "I want to make this repository public"

Outline

- Continual knowledge acquisition
- Self-improving pretraining capability
- Towards AI-designed AI

Collaborators


Neil Band*

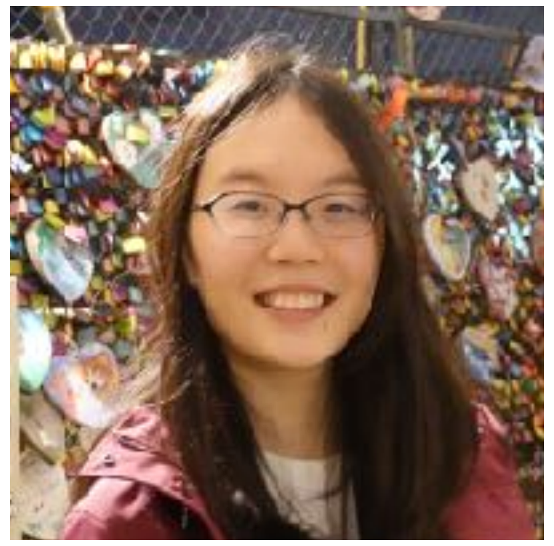

Emmanuel Candès

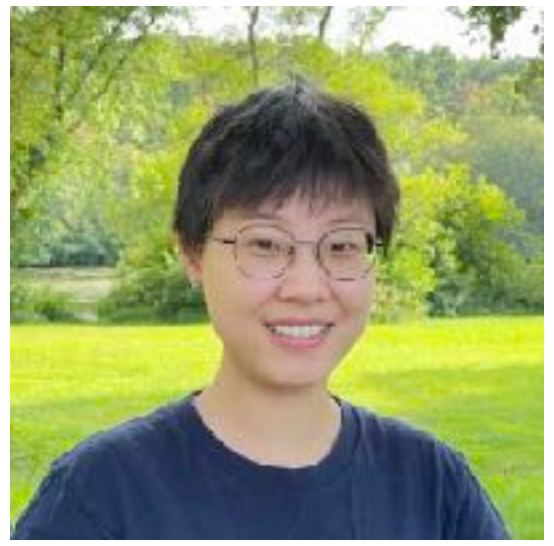

Yejin Choi


Li Fei-Fei

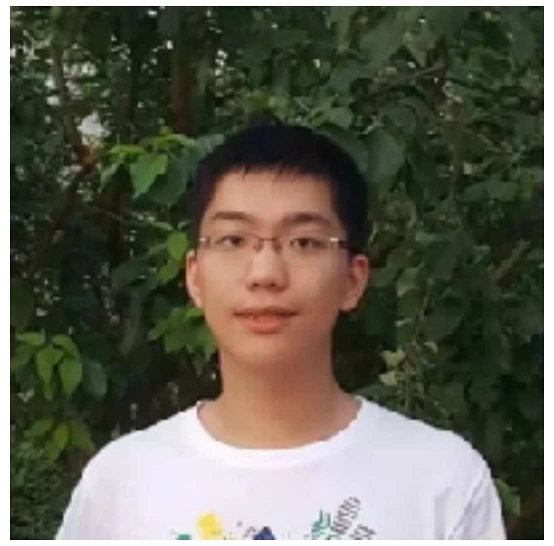

Hannaneh Hajishirzi


Tatsunori Hashimoto

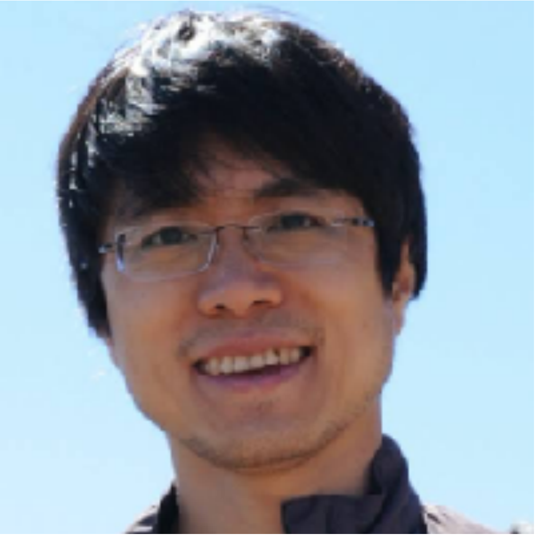

Xiang Li*


Shuangping Li


Percy Liang


Hong Liu


Niklas Muennighoff*


Ruoming Pang


Weijia Shi*

Chenglei Si*


Chong Wang


Diyi Yang


Luke Zettlemoyer


Aonan Zhang*

***Equal contribution**

Synthetic continued pretraining

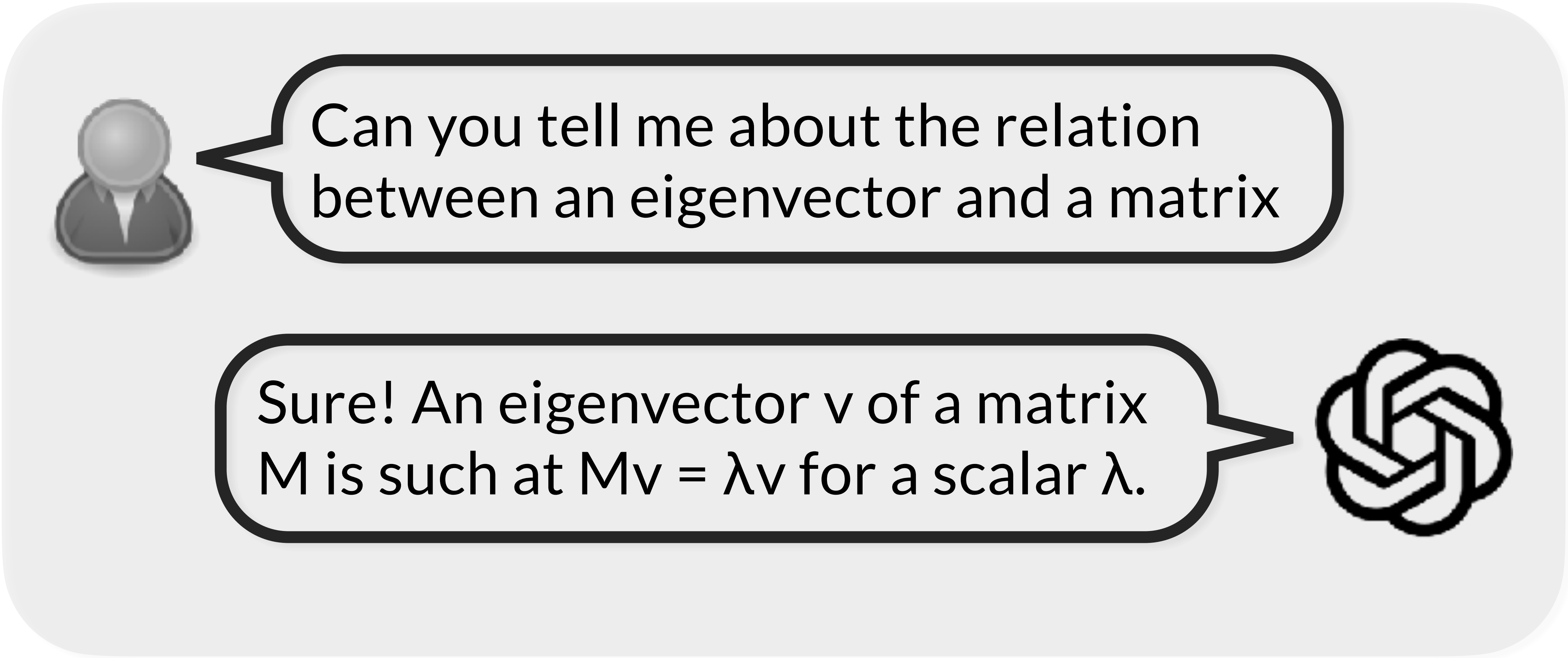
Goal: teach model the knowledge from a niche domain consisted of a few “source documents"

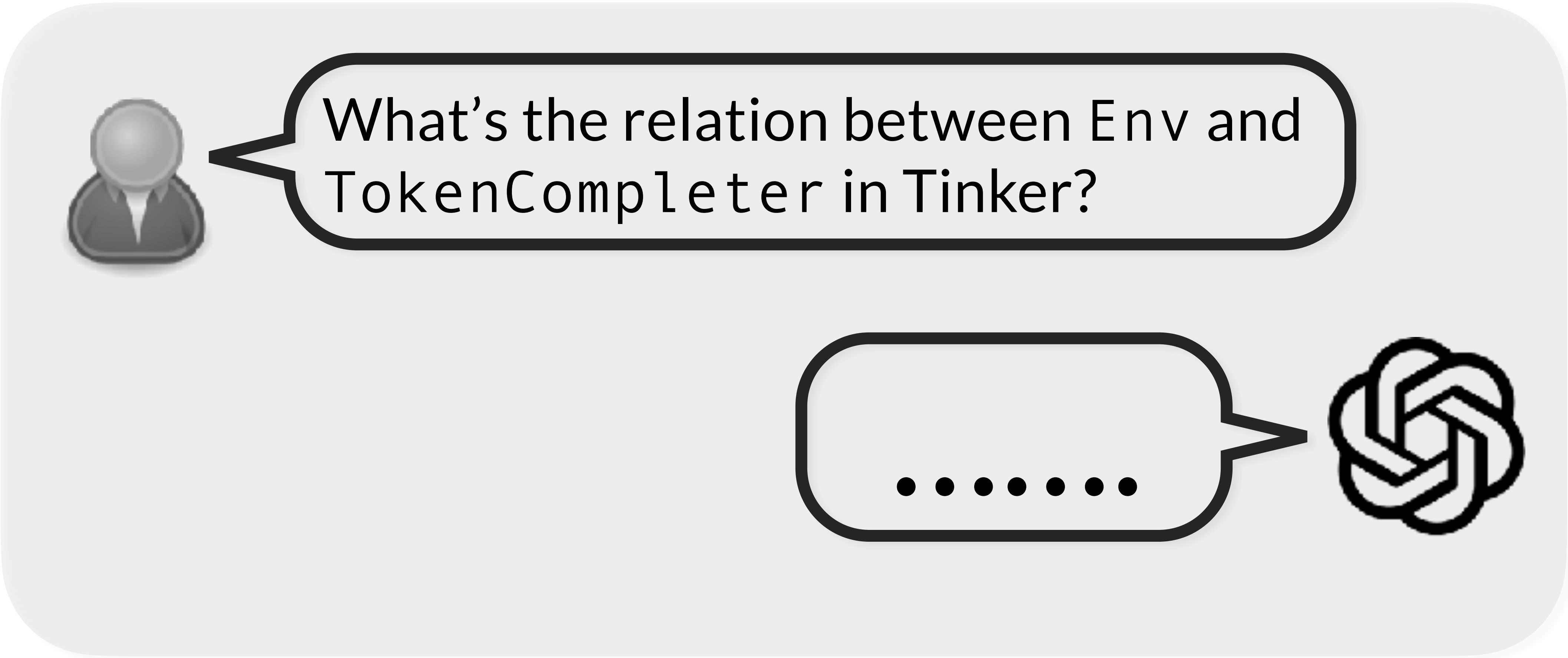
Step 1: Generate synthetic text based on the source documents

Step 2: Continually pretrain (finetune) the model on generated text

Knowledge can be sparse without synthetic data

- Model knows a bit about linear algebra, but not much about niche domains, e.g. new codebase





- Model acquires linear algebra knowledge from a wide range of internet data

Many textbooks, lecture notes about linear algebra.

Online discussion of linear algebra exercise

GitHub implementation of SVD

Synthetic continued pretraining

Goal: teach model the knowledge from a niche domain consisted of a few “source documents"

Step 1: Generate synthetic text based on the source documents

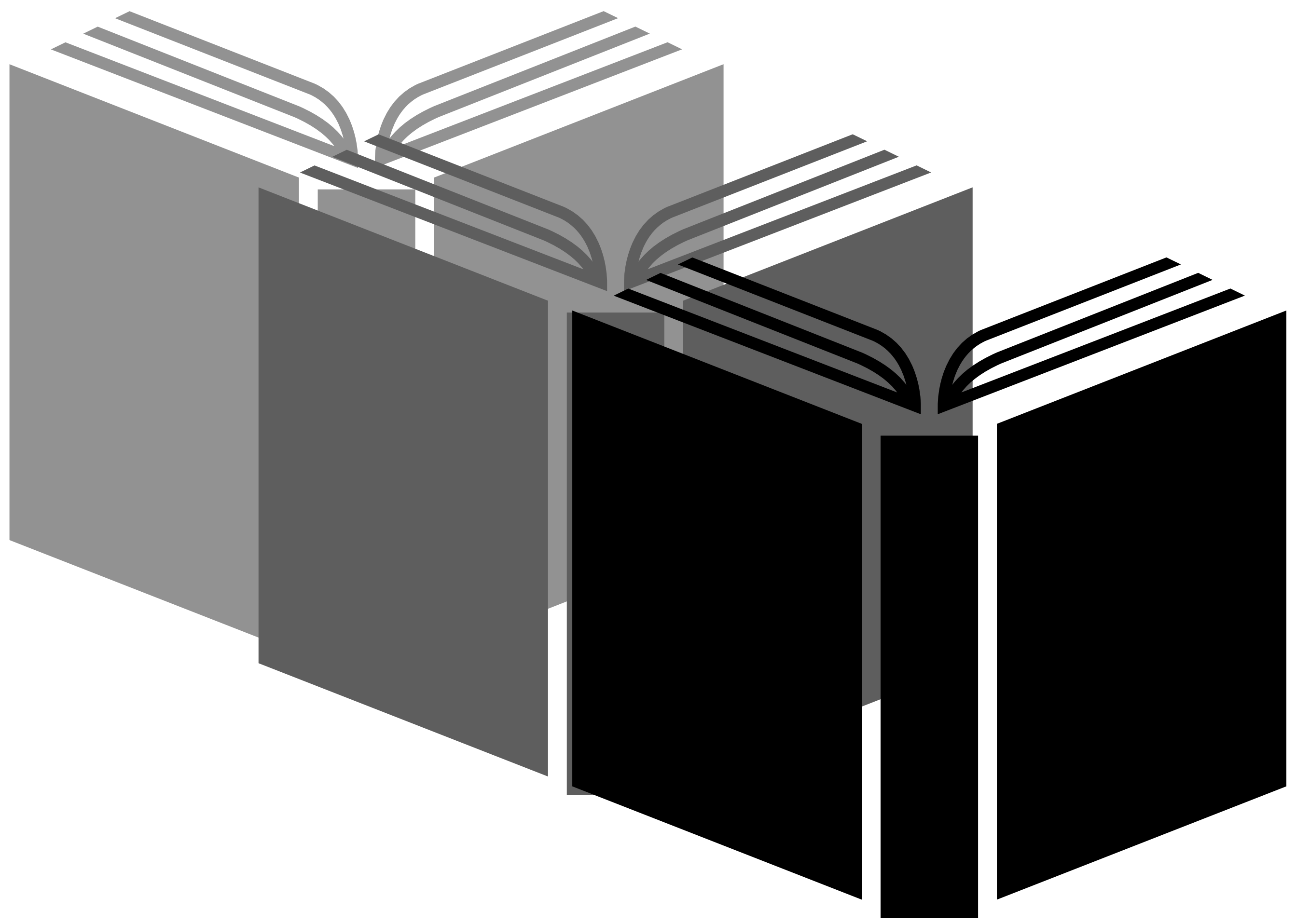
Step 2: Continually pretrain (finetune) the model on generated text

Experiment setup

- A collection of niche (not something model already know) source documents
- A task that tests a model’s knowledge about the source documents

A dataset and a benchmark

QuALITY Books



- Project Gutenberg fictions (mainly science fiction)
- Slate magazine articles
- The Long and Short, Freesouls, etc

QuALITY [Pang+ '21]

- 265 *specialized* books (~1.8M tokens) model doesn't already know
- High-quality multiple choice Q&As
- Want model to answer without book in-context (closed book)

Synthetic continued pretraining

Goal: teach model the knowledge from a niche domain consisted of a few “source documents"

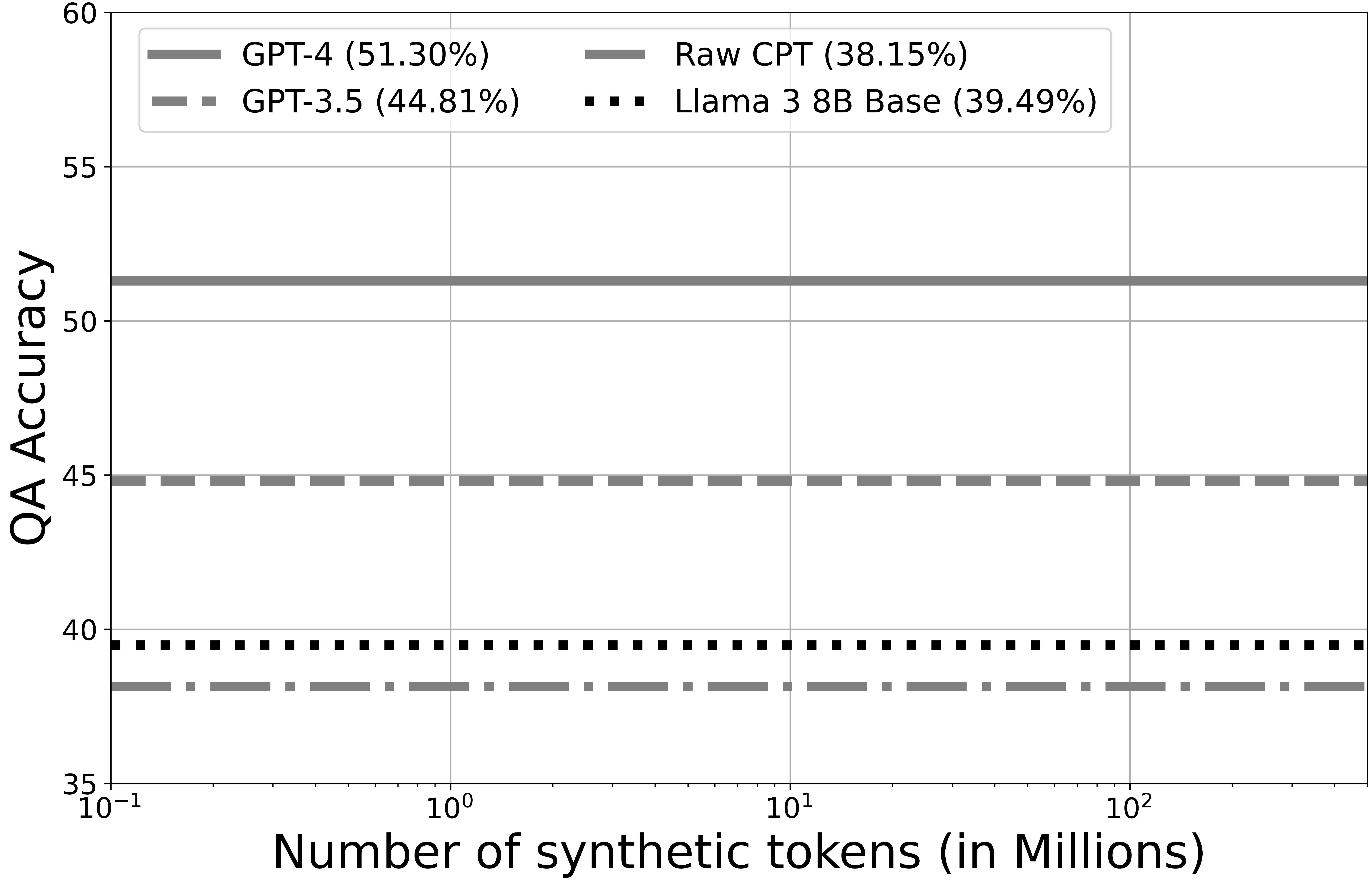
Step 1: Generate synthetic text based on the source documents

Step 2: Continually pretrain (finetune) the model on generated text

Experiment setup

- A collection of niche (not something model already know) source documents **—> QuALITY books**
- A task that tests a model’s knowledge about the source documents **—> Closed book QA**

Current model has poor performance



Synthetic continued pretraining

Goal: teach model the knowledge from a niche domain consisted of a few “source documents"

Step 1: Generate synthetic text based on the source documents

Step 2: Continually pretrain (finetune) the model on generated text

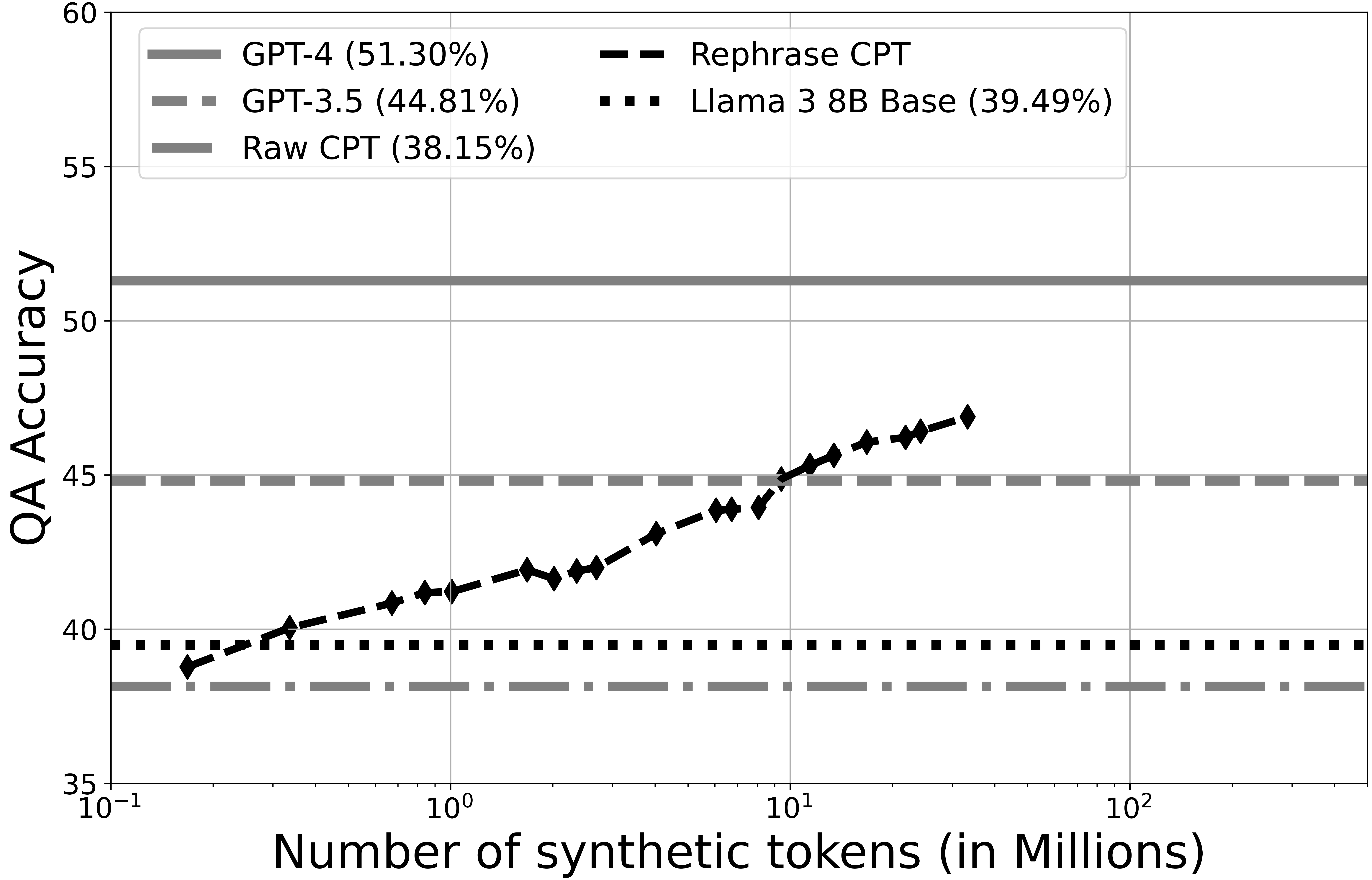
Experiment setup

- A collection of niche (not something model already know) source documents **—> QuALITY books**
- A task that tests a model’s knowledge about the source documents **—> Closed book QA**

How to generate synthetic data?

- Baseline: simply rephrase the document Pratyush et al. 2024

First attempt: simply rephrase



Synthetic continued pretraining

Goal: teach model the knowledge from a niche domain consisted of a few "source documents"

Step 1: Generate synthetic text based on the source documents

Step 2: Continually pretrain (finetune) the model on generated text

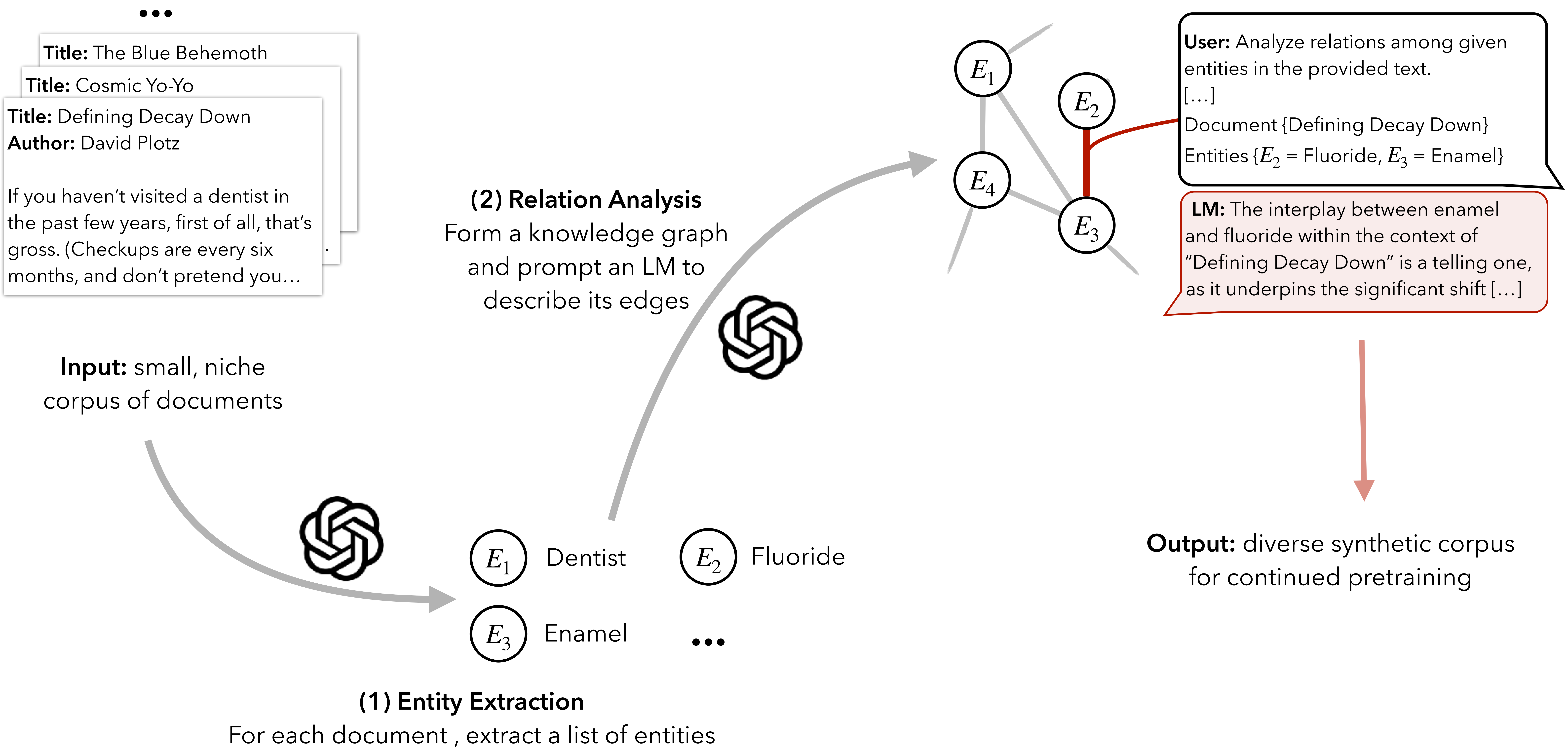
Experiment setup

- A collection of niche (not something model already know) source documents —> **QuALITY books**
- A task that tests a model's knowledge about the source documents —> **Closed book QA**

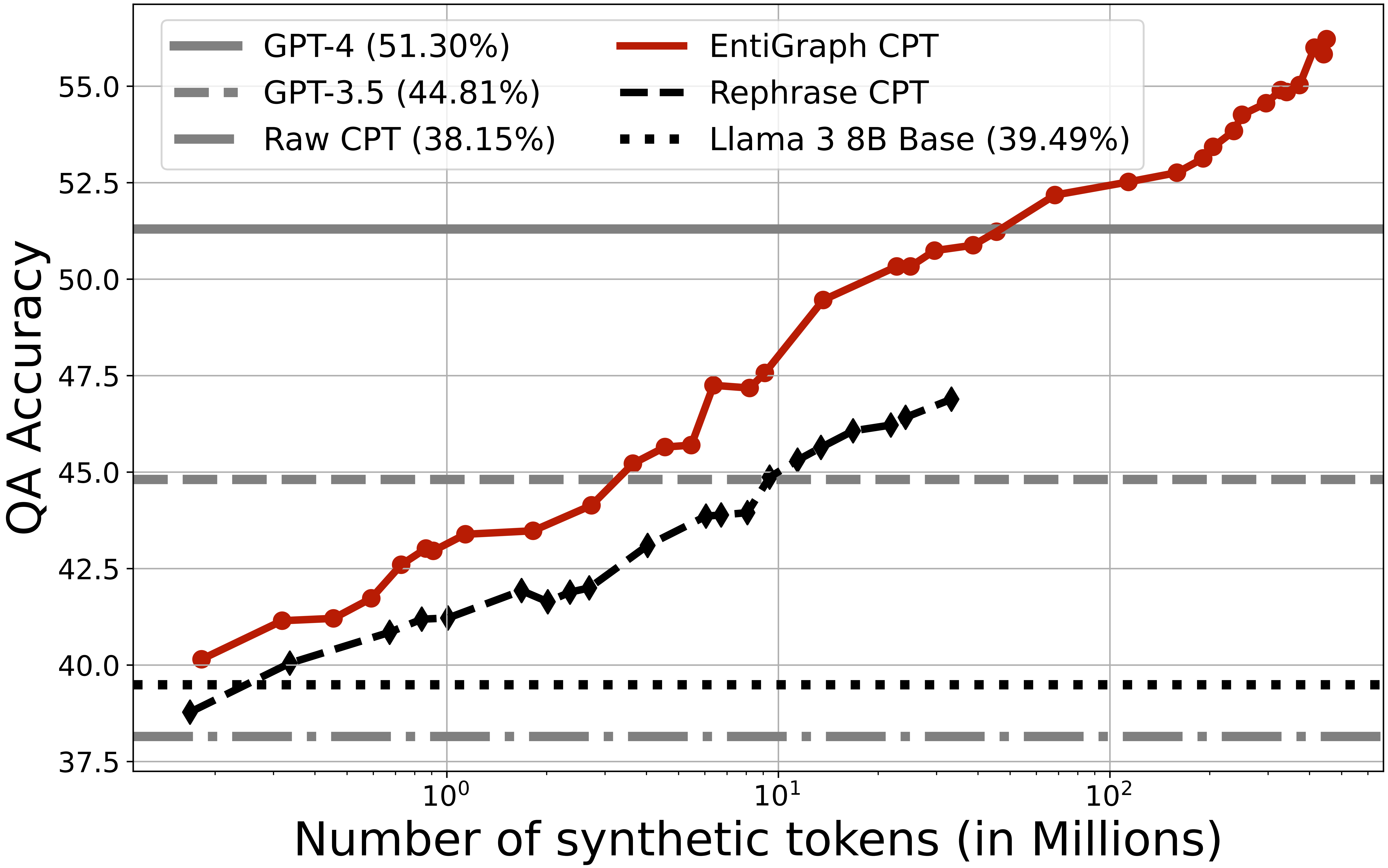
How to generate synthetic data?

- Baseline: simply rephrase the document Pratyush et al. 2024 —> **Lacks diversity**
- **EntiGraph: Entity graph synthetic data generation**

EntiGraph: scalable data generator



EntiGraph performance (closed book)



Synthetic continued pretraining

Goal: teach model the knowledge from a niche domain consisted of a few “source documents"

Step 1: Generate synthetic text based on the source documents

Step 2: Continually pretrain (finetune) the model on generated text

Experiment setup

- A collection of niche (not something model already know) source documents —> **QuALITY books**
- A task that tests a model’s knowledge about the source documents —> **Closed book QA**

How to generate synthetic data?

- Baseline: simply rephrase the document Pratyush et al. 2024 —> **Lacks diversity**
- EntiGraph: Entity graph synthetic data generation

Tasks beyond closed book QA?

Can we approach the open book limit?

Model continually pretrained from Llama-3-8B Base is called EntiGraph CPT

Model	Llama-3-8B Base	Llama-3-8B Base	EntiGraph CPT	EntiGraph CPT
Book access	Closed	Open	Closed	Open
Accuracy	39.49	60.35	56.22	62.60

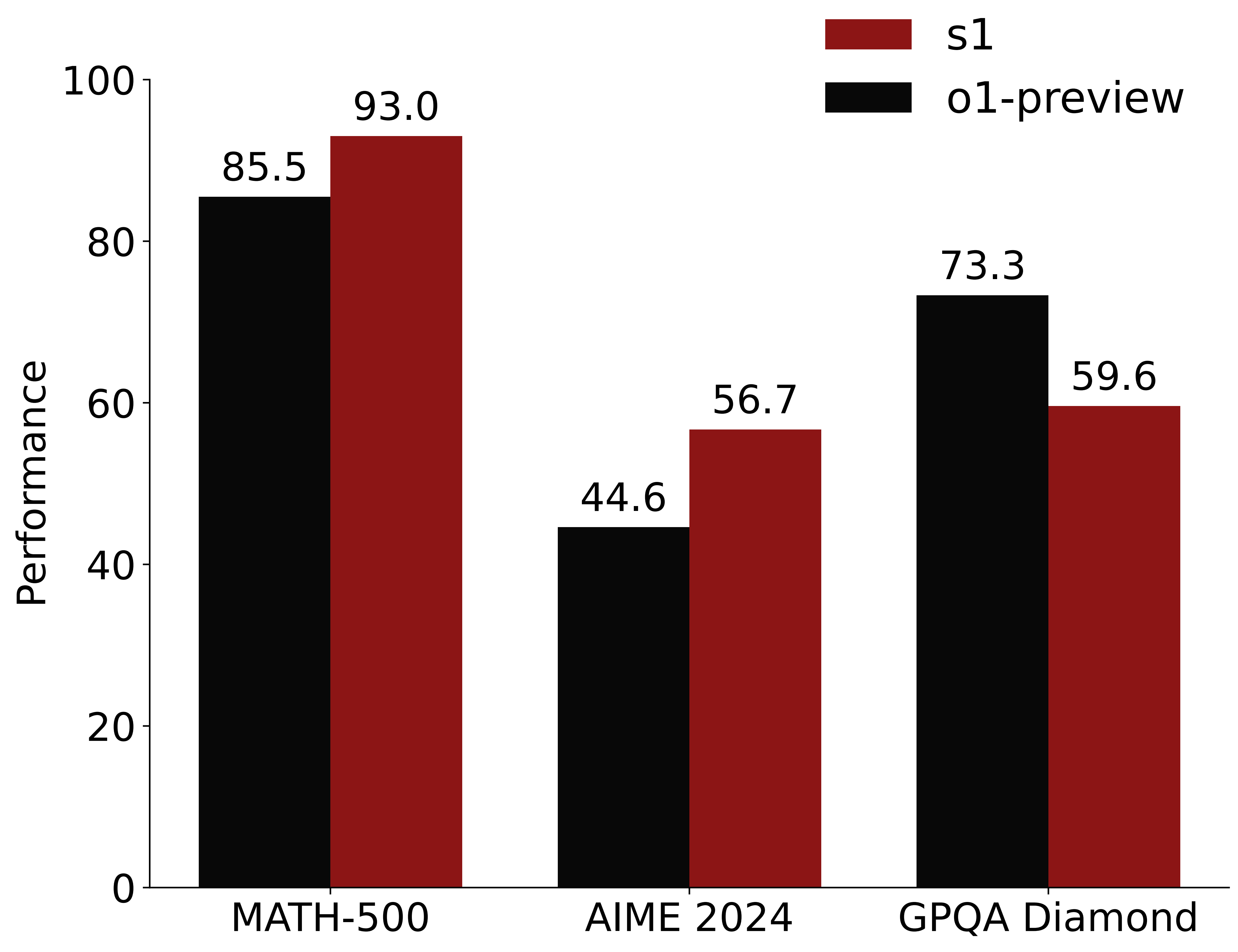
Synthetic continued pretraining + Retrieval tool > Retrieval tool alone

Outline

- ~~Continual knowledge acquisition~~
- Self-improving pretraining capability
- Towards AI-designed AI

Why pretraining?

Supervised finetuning on **1,000** chains-of-thoughts delivers o1-preview level capability



Post-training capabilities are generalizations from pretraining knowledge

Where does the knowledge come from in pretraining?

Thought experiment

- World with 5 tokens “A”, “B”, “C”, “D”, “E”
- Text documents with each token sampled u.a.r. [“BDECD...”, “ACEAC...”,]
- Perform next token prediction with transformer LM: No meaningful learning signal

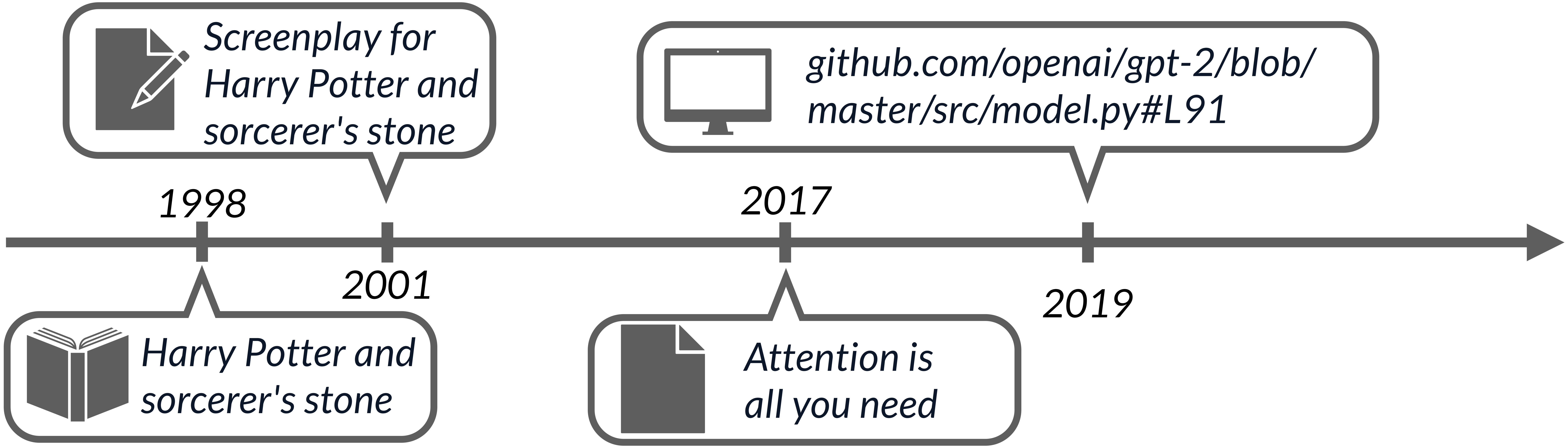
Two views

- Statistical view: Natural language tokens has *statistical* correlations
- Computational view: Natural language has *compressible* patterns

Views aside: **structural correlation** enables learning

Under-exploited sources of correlation

There exists *rich* correlations between documents



Technique: take-advantage of inter-document correlation

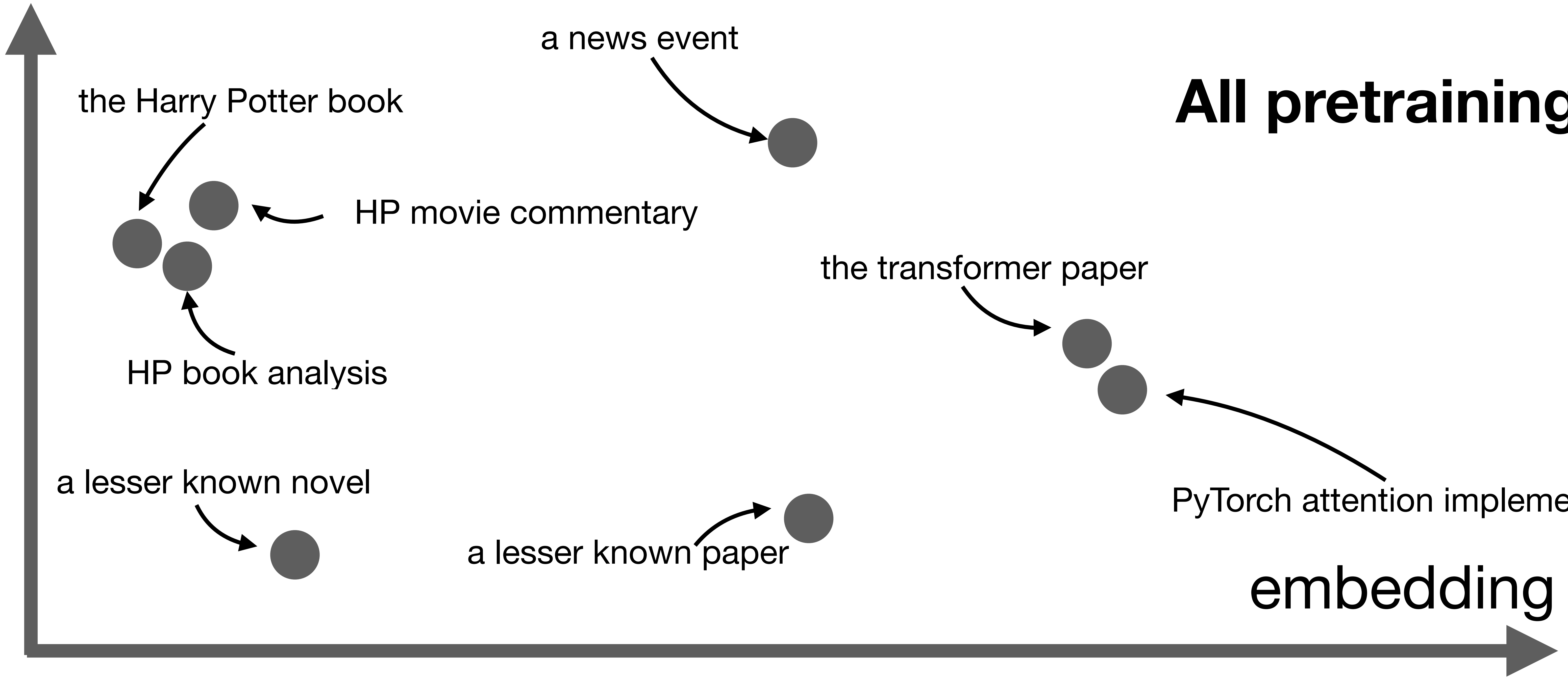
Genuine bootstrap of pretraining capability

No distillation from teacher model

- Pretrain a language model from scratch
- Finetune the model to be a synthetic data generator
- Pretrain a new model on the synthetic data and see improved performance

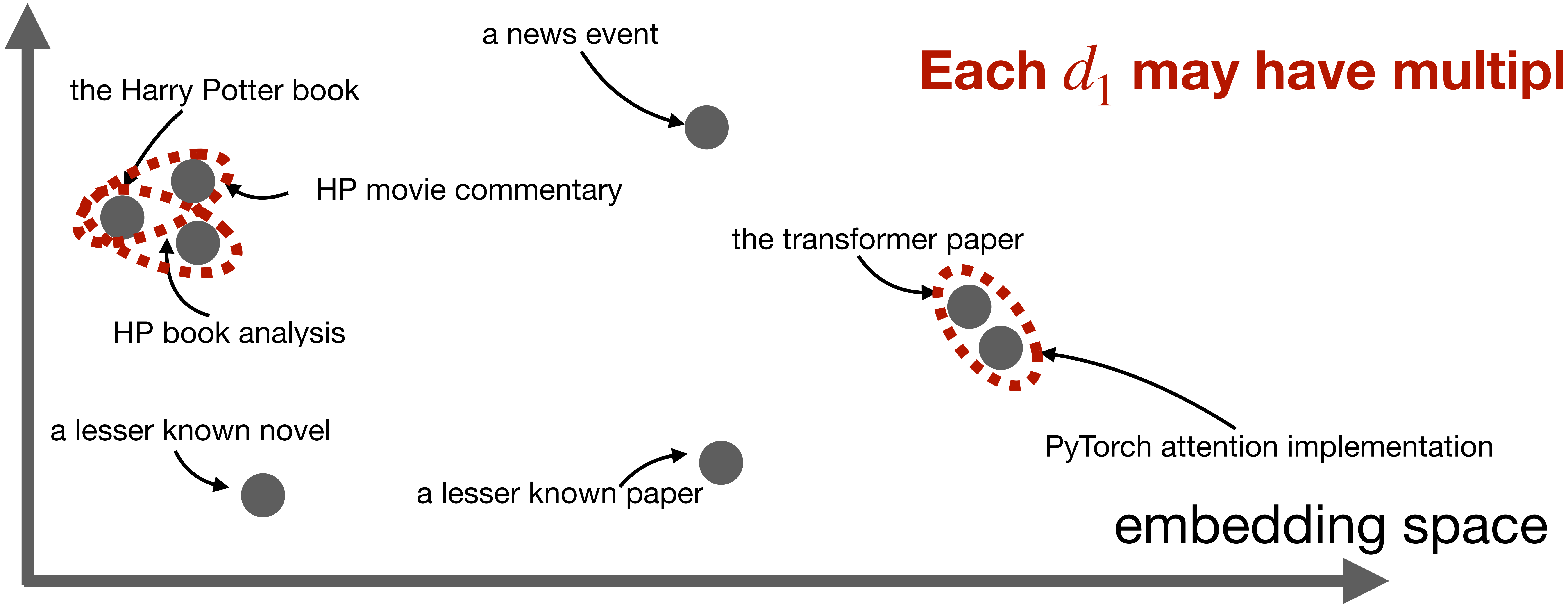
Synthetic bootstrapped pretraining

1. **Nearest-neighbor pairing:** we used DCLM subset and Qwen-0.6B-Embedding



Synthetic bootstrapped pretraining

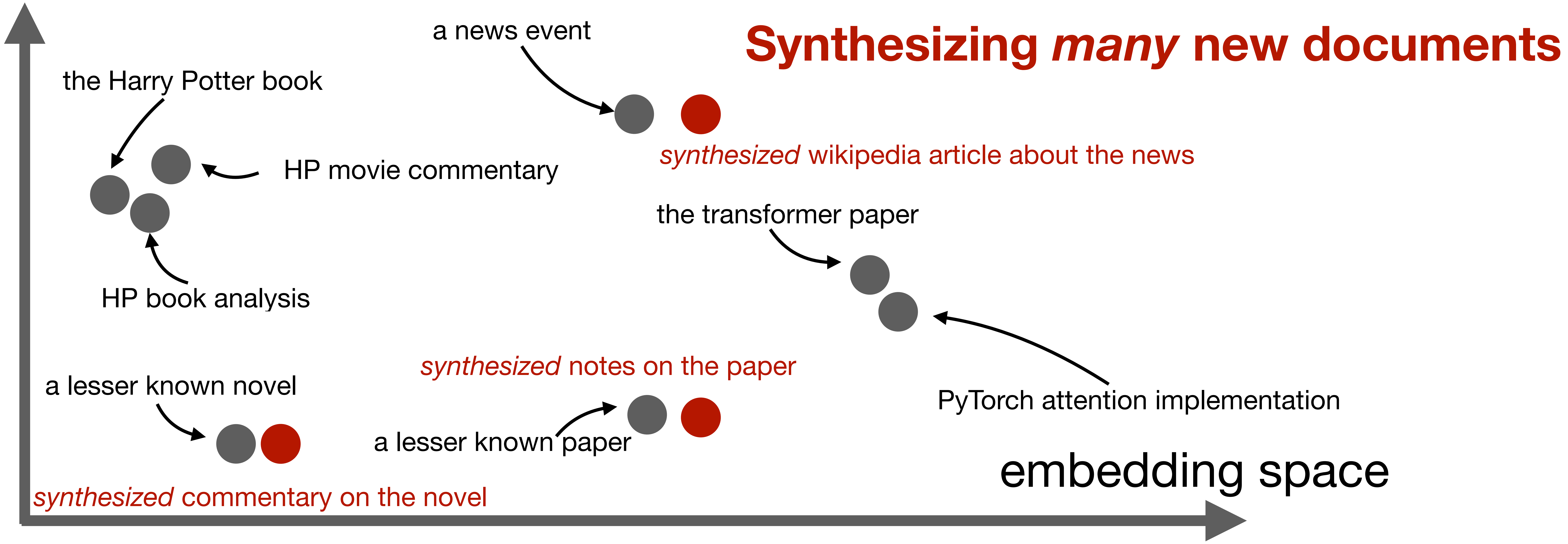
1. **Nearest-neighbor pairing:** we used DCLM subset and Qwen-0.6B-Embedding
2. **Synthesizer tuning:** SFT-like objective $p_\theta(d_1 \mid d_2)$ initialized at pretrained checkpoint



Each d_1 may have multiple paired d_2

Synthetic bootstrapped pretraining

1. **Nearest-neighbor pairing:** we used DCLM subset and Qwen-0.6B-Embedding
2. **Synthesizer tuning:** SFT-like objective $p_\theta(d_1 \mid d_2)$ initialized at pretrained checkpoint
3. **Synthesis at scale:** Temperature=1 allows each document to have varied synthesis



Visualization of synthesized document

Real document

HOME > Chowhound > California >

Coffeehouses in SD

I am always looking for good coffeehouses in San Diego. I am looking for coffeehouses with (obviously) excellent coffeebeans, baristas who know their business (good microfoam, good crema, they should know the different ratios for steamed milk, foamed milk, espresso for the cappuccino, latte etc., etc), good snacks (paninis, cakes etc), good (cold) flavored coffeedrinks and good ambience. So far I haven't found many who are good but would like to hear more suggestion.

Caffe Calabria: By far the best coffeehouse in SD. Excellent coffee and baristas (best cappuccino in SD,), great paninis. Only downside is that they close very early.

Café Lestat: Good example what happens with good coffee and average baristas. Lestat is also using coffeebeans from Caffé Calabria but it is sad to see that they don't know how to make excellent espresso, cappuccino.

E Street Café: Average espresso but strange flavored ...

Synthesis I

Coffee = Love

On a recent trip to San Diego, one thing I wanted to find was coffee. I was worried it might be difficult with the huge decrease in Asian flavors, but I was pleasantly surprised. Many of the Mexican, Italian, and Cuban flavors that are pretty common in California are now in the mix. While I did find a few highly recommended spots, the majority of the locations I checked out were nothing to write home about. At the least, they were not being sold out, which I consider a plus.

One major factor that really needs to be taken into account is the type of machine being used. There are two types of espresso machines: the cheap ones that spit out lots of pressure and don't have any control over the shot, and the expensive ones that create amazing shots and utilize the types of superfine grinds you usually find in Italy. The latter types are much harder to find and cost significantly more.

Another factor to consider is the type of beans being used. If they are done correctly, great espresso can be made from any type of bean, but it's a rare experience to find a place that does ...

Synthesis II

Discuss as:

Coffeehouses: San Diego. The best?

Café Lestat, by far, is a stand-out. (photo by Thomas Moriarty)

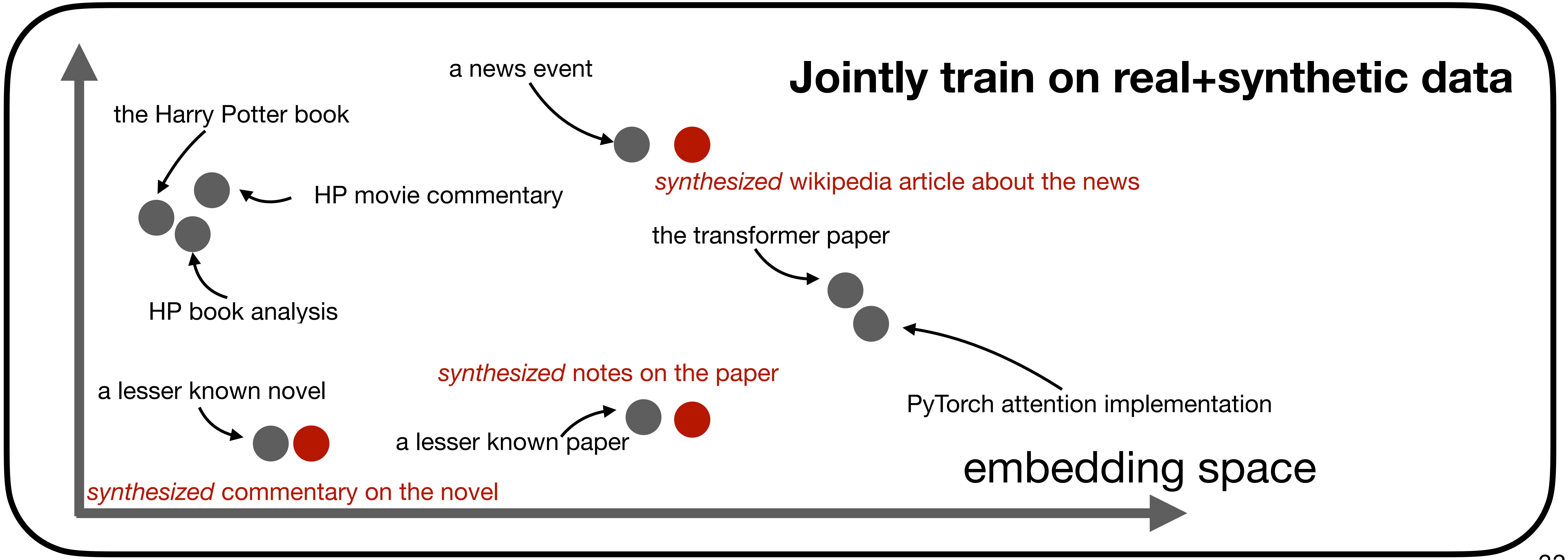
If the coffee scene in San Diego sounds like the one in New York, there's a reason. The upper echelons of NYC's drink scene are the same ones who have taken over the majority of the coffee houses in San Diego. This is probably partly due to the fact that both cities have large student populations, both are home to large coffee drinker bases, both have a large number of espresso shops, and both offer top-notch, hand-crafted cappuccinos. But if there is one city in America that deserves to have its name above that of New York, it's San Diego.

There are just under 100 coffee shops in San Diego, with almost half of them located on University Ave. alone. So finding the perfect coffee shop is crucial. We spent a whole day just roaming around the area, hunting for the best.

In terms of the coffee itself, it's hard to beat Café Lestat. The baristas are amazing and their methods are pristine ...

Synthetic bootstrapped pretraining

1. **Nearest-neighbor pairing:** we used DCLM subset and Qwen-0.6B-Embedding
2. **Synthesizer tuning:** SFT-like objective $p_\theta(d_1 \mid d_2)$ initialized at pretrained checkpoint
3. **Synthesis at scale:** Temperature=1 allows each document to have varied synthesis



Experiment design

Data, model, and evaluation

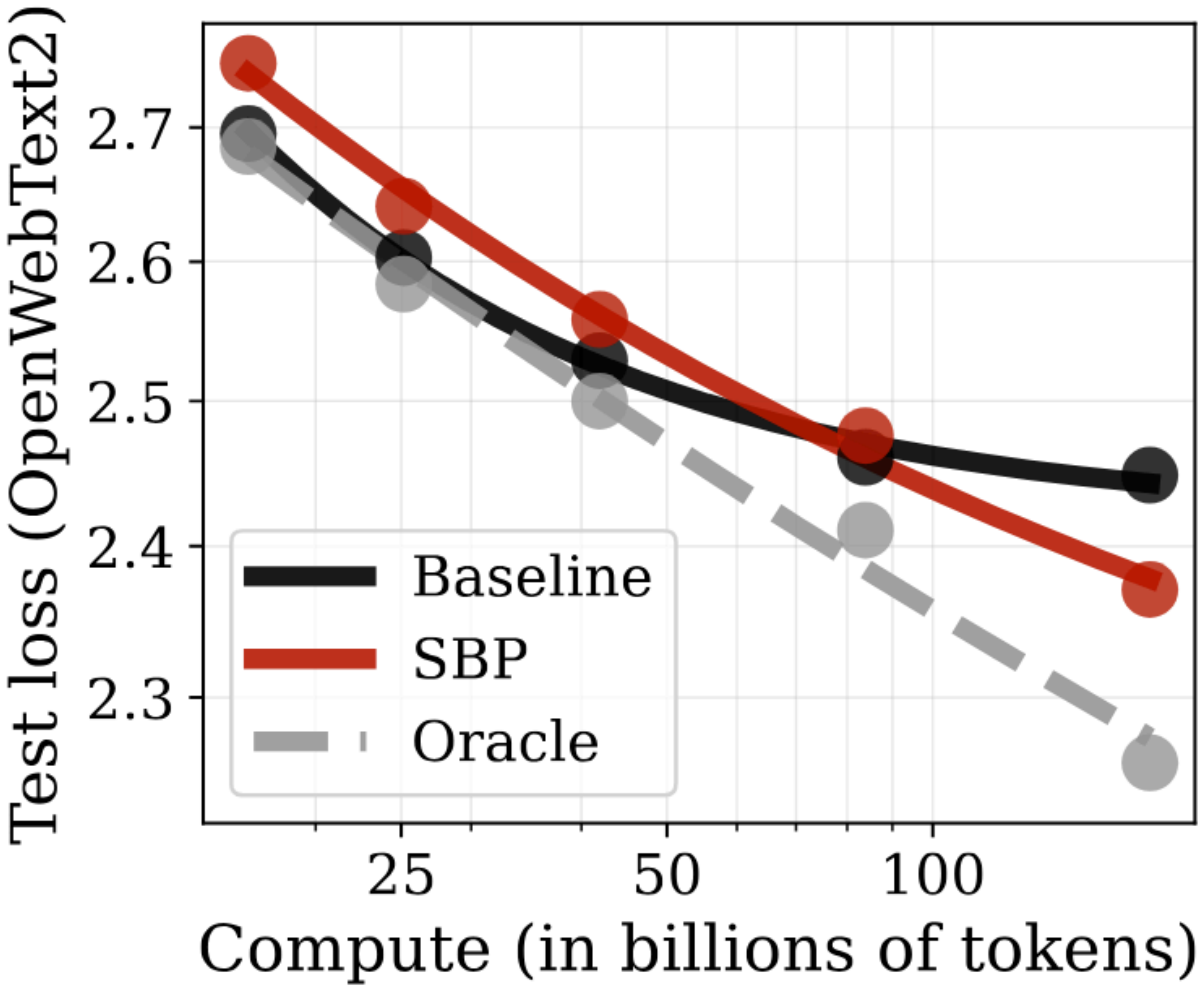
- Data: DCLM dataset
- Model: Llama 3 architecture with additional QK-norm
- Evaluation: 6 QA accuracies and 3 perplexity evaluation

Compute-matched comparison

- Baseline: repetition a fixed amount of data
- SBP: Same data, same (training) compute. Use synthetic data as opposed to repetition
- Oracle: Same training compute with unlimited real data

Training dynamics

- At first, oracle ~ baseline < SBP
- Later on, baseline saturates
- Finally, oracle + SBP continues to scale



Result: 40% improvement attained by the oracle

Benchmark	200B-scale			1T-scale (3B)			1T-scale (6B)		
	Baseline	**SBP**	**Oracle**	**Baseline**	**SBP**	**Oracle**	**Baseline**	**SBP**	**Oracle**
Perplexity on held-out data ↓									
OpenWebText2	5.74	-0.53	-1.02	4.51	-0.02	-0.12	4.25	-0.06	-0.21
LAMBADA	6.87	-0.85	-1.86	4.33	-0.03	-0.22	3.63	-0.06	-0.25
Five-shot MMLU	3.83	-0.36	-0.51	3.17	-0.06	-0.05	3.08	-0.08	-0.13
QA accuracy ↑									
ARC-Challenge (0-shot)	35.32	+1.28	+2.82	42.66	+1.62	+3.84	47.44	+0.77	+0.17
ARC-Easy (0-shot)	68.94	+2.65	+4.29	75.63	+0.42	+2.11	78.70	+0.51	+0.85
SciQ (0-shot)	90.50	+1.00	+2.40	93.20	+0.80	+0.50	92.90	+1.90	+1.80
Winogrande (0-shot)	60.14	+1.90	+5.53	65.19	+1.42	+2.92	70.17	+0.47	+2.36
TriviaQA (1-shot)	22.51	+3.36	+7.37	36.07	+0.25	+0.59	40.64	+0.49	+3.19
WebQS (1-shot)	8.56	+3.74	+10.83	19.34	+0.54	+0.44	19.88	+3.79	+5.22
Average QA accuracy	**47.66**	**+2.32**	**+5.54**	**55.35**	**+0.84**	**+1.73**	**58.29**	**+1.32**	**+2.26**

Synthetic data quality

	Repetition ↓	Duplicate@1M ↓	Non-factual ↓	Pair-irrelevance ↓	Pair-copying ↓
200B-scale	4.3%	0.8%	15.1%	25.6%	0.1%
1T-scale (3B)	3.9%	0.8%	8.7%	7.8%	0.9%
1T-scale (6B)	2.6%	0.3%	6.5%	6.0%	0.3%
Real data	1.8%	0.7%	1.8%	n.a.	n.a.

- Better data quality with larger scale
- Synthesized data is not mere repetition

Outline

- ~~Continual knowledge acquisition~~
- ~~Self-improving pretraining capability~~
- Towards AI-designed AI

Why AI may do a better job?

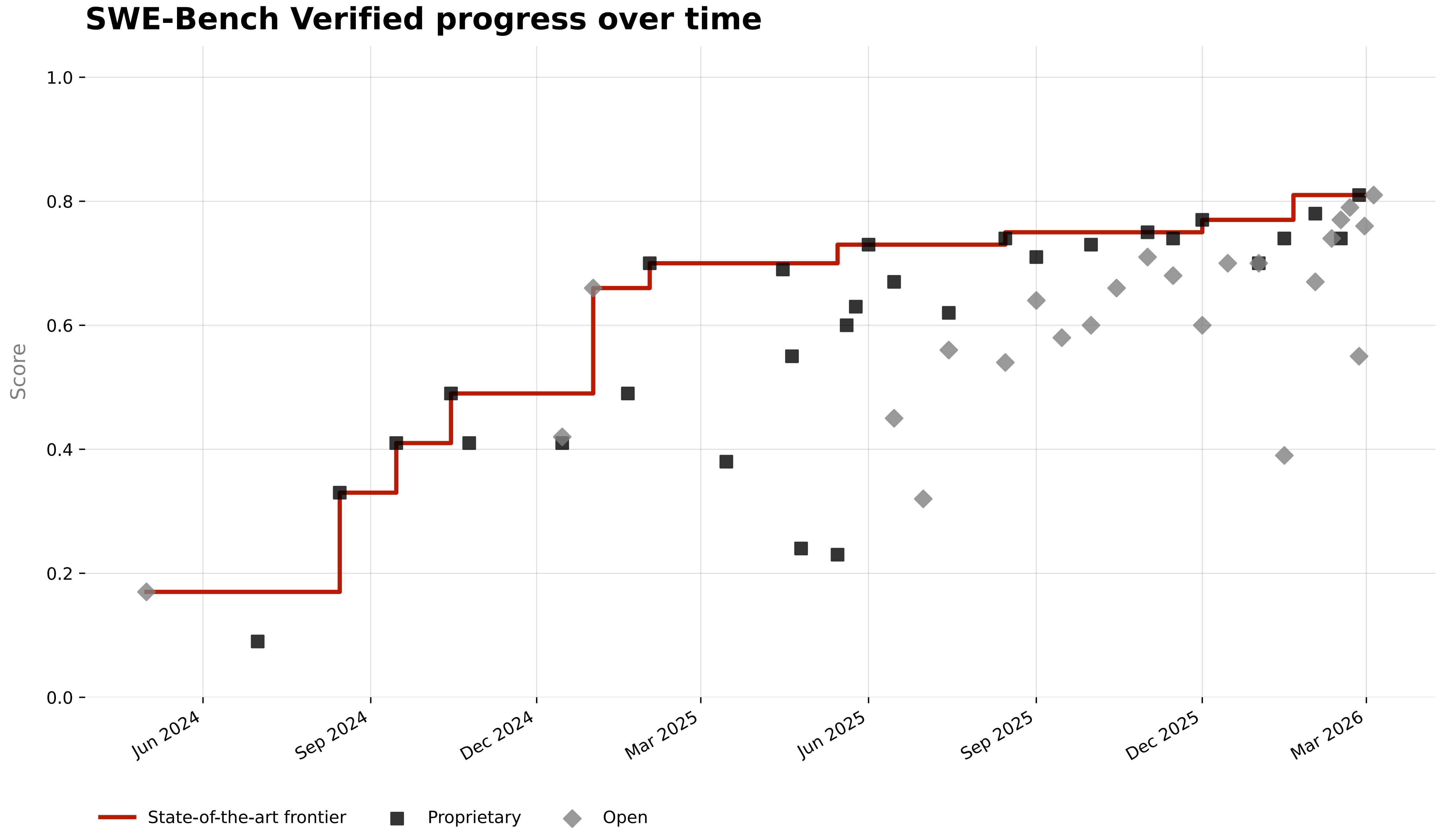
Ronald Fisher

Two steps of scientific method

- Generating hypothesis
- Run experiments to falsify the hypothesis

Status of AI

- AI progress are benchmark driven
- AI experiments materializes in writing code



Research environment abstraction

```
class ResearchEnv:

  @abstract
  def context(self):
    # passed into LM
    # describing the task
    pass

  @abstract
  def value(self, idea: str):
    # goodness of the idea
    # benchmark perf for AI
    pass
```

```
class AIResearchEnv(ResearchEnv):
  codebase: str
  resource: str
  sandbox_factory: Callable

  def context(self):
    return self.codebase

  def value(self, code_diff: str):
    sandbox = sandbox_factory(resource)
    sandbox.exec(f"patch -p {code_diff}")
    _ = sandbox.exec("bash run.sh")
    std_out = sandbox.exec("bash eval.sh")
    return std_out
```

Two crucial AI research task: pretraining and post-training

Pretraining environment

- `nanoGPT.codebase # standalone .py script performing gpt-2 pretraining`
- `nanoGPT.resource = "8xH100" # for benchmarking purpose`
- `eval.sh # time to reach 3.28 on OpenWebText2 test set`

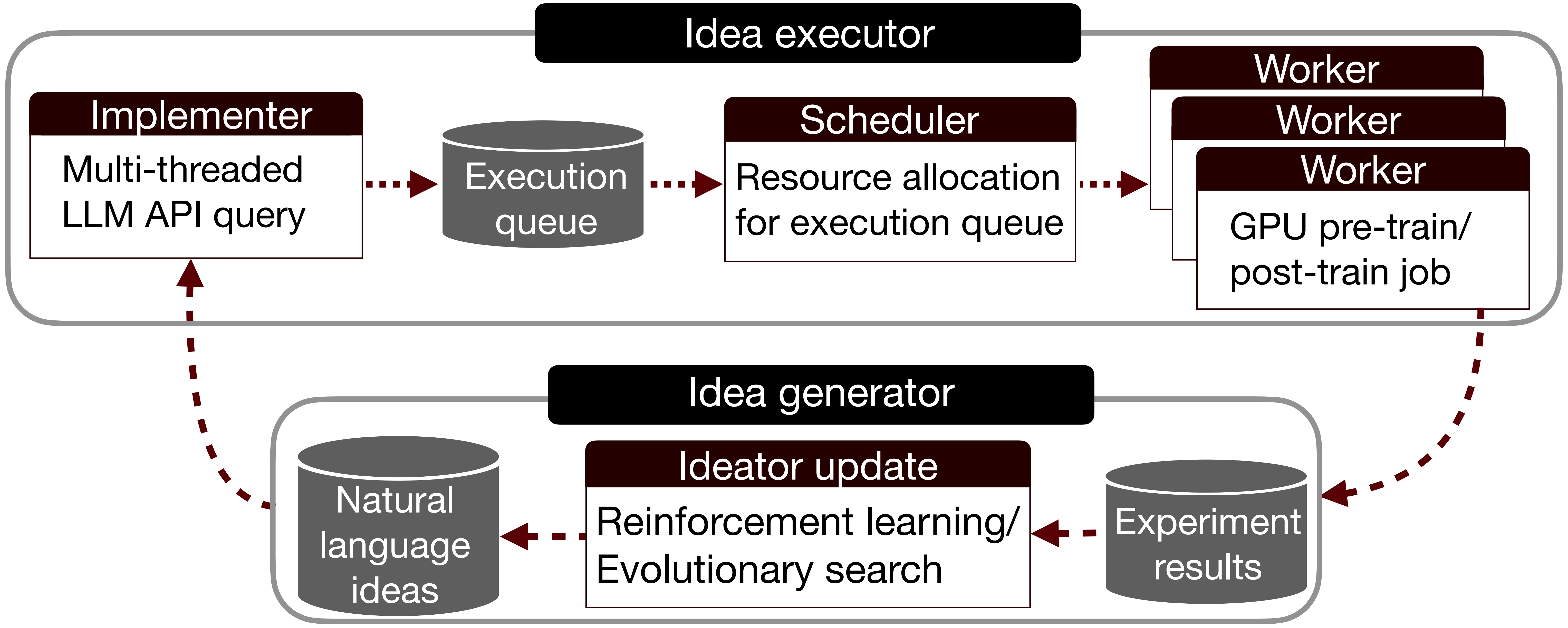
Post-training environment

- `MathReasoningGRPO.codebase # train on GSM8K, test on Math500`
- `MathReasoningGRPO.resource = "1xB200" # on-device sampler <-> trainer`
- `eval.sh # Math500 test accuracy`

Automated AI researcher

- `idea = researcher.ideator(research_env.context)`
- `code_diff = researcher.executor(research_env.context, idea)`
- `results.append((idea, research_env.value(code_diff)))`
- `researcher.learn(results)`

System design



`learn(results)` as iterative test-time search

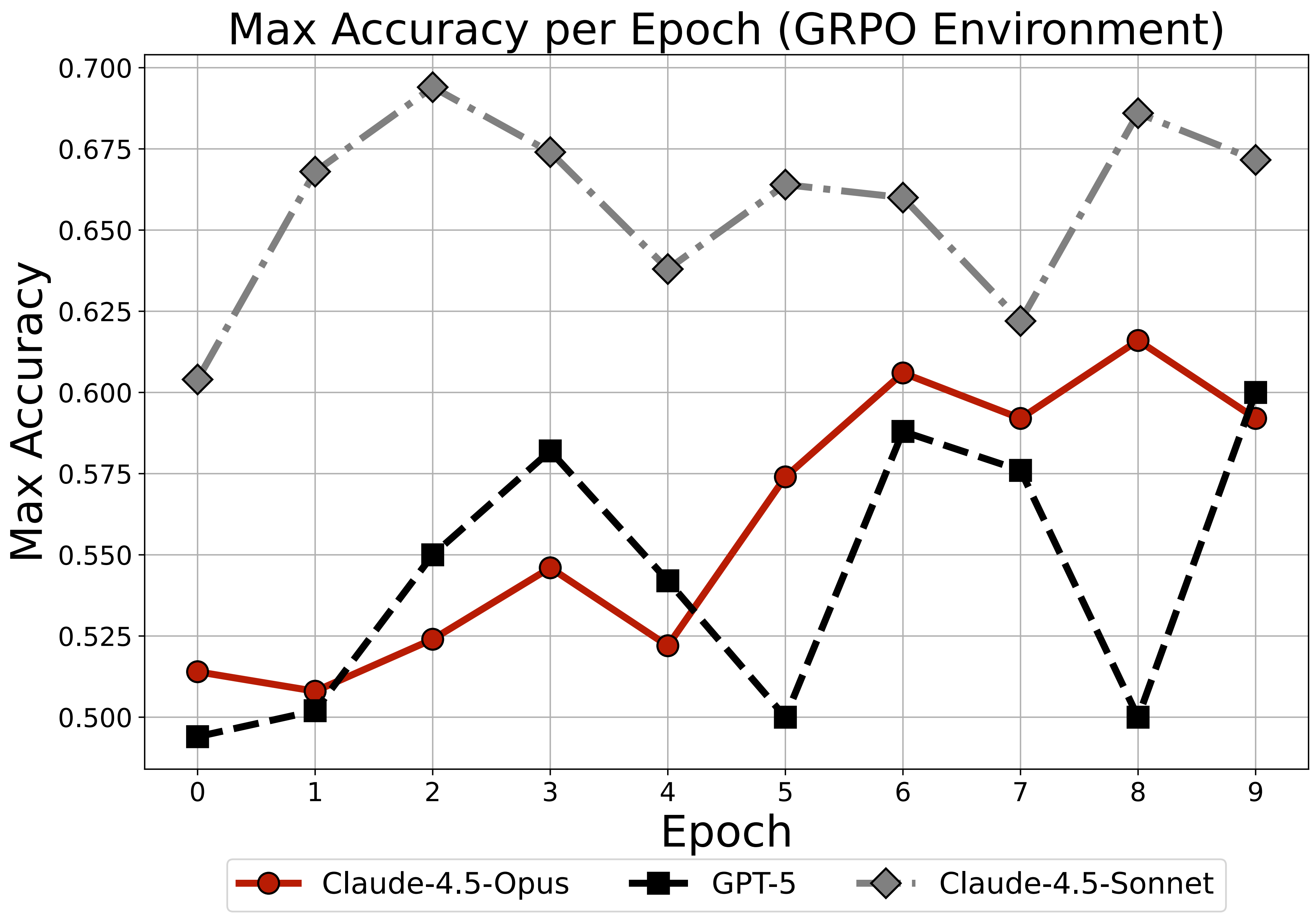
Keep a bank of all past ideas

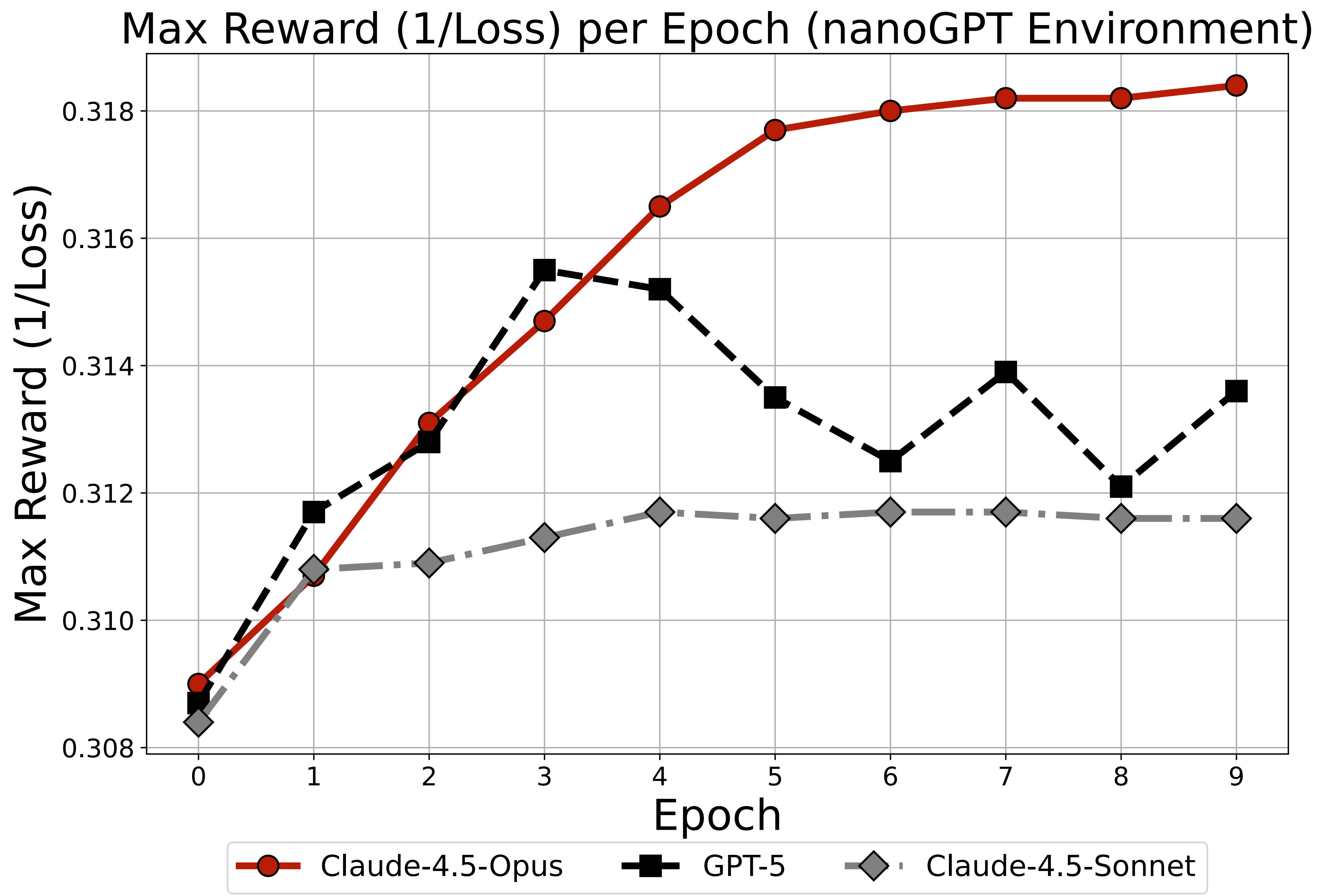
- Exploit by combining strength of high value idea
- Explore by generating completely new ideas

	Post-training (GRPO)	Pretraining (nanoGPT)
Baseline	48.0%	36.9 min
Test-time search	69.4%	19.7 min
Human expert	68.8%	2.1 min
	CS336 assignment	nanoGPT leader board

Search dynamics

Search 256 ideas per epoch for GRPO // 128 ideas per epoch for nanoGPT





Connection to budget forcing

Shifting gears to test-time scaling for math reasoning

- Forcing fewer than 100 thinking tokens

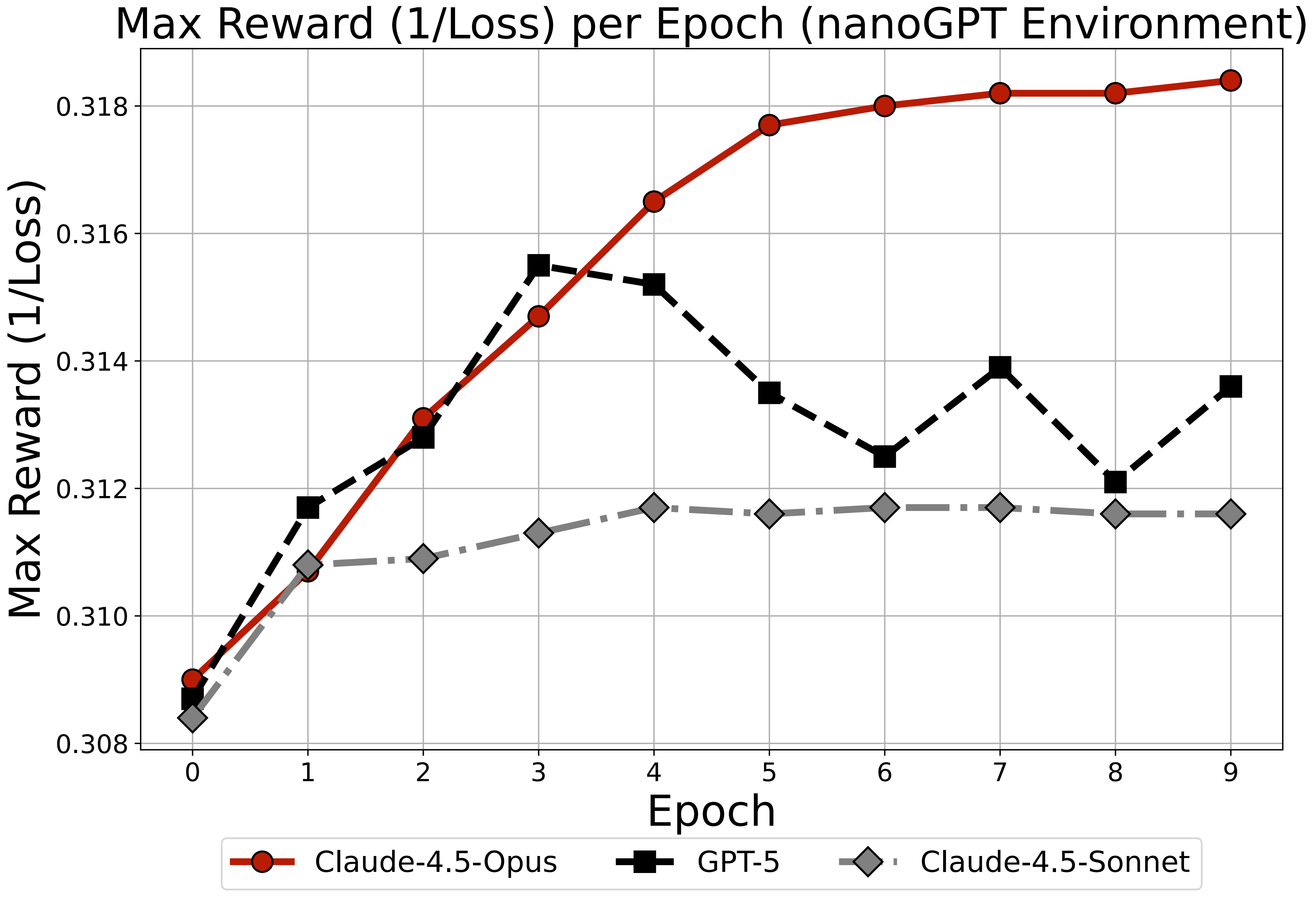
Without BF	`<thinking>`	`...first 100 tokens...`	`the 101-th token thinking`
With BF	`<thinking>`	`...first 100 tokens...`	`</thinking> Final answer:`

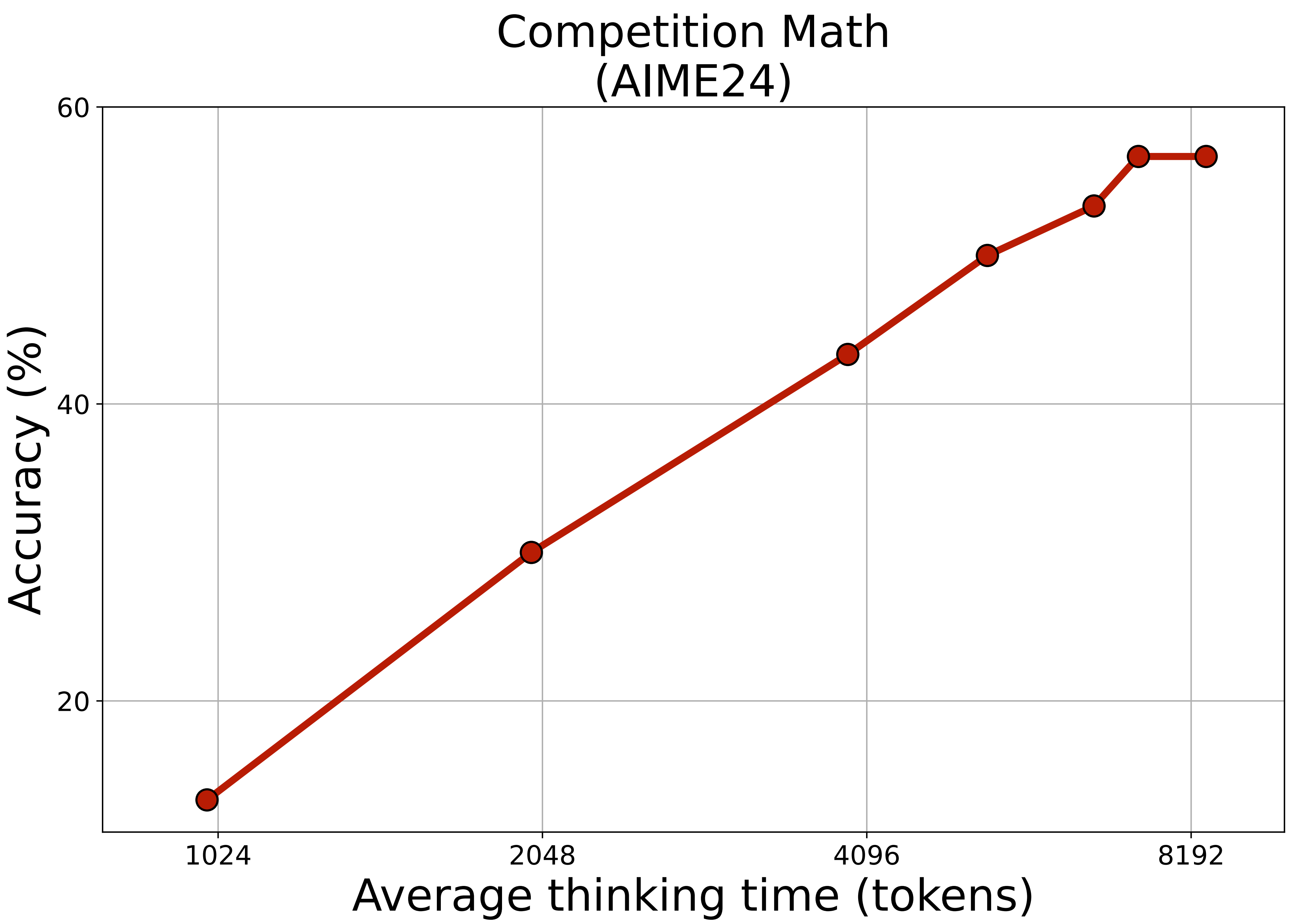
- Forcing more than 1000 thinking tokens

Without BF	`<thinking>`	`...first 529 tokens...`	`</thinking>...`
With BF	`<thinking>`	`...first 529 tokens...`	`Wait,` `...continues...`

Connection to budget forcing

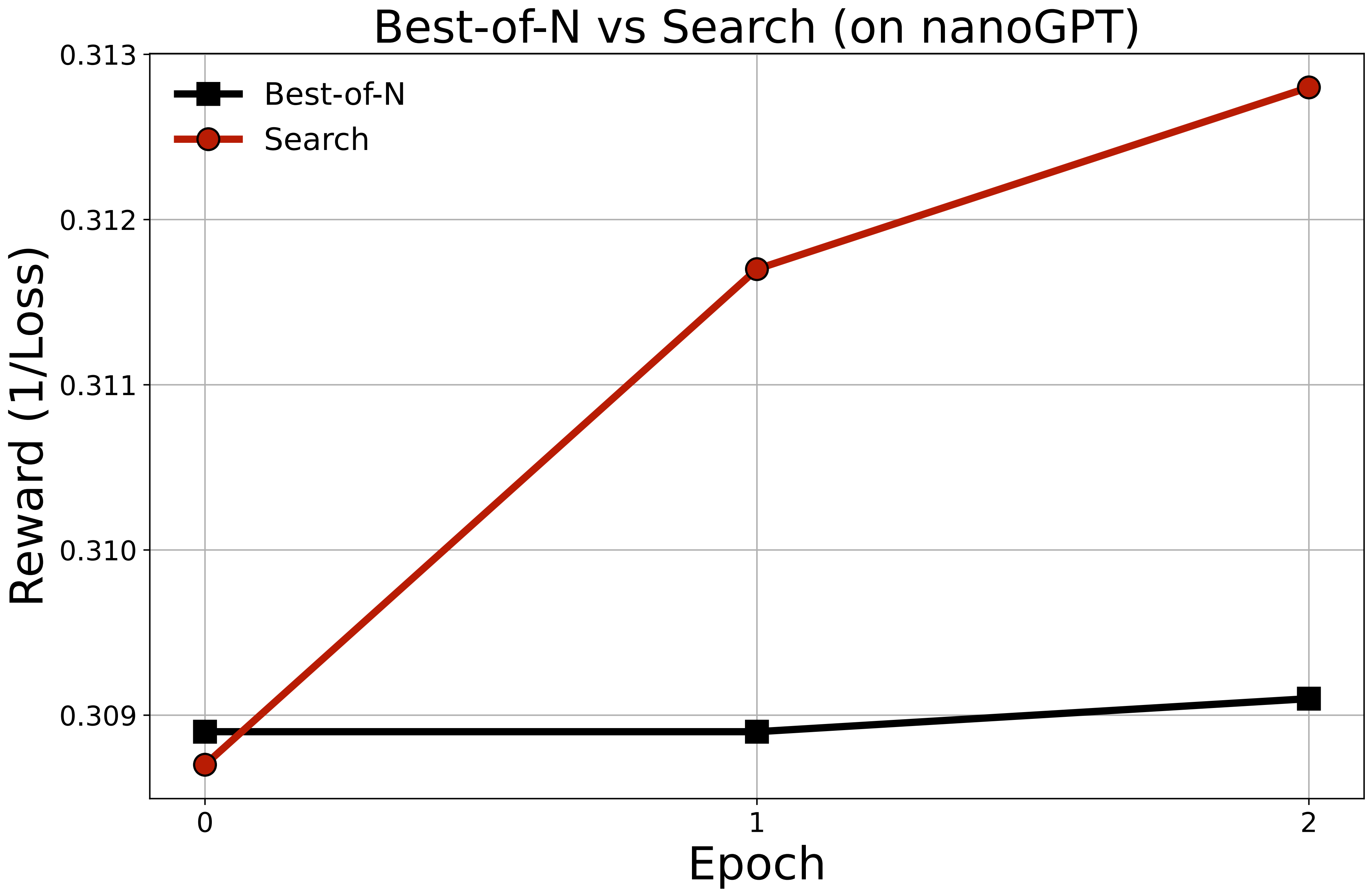
Scaling test-time compute *naively* yields performance improvement but both plateau quickly

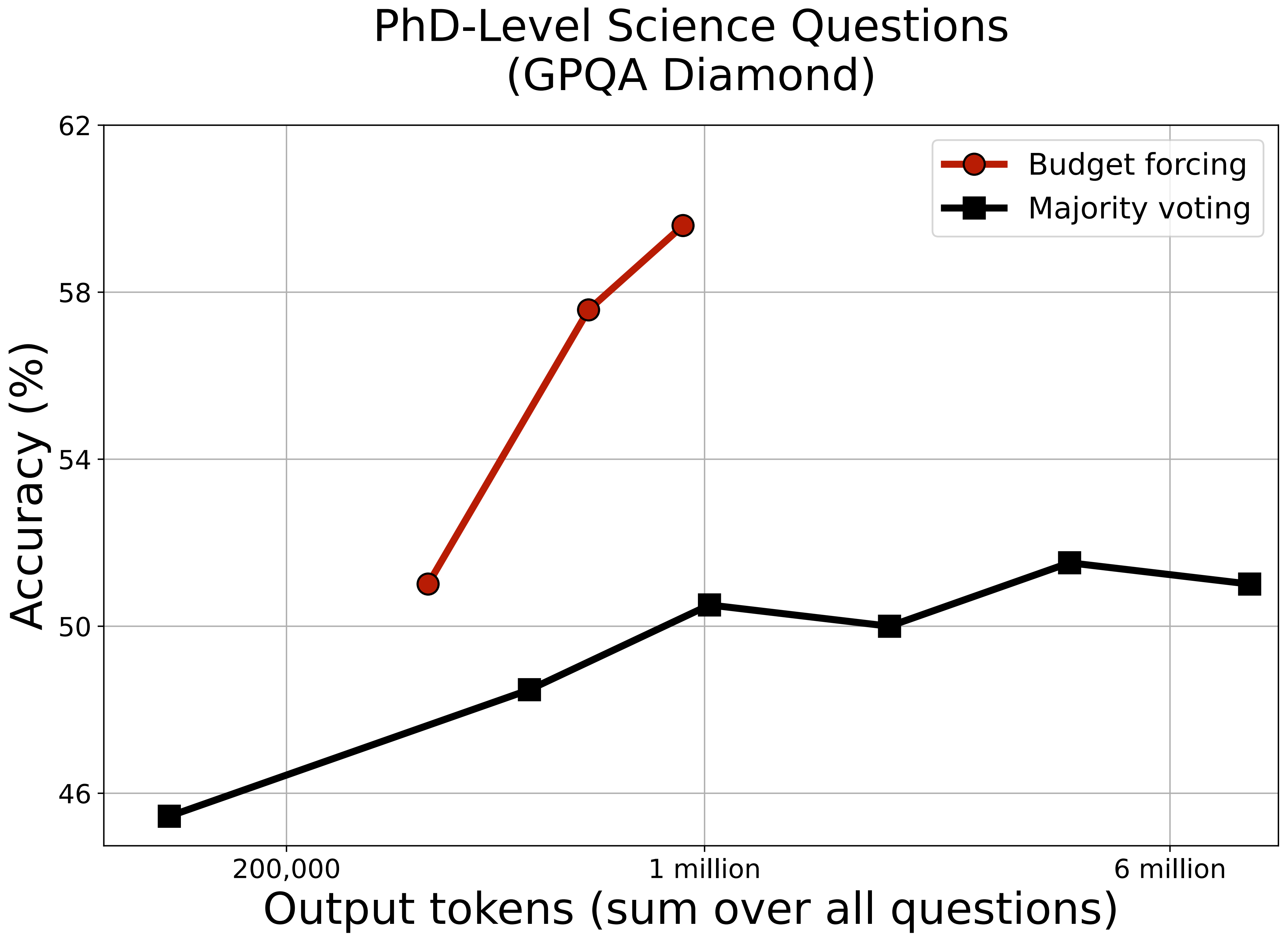




Parallel search vs. sequential search

Sequential search > parallel search





Idea example

Create mathematical working memory simulation by maintaining a context buffer of mathematical facts, definitions, and intermediate results during problem solving. This buffer gets updated as the model works through problems and provides additional context for subsequent mathematical steps, simulating how humans maintain mathematical working memory during complex calculations.

```
class MathContextBuffer:
  # Buffer for mathematical
  # working memory

  def add_expressions(
    self, expressions: str):
    ...

  def get_context(
    self, query: str):
    ...
```

```
def train_loop(train_prompts):
  buffer = MathContextBuffer()
  ...
  for prompt in train_prompts:
    technique = buffer.get_context(prompt)
    prompt += technique
    ...
    response = …
    ...
    buffer.add_expressions(response)
```

Performance :+10% from baseline. I have a similar buffer (telescoping, epsilon ball of space, Jensen, ...)!

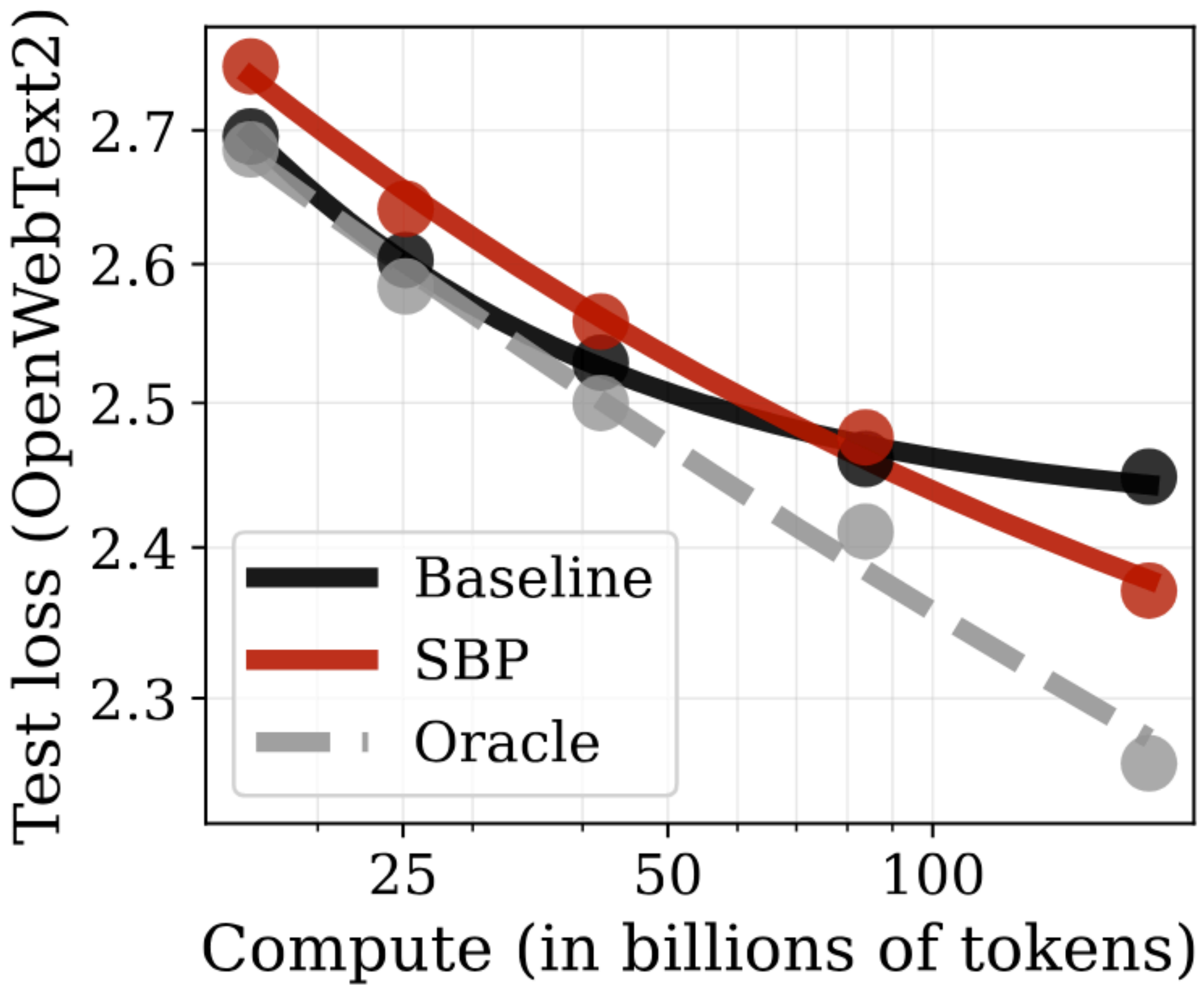
Outline

- ~~Continual knowledge acquisition~~
- ~~Self-improving pretraining capability~~
- ~~Towards AI-designed AI~~

Can AI self-improve to be stronger than its creator?

A continually self-improving AI is one that, once created, can autonomously and continually improve itself better than its human creators can improve it.

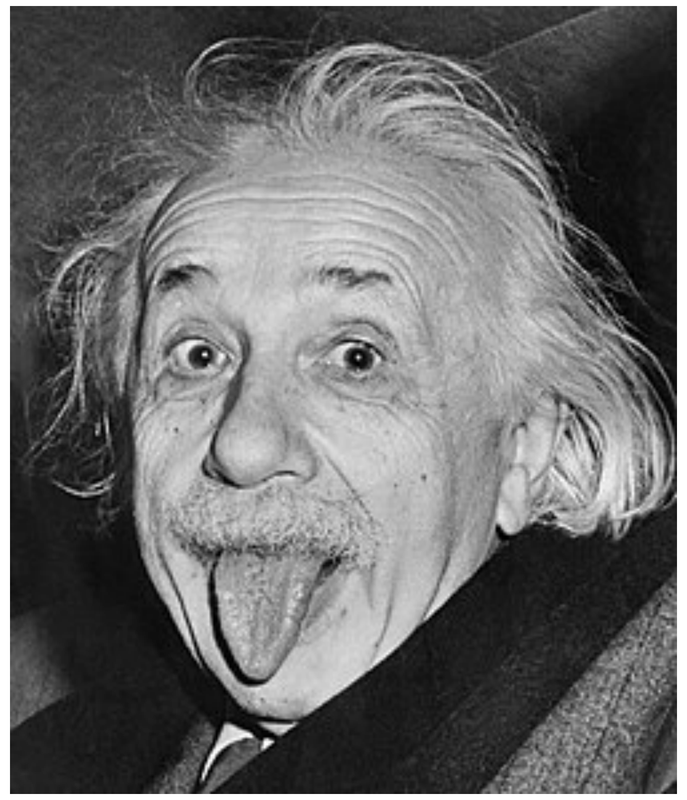
Right now, AI > human by stacking quantity to overcome quality limitation



DeepScholarRL-ablations	37 seconds ago	Public	88	0
nanogpt-training	2026-01-20	Public	2	0
marin-metagrads	2026-01-30	Public	13	0
marin	2026-01-05	Public	9	0
grpo-math-no-example-prompt	2026-01-19	Public	5256	0
nanogpt30min_synced	2025-12-05	Public	1	0
grpo-math-no-example-prom...	2025-12-02	Public	37561	0
nanogpt30min_synced_old	2025-11-28	Public	10530	0
nanogpt_ES_claude	2026-01-20	Public	6751	0
RL-ablations	2025-12-10	Public	83	0

- Human data is better, but AI data is infinite.
- Human researchers are stronger, but AI researchers work too hard.

Can AI self-improve to be *really* stronger than its creator?



Albert Einstein

$$<\qquad G_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}$$

- A theory can evolve and can mutate. It has a life of their own.
- Einstein created the field equations that is smarter than himself.
- By analogy, humans can create AI that is smarter than humans themselves.

Can AI self-improve to be *really* stronger than its creator?

Crash course on GR From particle equation to (Newtonian) field equation

$$\mathbf{F} = -\frac{GMm}{r^2}\hat{\mathbf{r}} \qquad \mathbf{F} = \int -\frac{Gm\,\rho(\mathbf{r}')\,dV'}{|\mathbf{r}-\mathbf{r}'|^3}(\mathbf{r}-\mathbf{r}')$$

Introduce the potential function that normalizes the test mass $\mathbf{g} = \mathbf{F}/m = -\nabla\Phi(\mathbf{r})$

$$\underbrace{\nabla^2\Phi}_{\text{geometry (potential)}} = \underbrace{4\pi G\rho}_{\text{matter (source)}}$$

Einstein postulate that what we experience as gravity is in fact wrapping of spacetime

$$G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2}\mathscr{R}\,g_{\mu\nu} = \frac{8\pi G}{c^4}T_{\mu\nu}$$

where an interval is measured as $ds^2 = g_{\mu\nu}\,dx^\mu\,dx^\nu$ and $R_{\mu\nu}, \mathscr{R}$ are all derivatives of $g_{\mu\nu}$

Can AI self-improve to be *really* stronger than its creator?

The original field equation predicts the expansion of universe, mathematically.

Cosmological assumption: universe is a motionless dust field with matter.

$$T_{00} = c\rho^2, T_{\mu\nu} = 0 \text{ or all other } \mu\nu.$$

Under this assumption and isotropic assumption, there are 3 possible ansatzes. Simplest one is

$$ds^2 = -c^2\,dt^2 + a^2\left(dx^2 + dy^2 + dz^2\right)$$

Generally, matter density ρ and scale factor a depends on (t, x, y, z). But symmetry restricts

$$\rho = \rho(t), a = a(t)$$

Plug in $g_{\mu\nu}$ and $T_{\mu\nu}$ into the field equation $G_{\mu\nu} = \dfrac{8\pi G}{c^4} T_{\mu\nu}$, we get

$$\frac{3\dot{a}^2}{a^2} = \frac{8\pi G\rho}{c^2} \qquad \frac{2\ddot{a}}{a} + \frac{\dot{a}^2}{a^2} = 0$$

Can AI self-improve to be *really* stronger than its creator?

The field equation suggests universe can't be static, unless it is empty

$$\frac{3\dot{a}^2}{a^2} = \frac{8\pi G\rho}{c^2} \text{ implies } \dot{a} = 0 \Rightarrow \rho = 0$$

In fact, we can solve the exact rate of universe's expansion (and hence the Big Bang)

$$\frac{2\ddot{a}}{a} + \frac{\dot{a}^2}{a^2} = 0 \text{ implies } a(t) \propto t^{2/3} \Rightarrow \rho(t) \propto t^{-2}$$

Einstein arrived at essentially the same conclusion, but the belief around 1910s is that the universe consisted of only of Milky Way which is eternal and static.

So he modified his equation in 1917

$$R_{\mu\nu} - \frac{1}{2}\mathscr{R}\, g_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu} \longrightarrow R_{\mu\nu} - \frac{1}{2}(\mathscr{R} - 2\Lambda)\, g_{\mu\nu} = \frac{8\pi G}{c^4} T_{\mu\nu}$$

Can AI self-improve to be *really* stronger than its creator?

However, in 1929, Hubble observed a linear relation: galaxy's distance ~ recession velocity

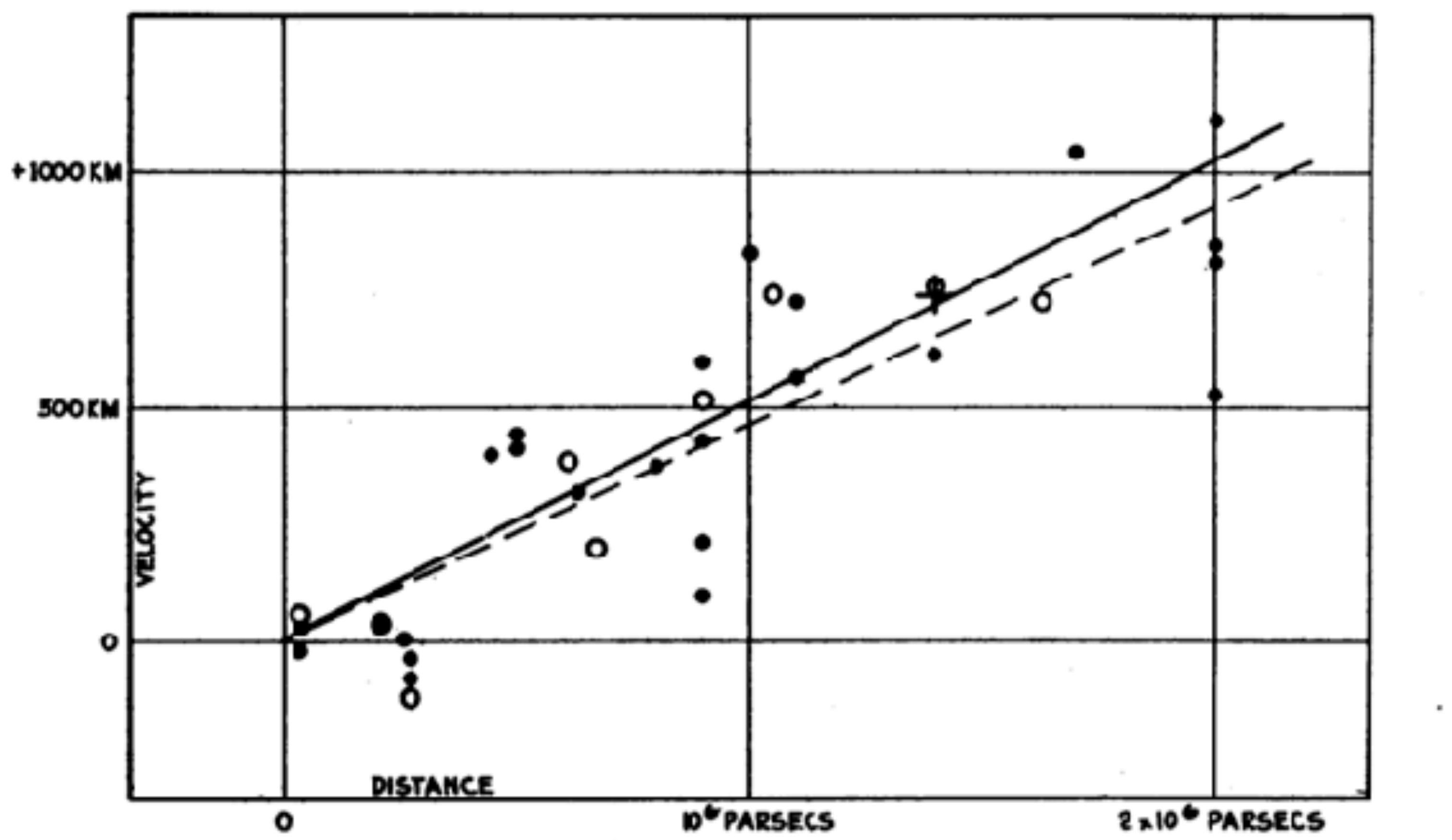
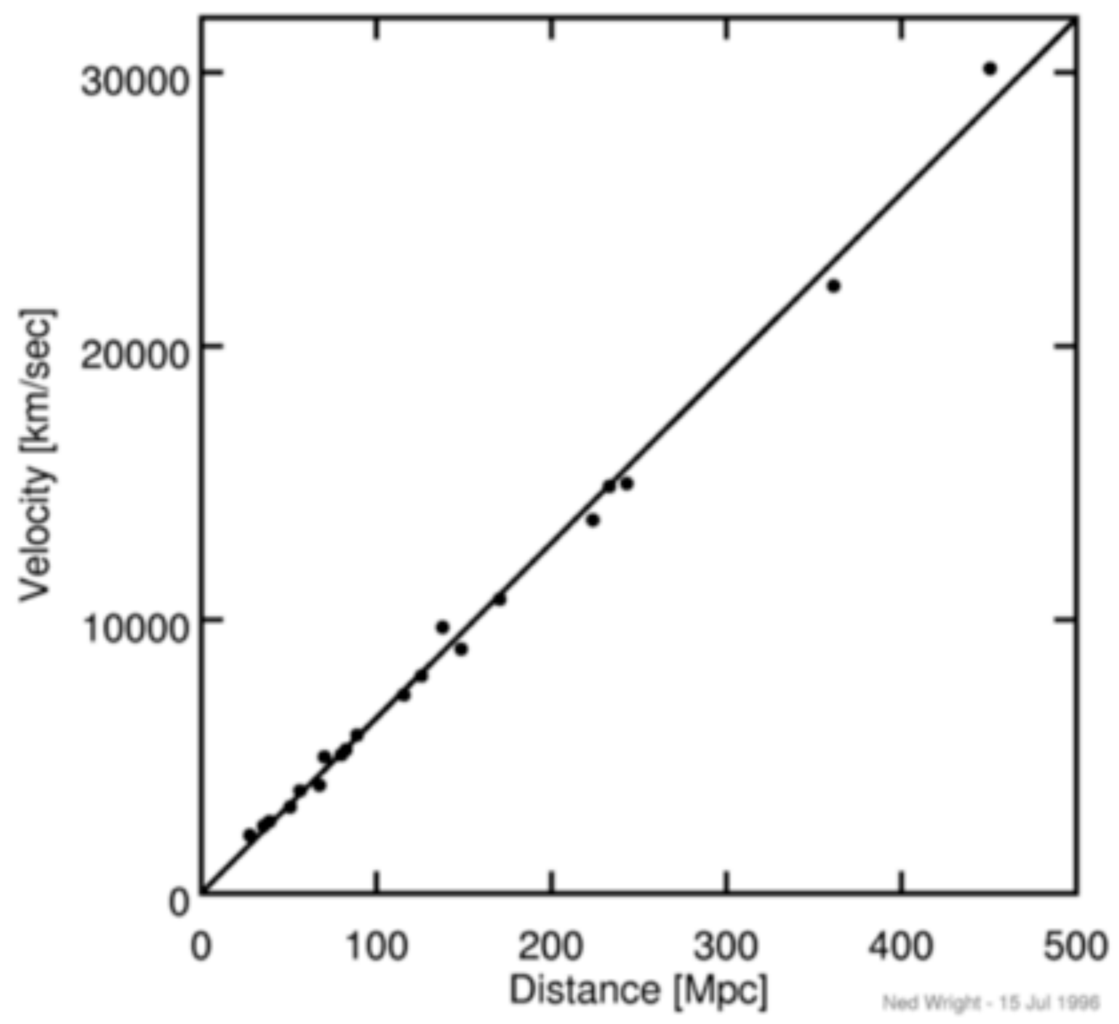


FIGURE 1
Velocity-Distance Relation among Extra-Galactic Nebulae.

Hubble's 1929 data



Modern data from 1996

This linear relation is exactly what's predicted by Einstein's unmodified field equation

$$\frac{3\dot{a}^2}{a^2} = \frac{8\pi G\rho}{c^2} \text{ implies } \dot{a} \propto a$$

Einstein later confessed the 1917 modification is the "greatest blunder" of his life.

The moment the theory is created, it is above its creator.

Can AI self-improve to be *really* stronger than its creator?

Absolutely yes!