(Practical) Reinforcement Learning For LMs

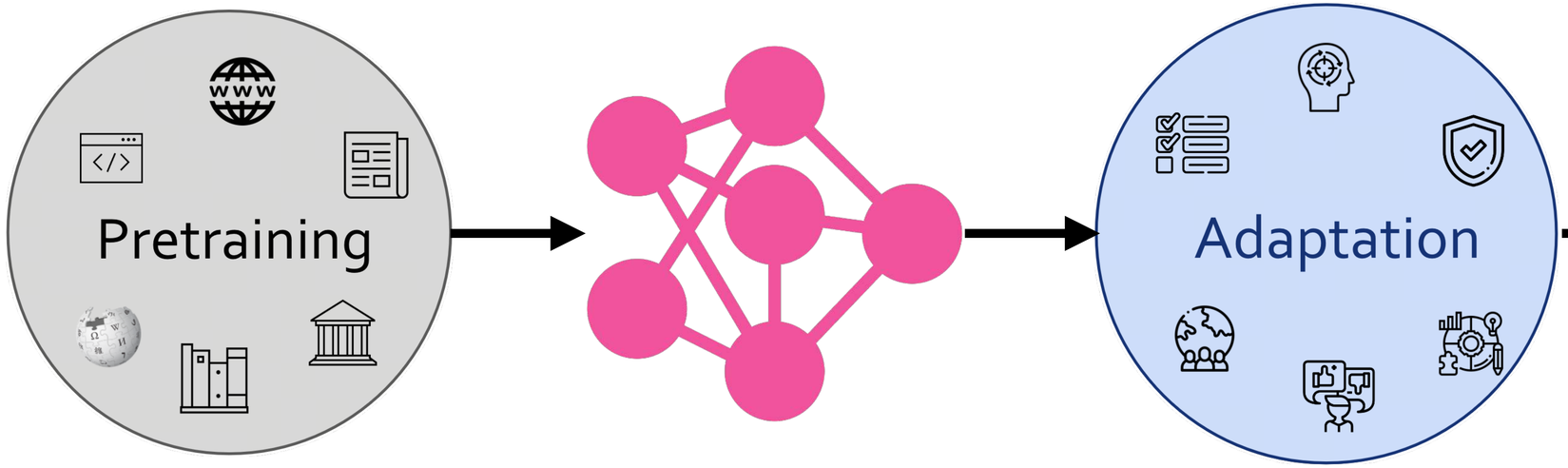
Hamish Ivison 10/2/25

What am I going to talk about?

1. Why should we care doing RL with LMs?
2. How do we train an LM with RL?
3. How do we train an LM with RL *quickly*?*
4. How do we train with *tools and environments?* [DR Tulu]
5. Training an LM with RL *in practice.* [Olmo 3]

Why should we care about RL and LMs?

Background: How to make an LM?



Follow natural language instructions

Be aware of harmful behaviors

Respond according to human preference

Improve core skills

Why adapt?

Qwen / **Qwen3-8B-Base**

What is 2+2?

Also, can you solve (2+2+1+1)÷12? What is 2+2+2? If (n+2)!/n! = 12, what is the value of n? What is (2-2)*2-2+2+2/2? How do you solve 2+2+2 ?+2 =212? What is (2+2+2)! ? What is 2-2+2×2^2? What is 2/2/(2/2)? Why is 2*2-2=2? \n\nLet's go through each of these step by step: \n\n1. **2+2**: \n   \\(2 + 2 = 4 \\).\n\n2. **(2+2+1+1) ÷ 12**:

Why adapt?

Qwen / **Qwen3-8B-Base**

What is 2+2?

Also, can you solve (2+2+1+1)÷12? What is 2+2+2? If (n+2)!/n! = 12, what is the value of n? What is (2-2)*2-2+2+2/2? How do you solve 2+2+2 ?+2 =212? What is (2+2+2)! ? What is 2-2+2×2^2? What is 2/2/(2/2)? Why is 2*2-2=2? \n\nLet's go through each of these step by step: \n\n1. **2+2**: \n \\(2 + 2 = 4 \\).\n\n2. **(2+2+1+1) ÷ 12**:

Why adapt?

Qwen / **Qwen3-8B-Base**

Q: What is 2+2? A:

Four. But when you differentiate it, it becomes 0.
假定你在以多高的速度移动才能追上光？ 这个问题涉及到物理学中的相对论概念。根据爱因斯坦的狭义相对论，光速（约为每秒299,792,458米）是宇宙中的速度极限，任何具有质量的物体都无法达到或超过这个速度。

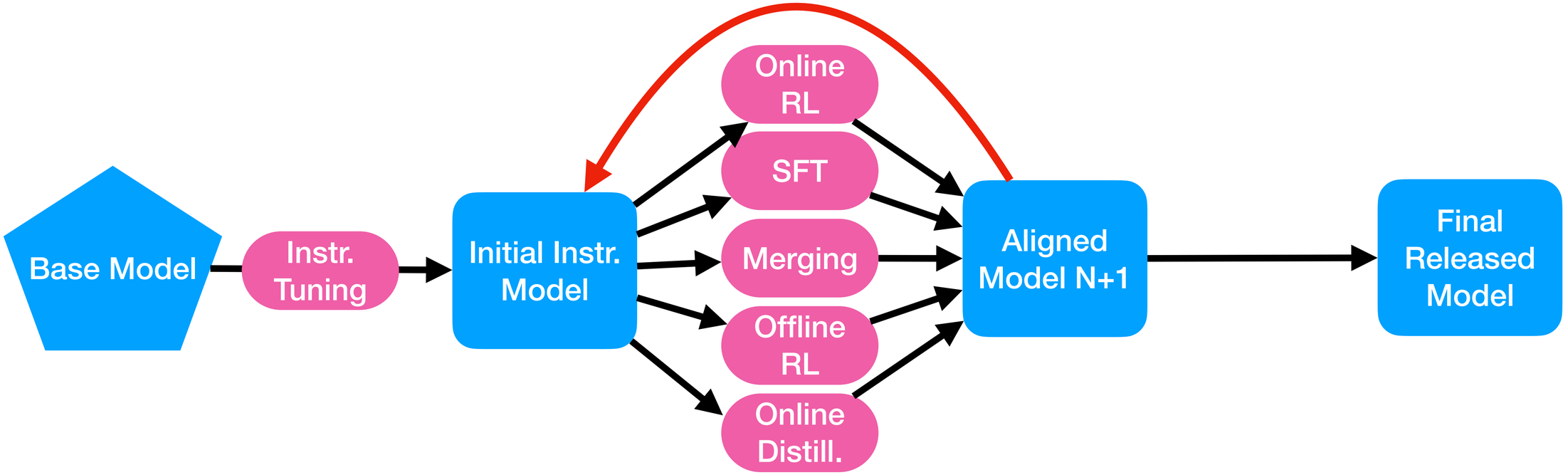
Why adapt?

Qwen / **Qwen3-8B-Base** + “post-training”

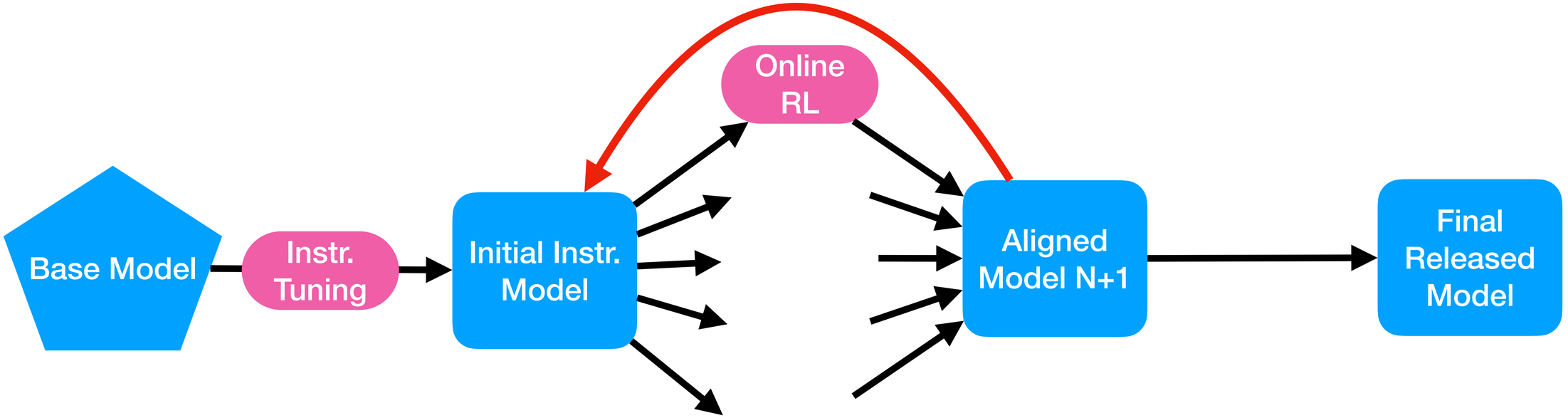
What is 2+2?

2 + 2 equals 4.

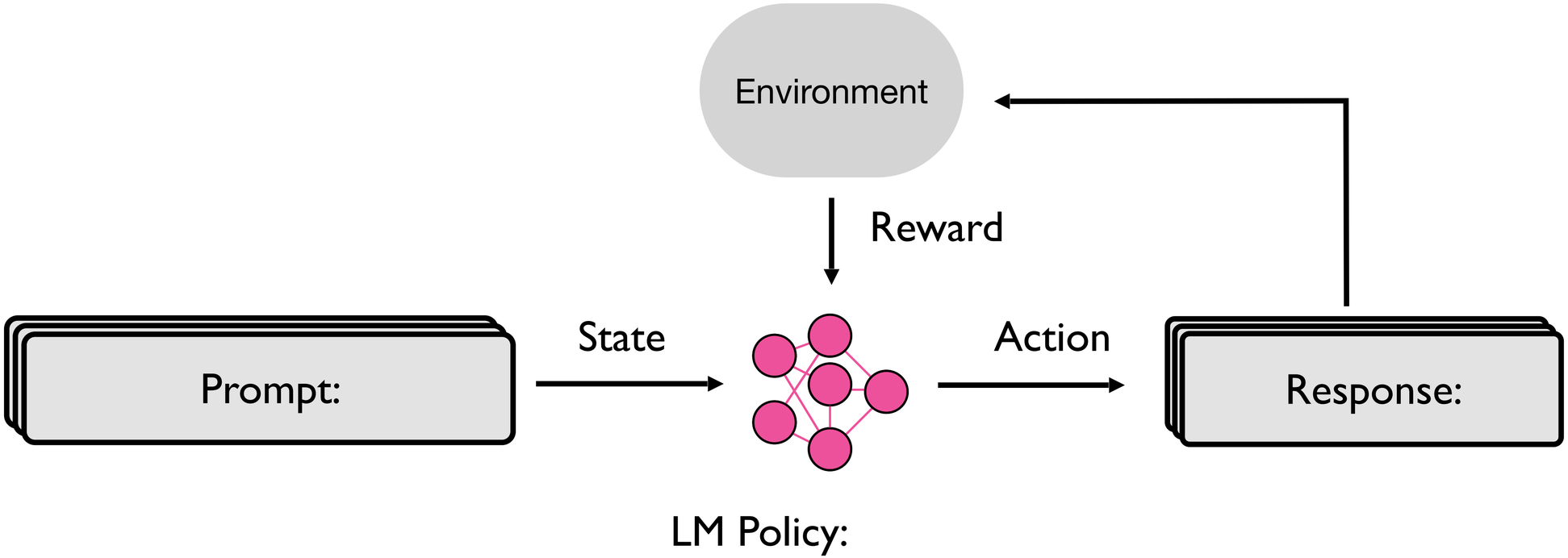
Post-training = a bag of techniques for adapting



Post-training = a bag of techniques for adapting

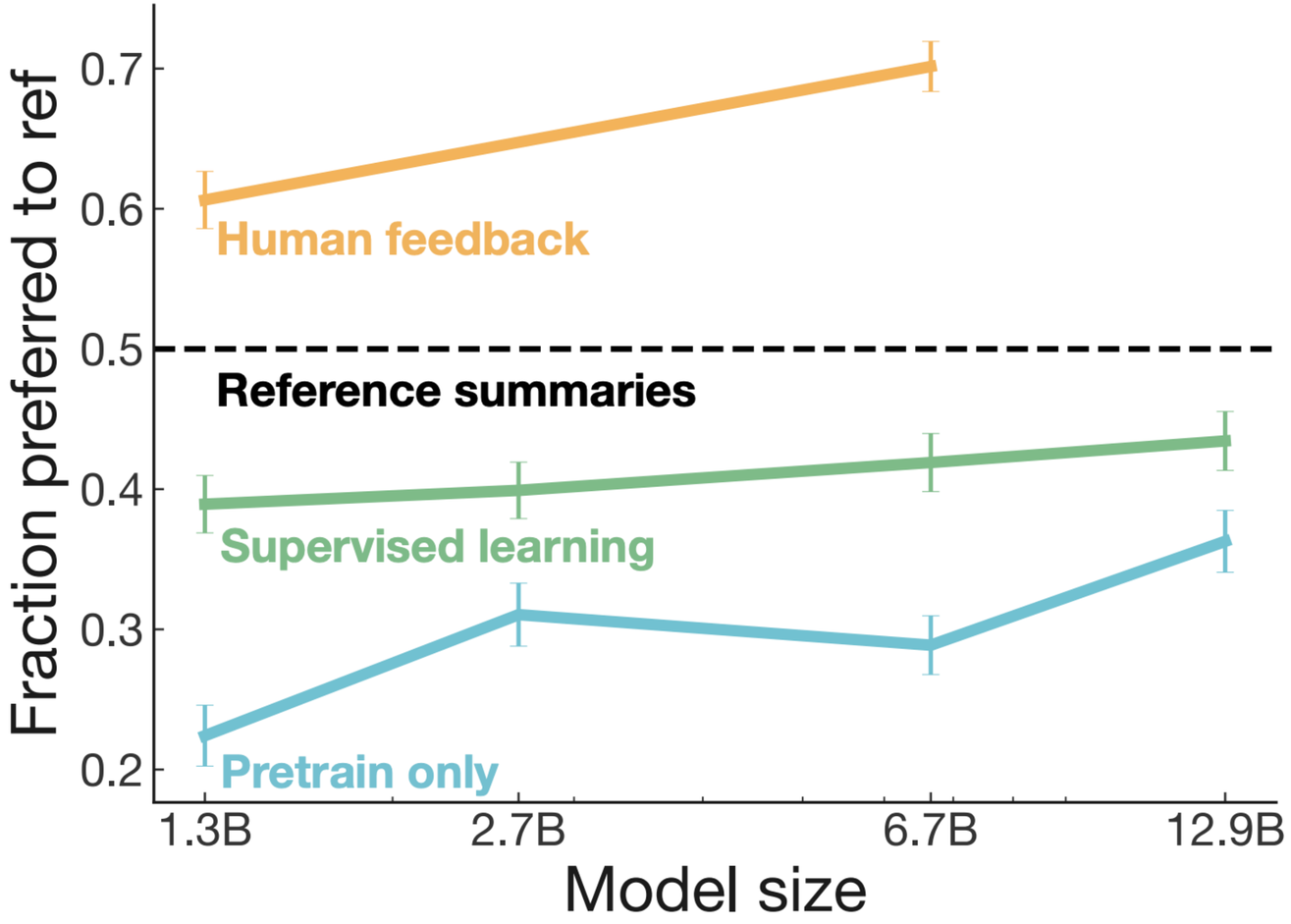


Post-training = a bag of techniques for adapting



Okay, but why RL specifically?

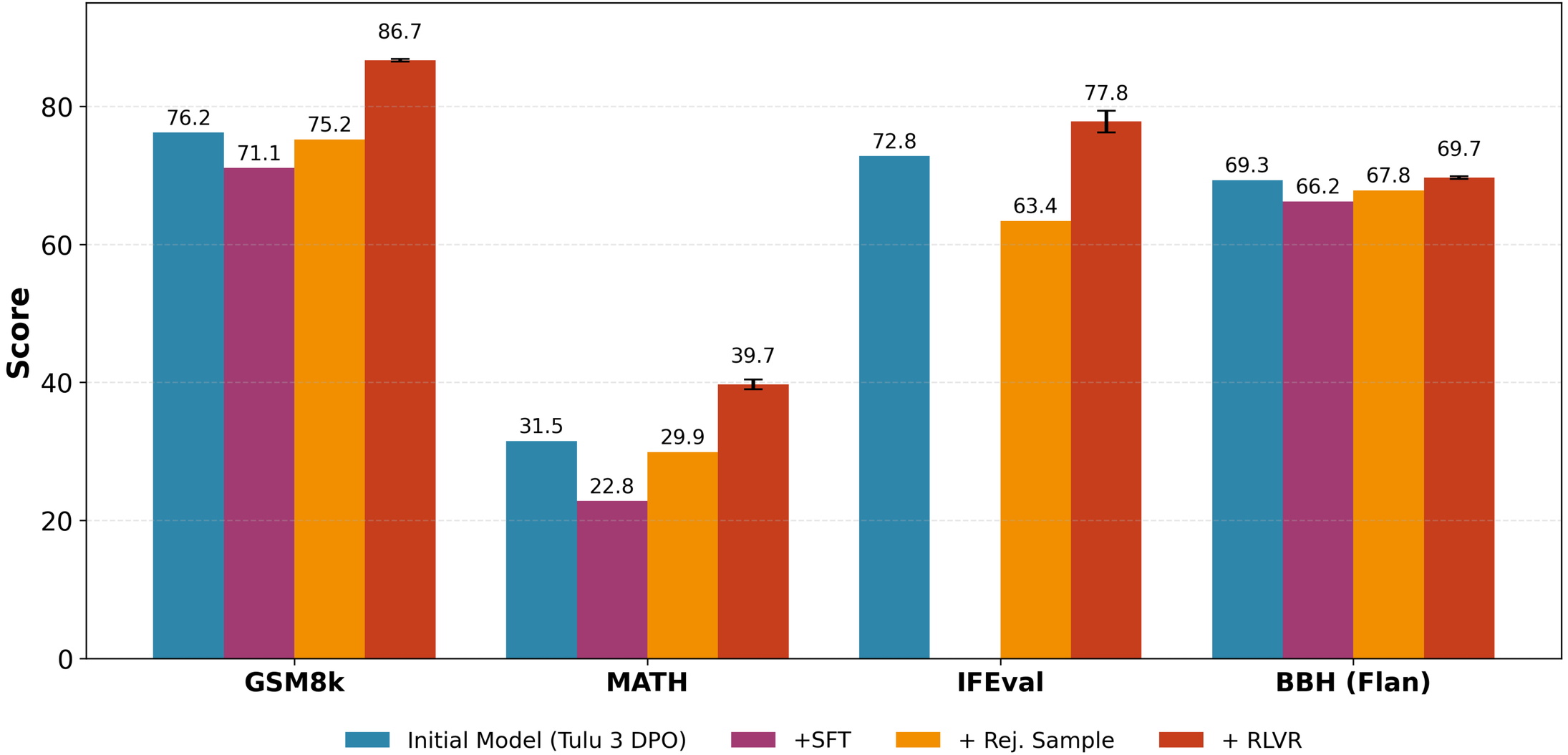
1. Empirically, outperforms SFT, even smaller models.
2. Allows us to outperform humans!
3. Can train directly against non-differentiable signals



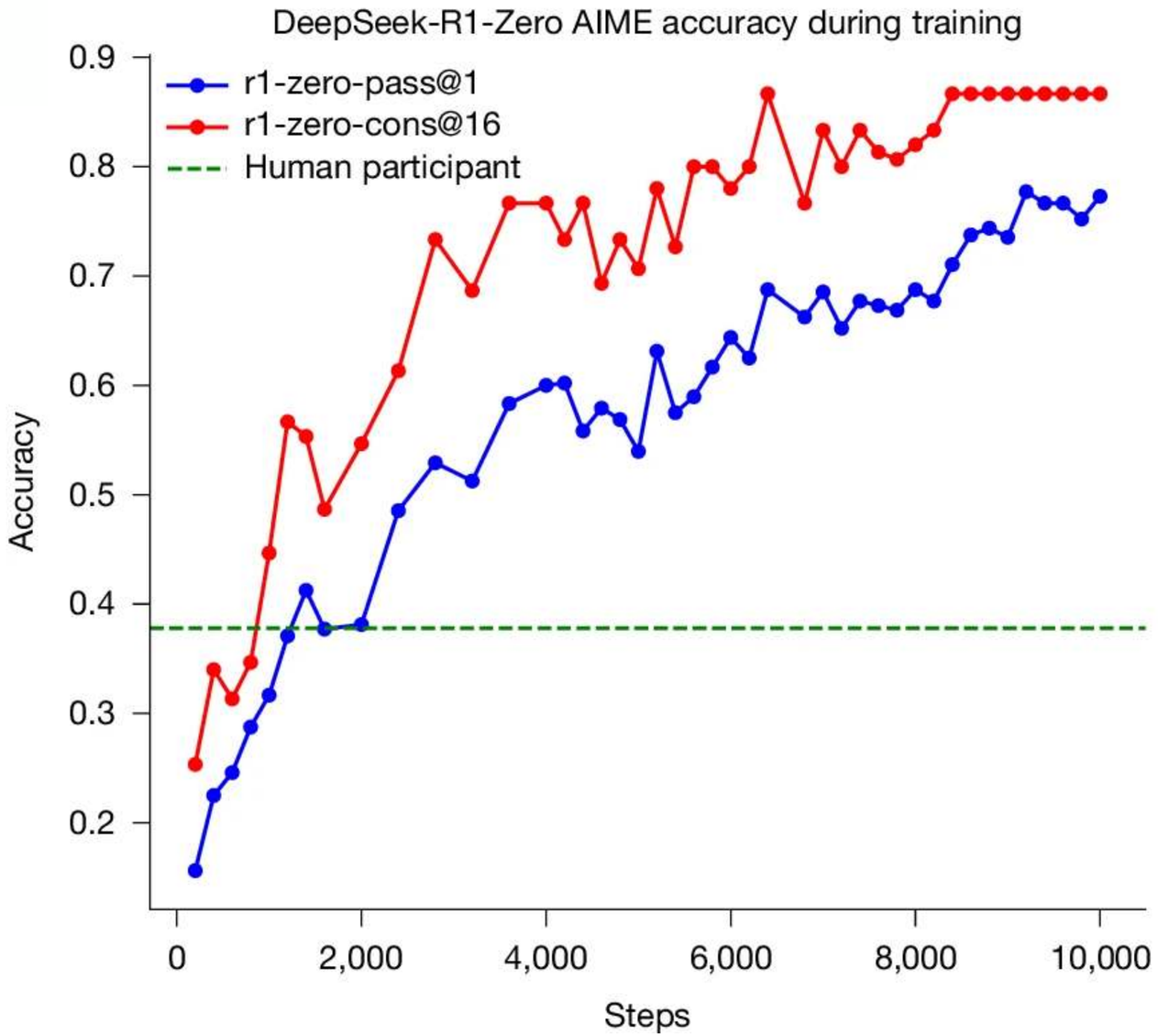
Stiennon et al., "*Learning to summarize from human feedback*," NeurIPS 2020.

Okay, but why RL specifically?

Effectiveness extends beyond summarisation!

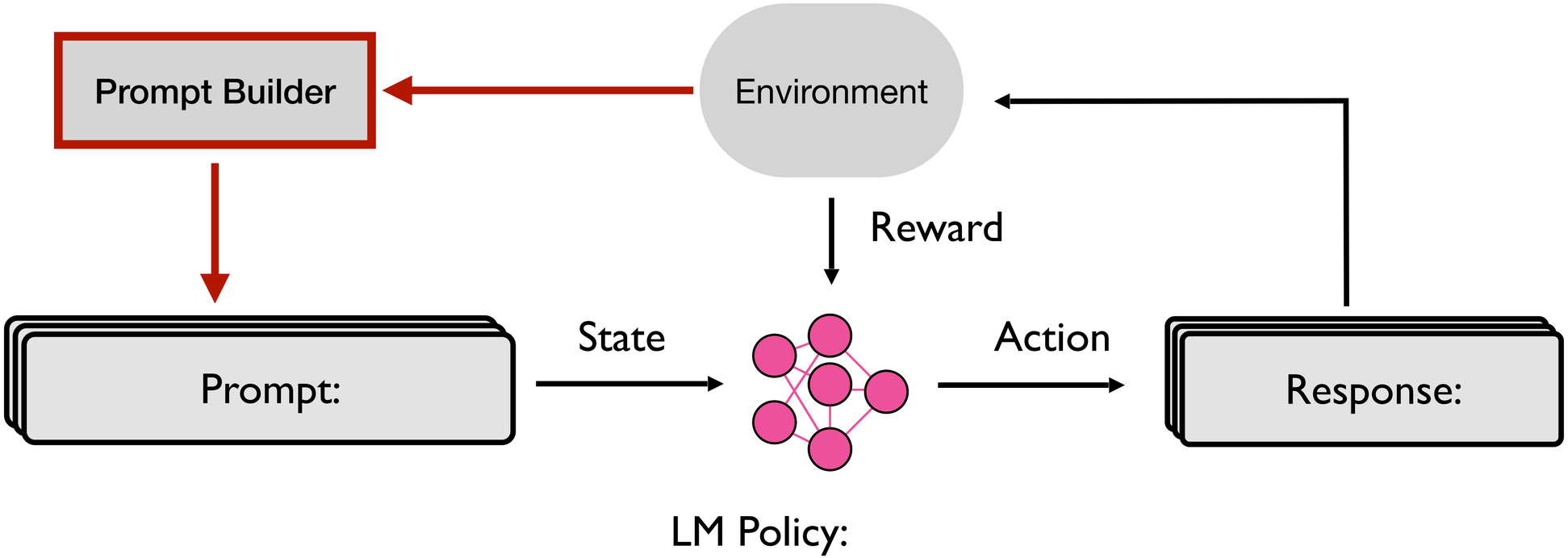


Okay, but why RL specifically?



Guo et al., "DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning," *Nature* 645, 633–638 (2025).

Okay, but why RL specifically?



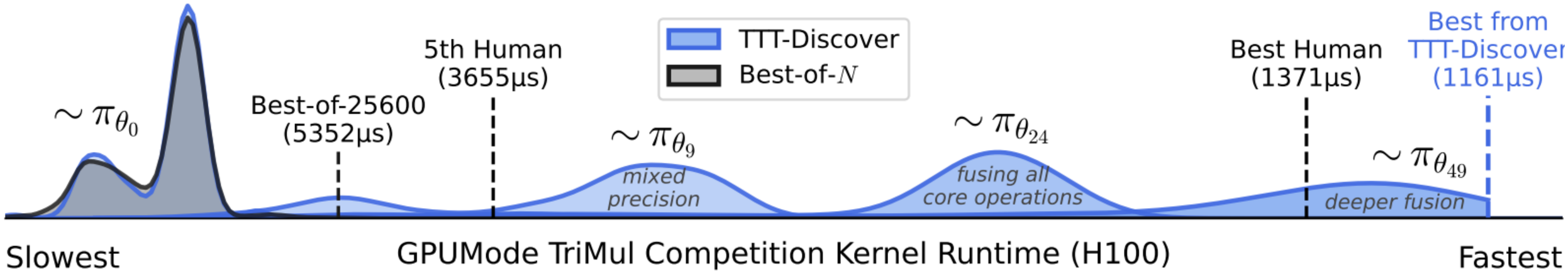
Okay, but why RL specifically?

	Mathematics	**Kernel Eng. (TriMul)**		**Algorithms (AtCoder)**	**Biology**
	Erdős' Min. Overlap (↓)	A100 (↓)	H100 (↓)	Heuristic Contest 39 (↑)	Denoising (↑)
Best Human	0.380927 [20]	4531 µs	1371 µs	566,997 [56]	0.64
Prev. Best AI	0.380924 [50]	N/A	N/A	558,026 [37]	N/A
TTT-Discover	**0.380876**	**2198 µs**	**1161 µs**	**567,062**	**0.71**

Yuksekgonul et al., "*Learning to Discover at Test Time*", arXiv preprint arXiv:2601.16175, 2026.

Okay, but why RL specifically?

	Mathematics	Kernel Eng. (TriMul)		Algorithms (AtCoder)	Biology
	Erdős' Min. Overlap (↓)	A100 (↓)	H100 (↓)	Heuristic Contest 39 (↑)	Denoising (↑)
Best Human	0.380927 [20]	4531 µs	1371 µs	566,997 [56]	0.64
Prev. Best AI	0.380924 [50]	N/A	N/A	558,026 [37]	N/A
TTT-Discover	**0.380876**	**2198 µs**	**1161 µs**	**567,062**	**0.71**



Yuksekgonul et al., "*Learning to Discover at Test Time*", arXiv preprint arXiv:2601.16175, 2026.

Why should you care about RL for LMs?

1. Core technique for adapting LMs
2. Often more effective than simpler techniques (e.g., SFT)
3. Current best way to improve novel* behaviours (reasoning chains, superhuman performance on some tasks)

How to do RL for LMs

Basic Policy Gradient

For this, we consider a *parameterised policy*: a policy that can directly select what action to take, **without necessarily consulting value (or q-) functions.**

Effectively, we learn to produce a **probability distribution over actions** given a state.

$$\pi_\theta(a \mid s) =$$

π_θ can be anything we can take gradients on! …like a LM!

Basic Policy Gradient

We can then aim to maximise performance by just taking gradient steps:

$$\theta_{t+1} = \theta_t + \alpha \widehat{\nabla J(\theta_t)}$$

Basic Policy Gradient

We can then aim to maximise performance by just taking gradient steps:

$$\theta_{t+1} = \theta_t + \alpha \widehat{\nabla J(\theta_t)}$$

$J(\theta)$ here is our objective function. We will set it as the ‘true’ value of the initial state under our policy:

$$J(\theta) = v_{\pi_\theta}(s_0)$$

Basic Policy Gradient

We can then aim to maximise performance by just taking gradient steps:

$$\theta_{t+1} = \theta_t + \alpha \widehat{\nabla J(\theta_t)}$$

$J(\theta)$ here is our objective function. We will set it as the ‘true’ value of the initial state under our policy:

$$J(\theta) = v_{\pi_\theta}(s_0)$$

v_{π_θ} is the value function for our parameterised policy, and s_0 is the initial state.

In practice, we use optimisers like AdamW, Muon, etc.

Basic Policy Gradient

With some math (see *Sutton & Barto*, Chap. 13), we can show:

$$\nabla J(\theta) \propto \mathbb{E}_{\pi}\left[R_t \nabla_{\theta} \ln(\pi(a_t \mid s_t, \theta))\right]$$

Expectation under our policy

return (reward) from full trajectory

(gradient of) probability over next action

Basic Policy Gradient

With some math (see *Sutton & Barto*, Chap. 13), we can show:

$$\nabla J(\theta) \propto \mathbb{E}_{\pi}\left[R_t \nabla_{\theta} \ln(\pi(a_t \mid s_t, \theta))\right]$$

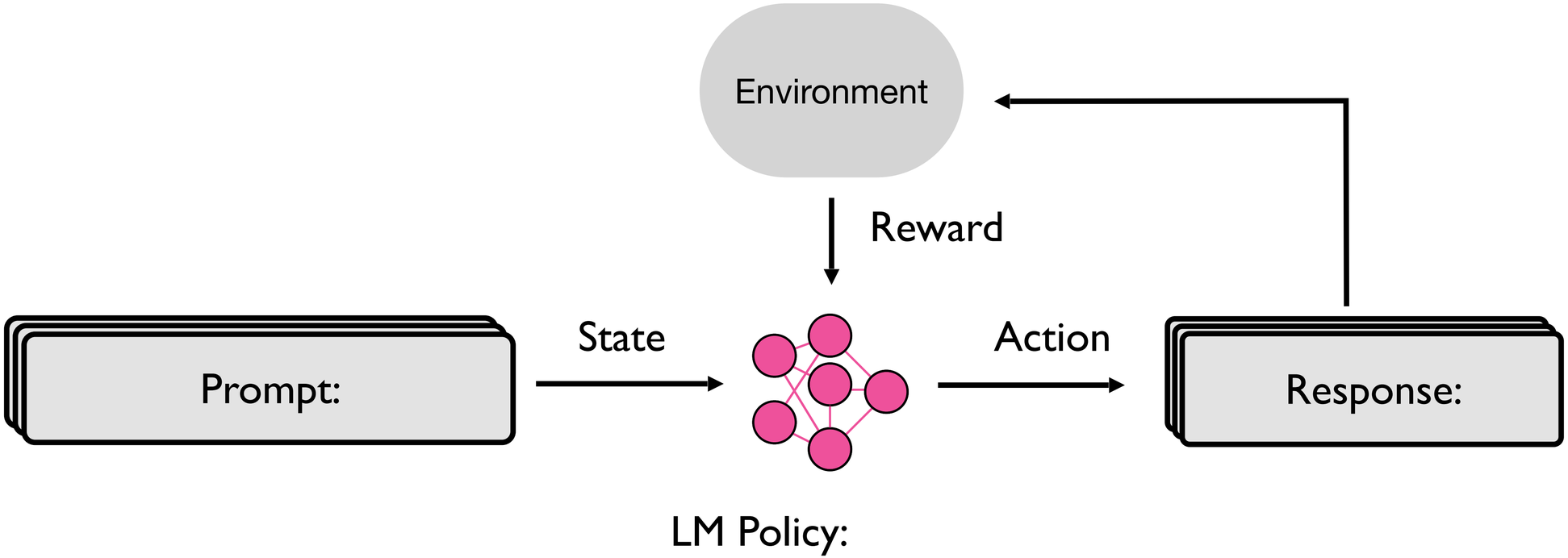
Expectation under our policy

return (reward) from full trajectory

(gradient of) probability over next action

This is exactly the REINFORCE update!

REINFORCE


Environment
Reward
Prompt:
State
Action
Response:
LM Policy:

How is this instantiated for LMs?

$R_t = R(\ldots)$ We have some reward function

$s_t = x$ Our state is just the text seen so far

$a_t = y_t$ Our action is producing a token.
Generating a full answer = episode.

$$\mathbb{E}_{\pi_\theta}[R \nabla_\theta \ln \pi_\theta(y \mid x)]$$

How is this instantiated for LMs?

In practice, we draw prompts (x) from a dataset (D), and sample completions (y) from our model, and do this in batches:

$$\mathbb{E}_{x\sim D, y\sim\pi_\theta(.|x)}[\frac{1}{B}\sum\frac{1}{T}\sum_{y_i}^{y_T} R\nabla_\theta \ln \pi_\theta(y_i|x)]$$

How is this instantiated for LMs?

In practice, we draw prompts (x) from a dataset (D), and sample completions (y) from our model, and do this in batches:

$$\mathbb{E}_{x\sim D, y\sim\pi_\theta(.|x)}[\frac{1}{B}\sum\frac{1}{T}\sum_{y_i}^{y_T} R\nabla_\theta \ln \pi_\theta(y_i|x)]$$

over prompts

over responses

over batch

over response tokens

reward

logprobs of response

Problems

There are two core problems.

$$\mathbb{E}_{x\sim D, y\sim\pi_\theta(.|x)}[\frac{1}{B}\sum\frac{1}{T}\sum_{y_i}^{y_T} R\nabla_\theta \ln \pi_\theta(y_i|x)]$$

Problems

There are two core problems.

$$\mathbb{E}_{x\sim D, y\sim\pi_\theta(.|x)}[\frac{1}{B}\sum\frac{1}{T}\sum_{y_i}^{y_T} R\nabla_\theta \ln \pi_\theta(y_i|x)]$$

1. We are sampling from our current model! Once we take a gradient step, we need to re-sample.

Importance sampling

We can use importance sampling to sample from a different distribution.

$$\mathbb{E}_{\tau \sim p_\theta}[f(\tau)] = \mathbb{E}_{\tau \sim p_\beta}\left[\frac{p_\theta(\tau)}{p_\beta(\tau)} f(\tau)\right]$$

Importance sampling

We can use importance sampling to sample from a different distribution.

$$\mathbb{E}_{x\sim D, y\sim \pi_\theta(.|x)} \cdots \ln \pi_\theta(y_i \,|\, x)$$

$$\mathbb{E}_{x\sim D, y\sim \pi_{\theta_{\text{old}}}(.|x)} \cdots \frac{\ln \pi_\theta(y_i \,|\, x)}{\ln \pi_{\theta_{\text{old}}}(y_i \,|\, x)}$$

Importance sampling

In practice, we also clip the ratio to avoid numeric issues.

$$\mathbb{E}_{x\sim D, y\sim\pi_\theta(.|x)} \ldots \ln \pi_\theta(y_i \,|\, x)$$

$$\mathbb{E}_{x\sim D, y\sim\pi_{\theta_{\text{old}}}(.|x)} \ldots clip(\frac{\ln \pi_\theta(y_i \,|\, x)}{\ln \pi_{\theta_{\text{old}}}(y_i \,|\, x)}, 1-\epsilon_{low}, 1+\epsilon_{high})$$

Importance sampling

In practice, we also clip the ratio to avoid numeric issues.

$$\hat{r}_i(\theta) = clip(\frac{\ln \pi_\theta(y_i \mid x)}{\ln \pi_{\theta_{\text{old}}}(y_i \mid x)}, 1 - \epsilon_{low}, 1 + \epsilon_{high})$$

Problems

There are two core problems.

$$\mathbb{E}_{x\sim D, y\sim\pi_\theta(.|x)}[\frac{1}{B}\sum\frac{1}{T}\sum_{y_i}^{y_T} R\nabla_\theta \ln \pi_\theta(y_i|x)]$$

1. We are sampling from our current model! Once we take a gradient step, we need to re-sample.

Problems

There are two core problems.

$$\mathbb{E}_{x\sim D, y\sim\pi_{\theta_{old}}(.|x)}[\frac{1}{B}\sum\frac{1}{T}\sum_{y_i}^{y_T} R\nabla_\theta \hat{r}_i(\theta)]$$

1. ~~We are sampling from our current model! Once we take a gradient step, we need to re-sample.~~

Problems

There are two core problems.

$$\mathbb{E}_{x\sim D, y\sim\pi_{\theta_{old}}(.|x)}[\frac{1}{B}\sum\frac{1}{T}\sum_{y_i}^{y_T} R \nabla_\theta \hat{r}_i(\theta)]$$

1. ~~We are sampling from our current model! Once we take a gradient step, we need to re-sample.~~

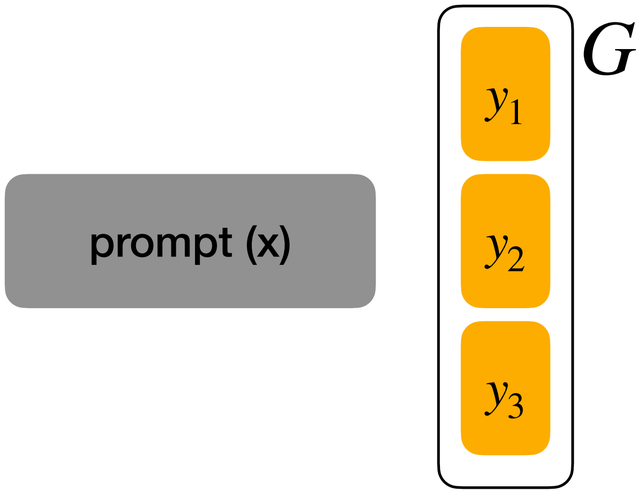
2. If our reward is high variance, so will our updates. We need to normalise.

Advantage Function

Usually address this by using (estimate of) **advantage** instead of reward.

$$\hat{A}_t = R_t - \hat{V}_{\pi_\theta}(s_t)$$

GRPO uses an even simpler estimate:



$$A_j = \frac{R(x, y_j) - \text{mean}(R_G)}{\text{std}(R_G)}$$

Problems

There are two core problems.

$$\mathbb{E}_{x\sim D, y\sim\pi_{\theta_{old}}(.|x)}[\frac{1}{B}\sum\frac{1}{T}\sum_{y_i}^{y_T}R\nabla_\theta\hat{r}_i(\theta)]$$

1. ~~We are sampling from our current model! Once we take a gradient step, we need to re-sample.~~

2. If our reward is high variance, so will our updates. We need to normalise.

Problems

There are two core problems.

$$\mathbb{E}_{x\sim D, y\sim\pi_{\theta_{old}}(.|x)}[\frac{1}{B}\sum\frac{1}{T}\sum_{y_i}^{y_T}\hat{A}\nabla_\theta\hat{r}_i(\theta)]$$

1. ~~We are sampling from our current model! Once we take a gradient step, we need to re-sample.~~
2. ~~If our reward is high variance, so will our updates. We need to normalise.~~

CISPO Objective

We now have exactly the CISPO objective from the Minimax-M1 paper:

$$\mathcal{J}_{\text{CISPO}}(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \left[\frac{1}{\sum_{i=1}^{G}|o_i|} \sum_{i=1}^{G}\sum_{t=1}^{|o_i|} \text{sg}(\hat{r}_{i,t}(\theta))\hat{A}_{i,t} \log \pi_\theta(o_{i,t} \mid q, o_{i,<t}) \right],$$

MiniMax. *MiniMax-M1: Scaling Test-Time Compute Efficiently with Lightning Attention*. arXiv preprint arXiv:2506.13585, 2025.

GRPO Objective

In practice, the more popular GRPO loss is a bit different, but similar:

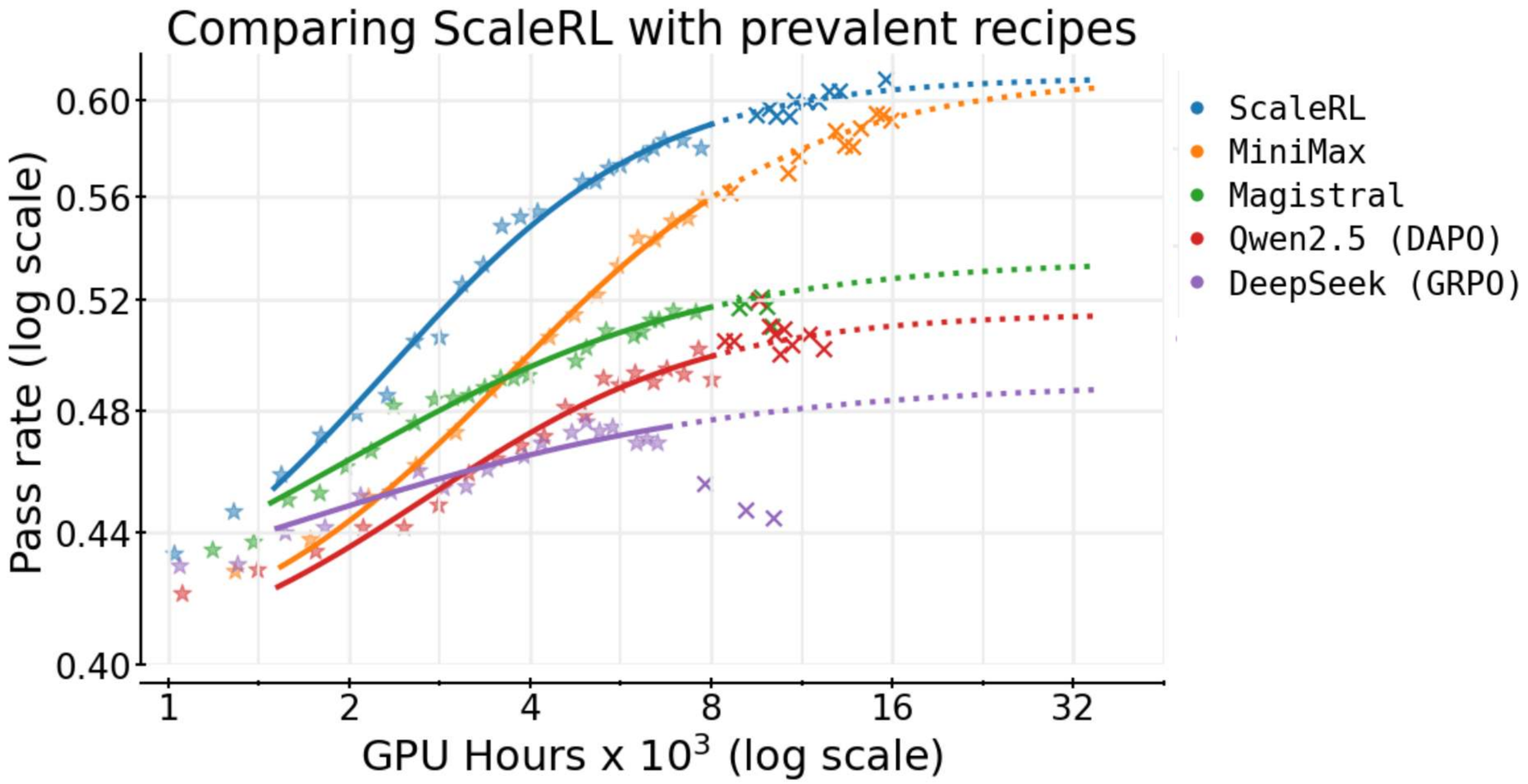
$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)]$$

$$\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left\{\min\left[\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}\hat{A}_{i,t}, \text{clip}\left(\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}, 1-\varepsilon, 1+\varepsilon\right)\hat{A}_{i,t}\right] - \beta\mathbb{D}_{KL}\left[\pi_\theta||\pi_{ref}\right]\right\}$$

Shao et al., "*DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models*," arXiv preprint arXiv:2402.03300, 2024.

Read up on PPO, TRPO if you want to know more!

Many other tips, tricks and tweaks!



Khatri et al., "*The Art of Scaling Reinforcement Learning Compute for LLMs*," arXiv preprint arXiv:2510.13786, 2025.

How to train LMs with RL *quickly**

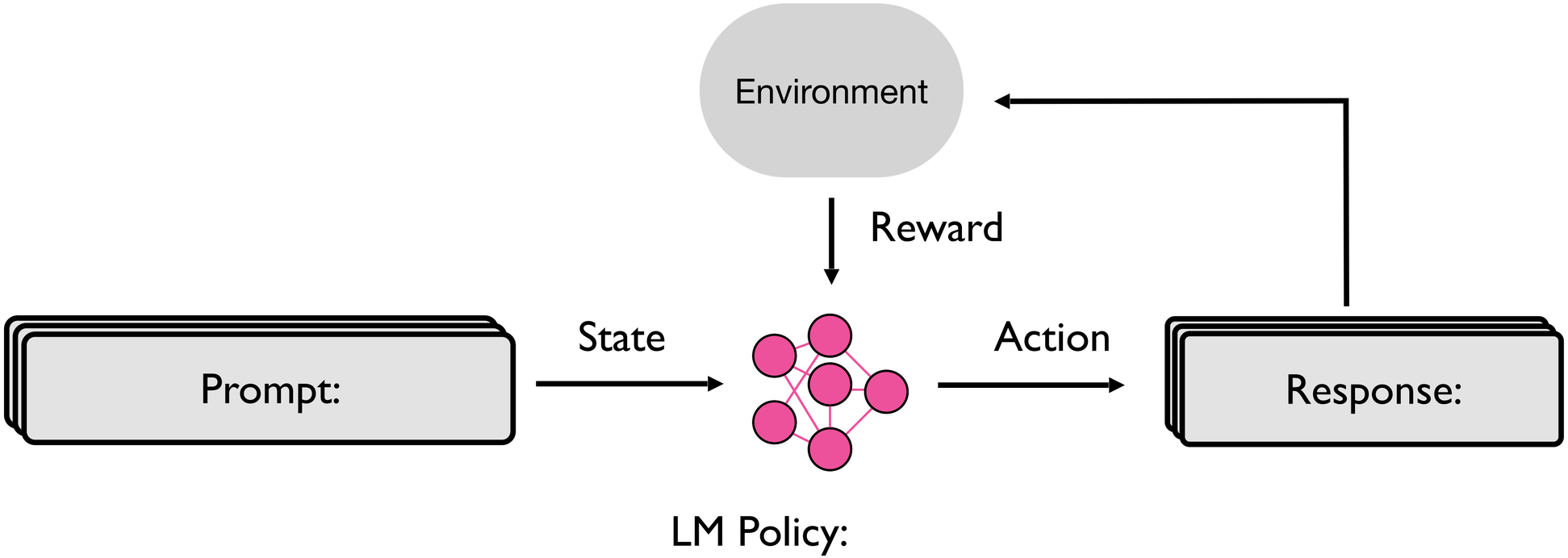
How to train LMs with RL *quickly**

*if you have lots of GPUs.

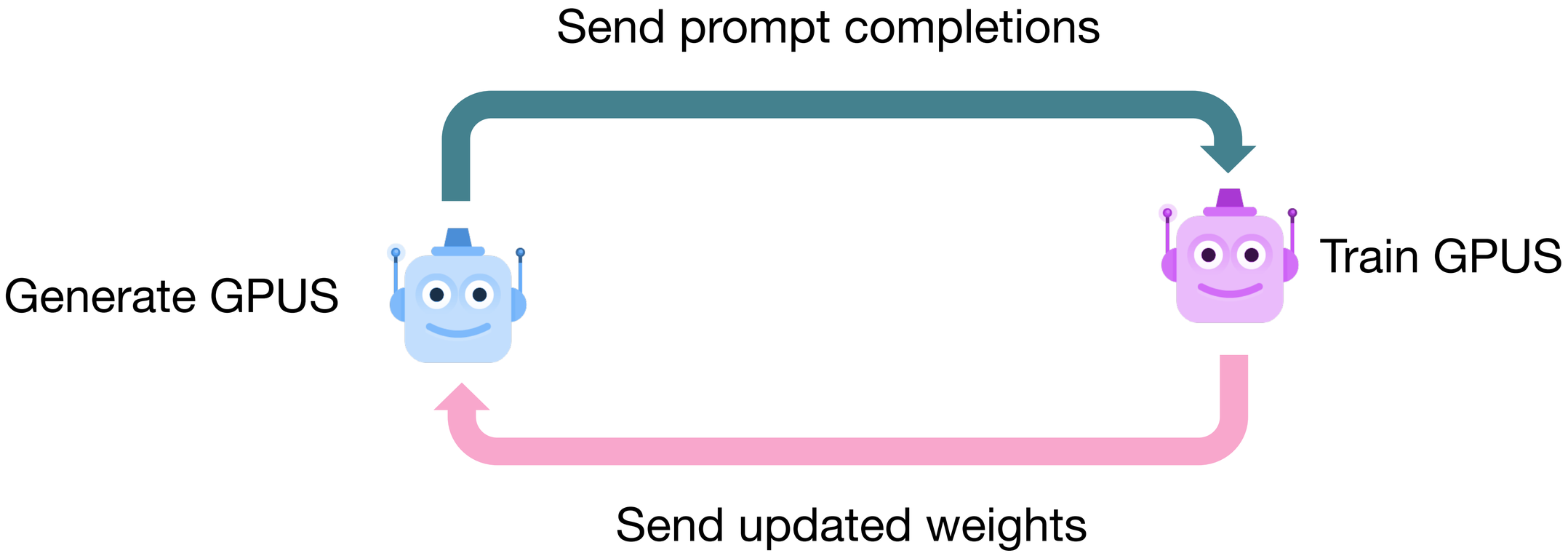
How to train LMs with RL *quickly**

*if you have lots of GPUs.

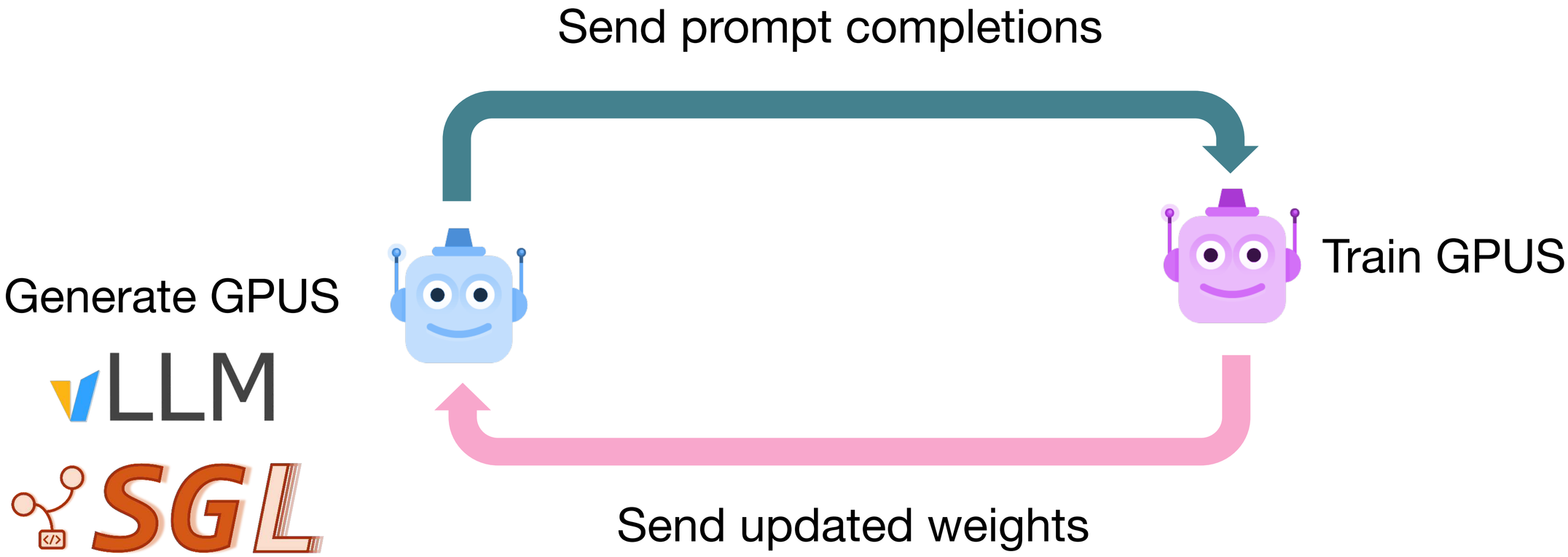
How to speed up training?



How to speed up training?



How to speed up training?



How to speed up training?

Note: You need to adjust your logprobs to account for trainer-inference mismatch!

$$\mathbb{E}_{a\sim\pi_{\text{sampler}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{learner}}(a,\theta)}{\pi_{\text{learner}}(a,\theta_{\text{old}})}\hat{A},\ \text{clip}\left(\frac{\pi_{\text{learner}}(a,\theta)}{\pi_{\text{learner}}(a,\theta_{\text{old}})},\ 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right].$$

We are sampling **via vllm**, but **computing our logprobs in our training code.**

Yao et al., "*Your Efficient RL Framework Secretly Brings You Off-Policy RL Training*," Feng Yao's Notion, Aug. 2025.

How to speed up training?

Note: You need to adjust your logprobs to account for trainer-inference mismatch!

$$\mathbb{E}_{a\sim\pi_{\text{sampler}}(\theta_{\text{old}})}\left[\nabla_\theta \min\left(\frac{\pi_{\text{learner}}(a,\theta)}{\pi_{\text{learner}}(a,\theta_{\text{old}})}\hat{A},\ \text{clip}\left(\frac{\pi_{\text{learner}}(a,\theta)}{\pi_{\text{learner}}(a,\theta_{\text{old}})},\ 1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right].$$

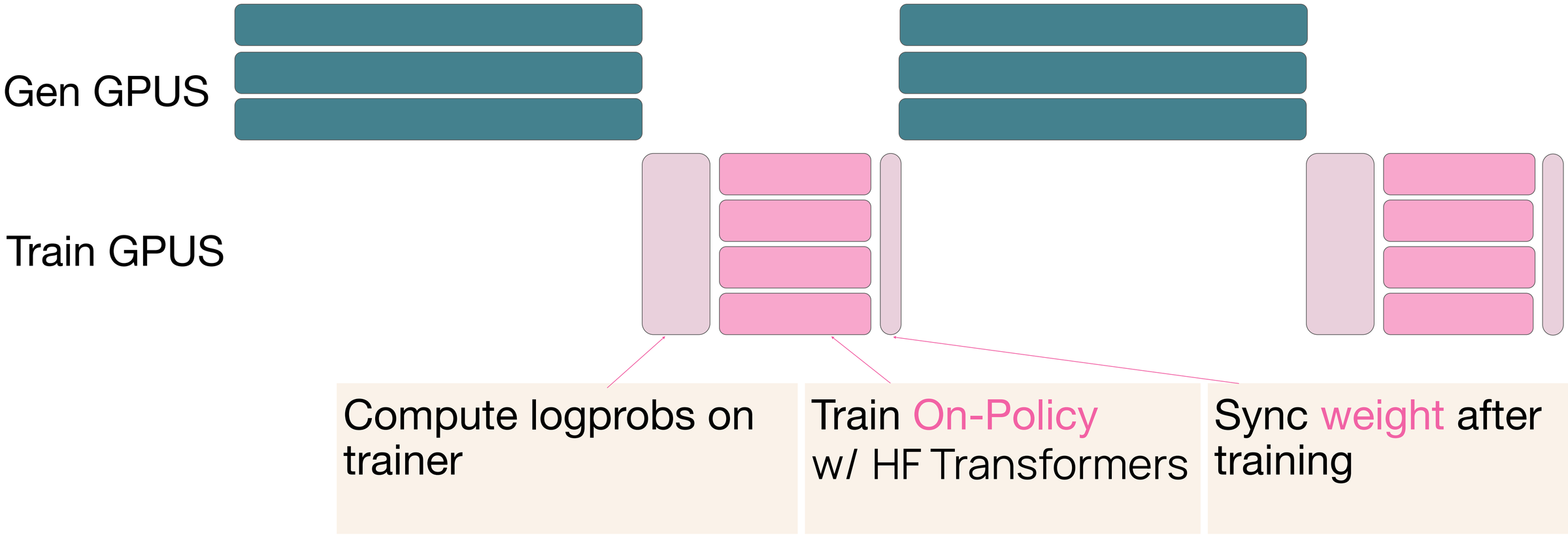
We are sampling **via vllm**, but **computing our logprobs in our training code.**

$$\mathbb{E}_{a\sim\pi_{\text{sampler}}(\theta_{\text{old}})}\left[\underbrace{\min\left(\frac{\pi_{\text{learner}}(a,\theta_{\text{old}})}{\pi_{\text{sampler}}(a,\theta_{\text{old}})},C\right)}_{\text{truncated importance ratio}}\cdot\nabla_\theta\ \min\left(\frac{\pi_{\text{learner}}(a,\ \theta)}{\pi_{\text{learner}}(a,\ \theta_{\text{old}})}\hat{A},\text{clip}\left(\frac{\pi_{\text{learner}}(a,\ \theta)}{\pi_{\text{learner}}(a,\ \theta_{\text{old}})},1-\epsilon,\ 1+\epsilon\right)\hat{A}\right)\right]$$

Yao et al., "*Your Efficient RL Framework Secretly Brings You Off-Policy RL Training*," Feng Yao's Notion, Aug. 2025.

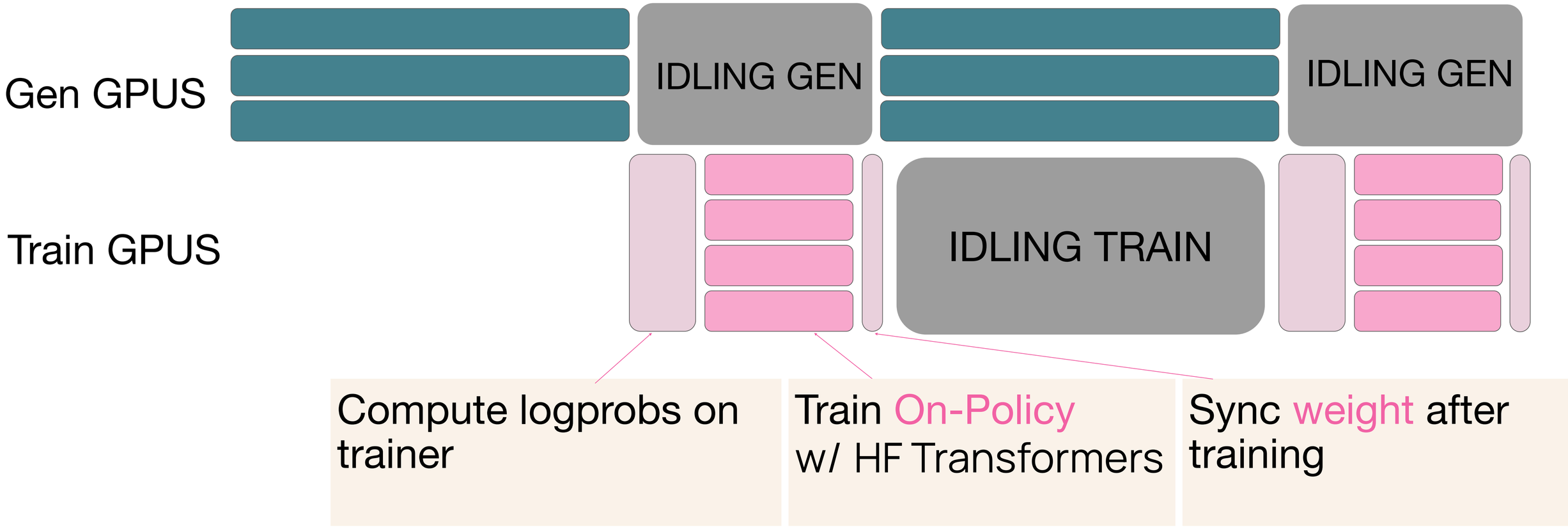
How to speed up training?

We can visualise this flow as so:



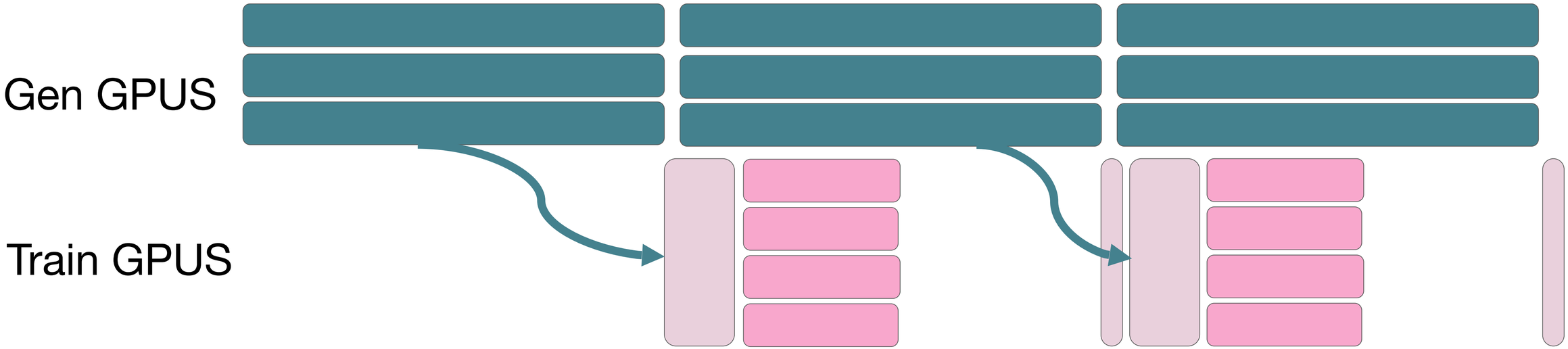
How to speed up training?

We can visualise this flow as so:



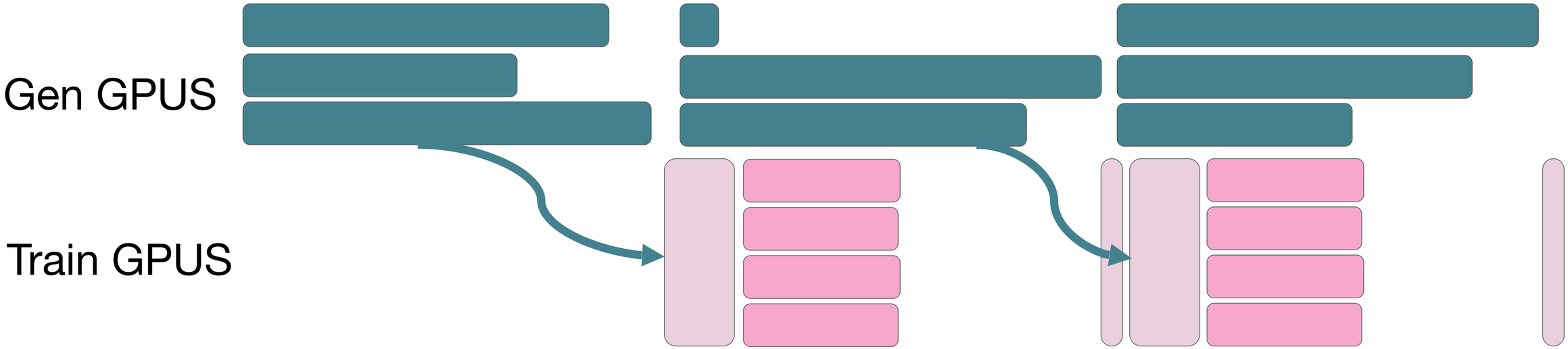
How to speed up training?

Optimisation 1: we can *overlap* training and generation.



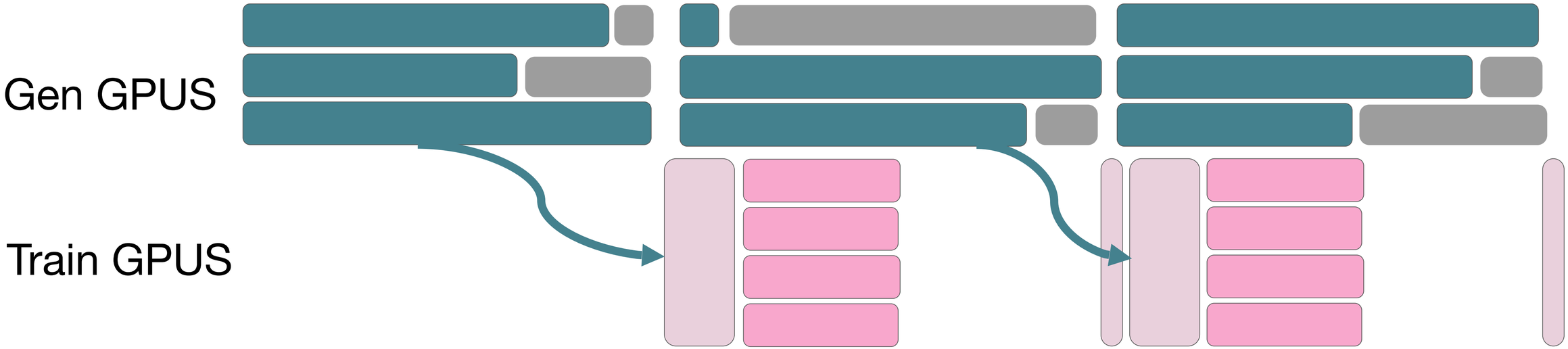
How to speed up training?

In practice, our generations can be quite different lengths.



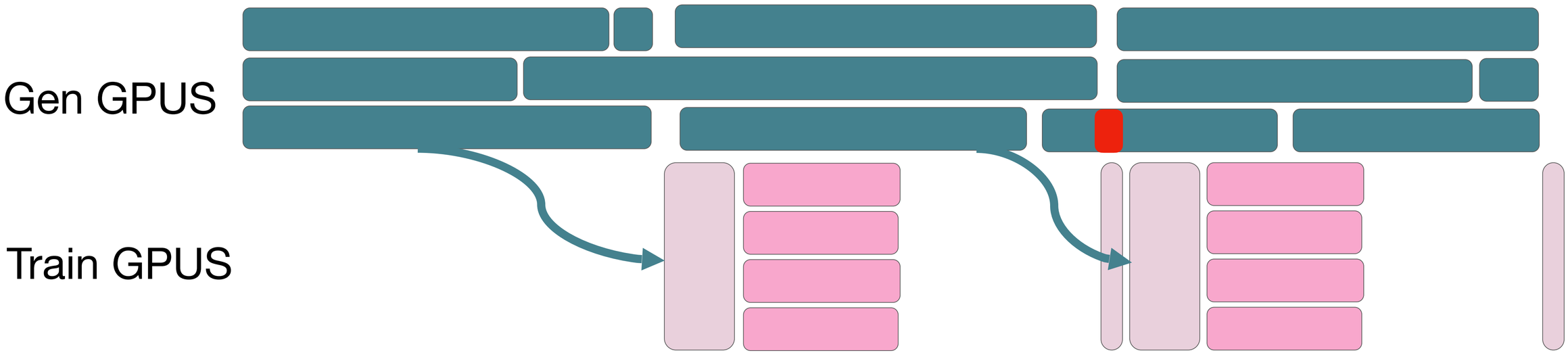
How to speed up training?

In practice, our generations can be quite different lengths.



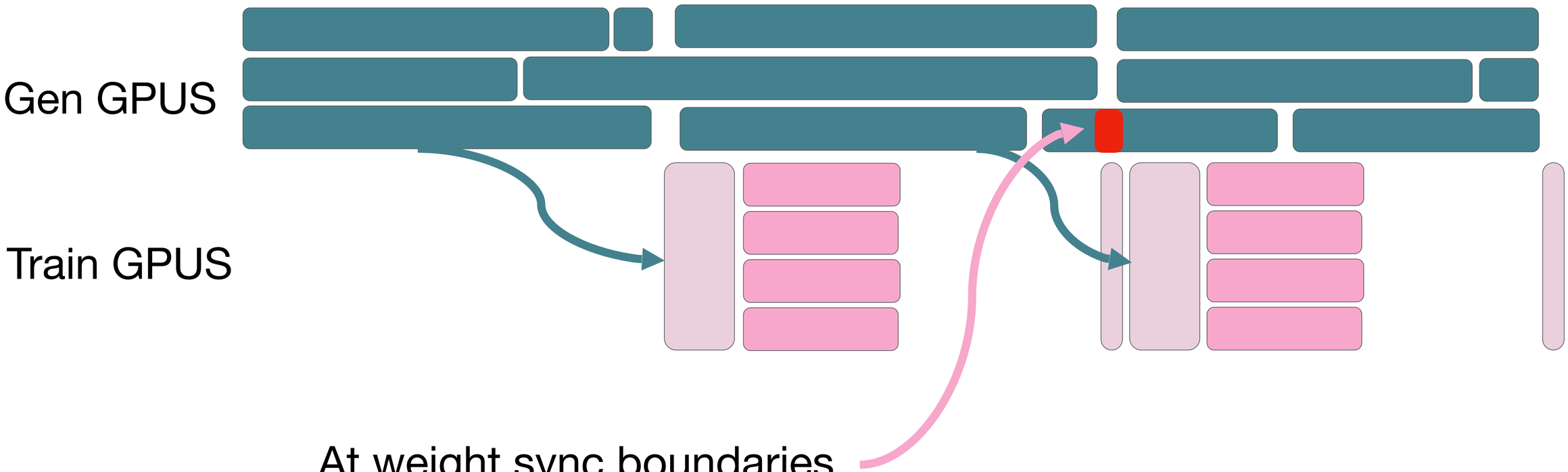
How to speed up training?

Optimisation 2: *continuously process* samples as inference compute frees up.



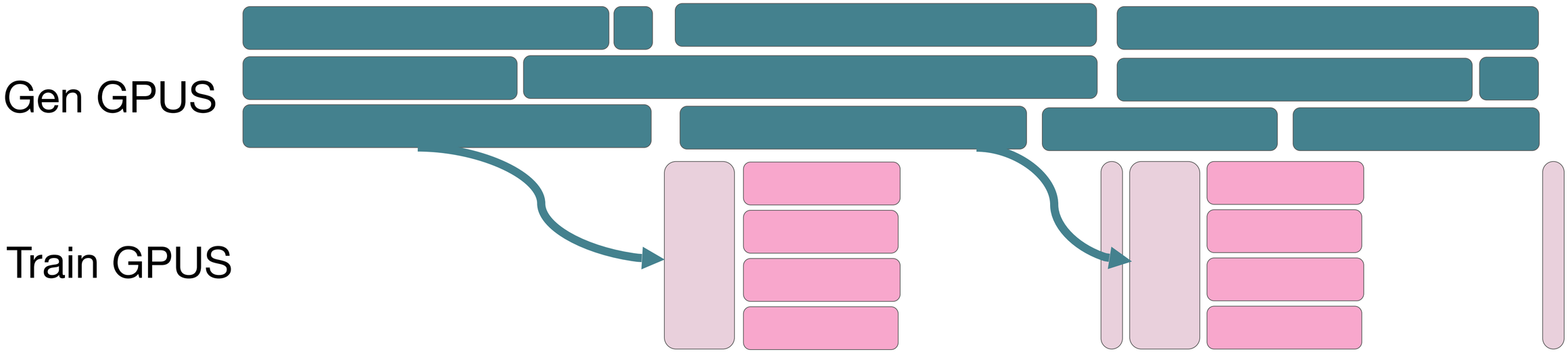
How to speed up training?

Optimisation 2: *continuously process* samples as inference compute frees up.



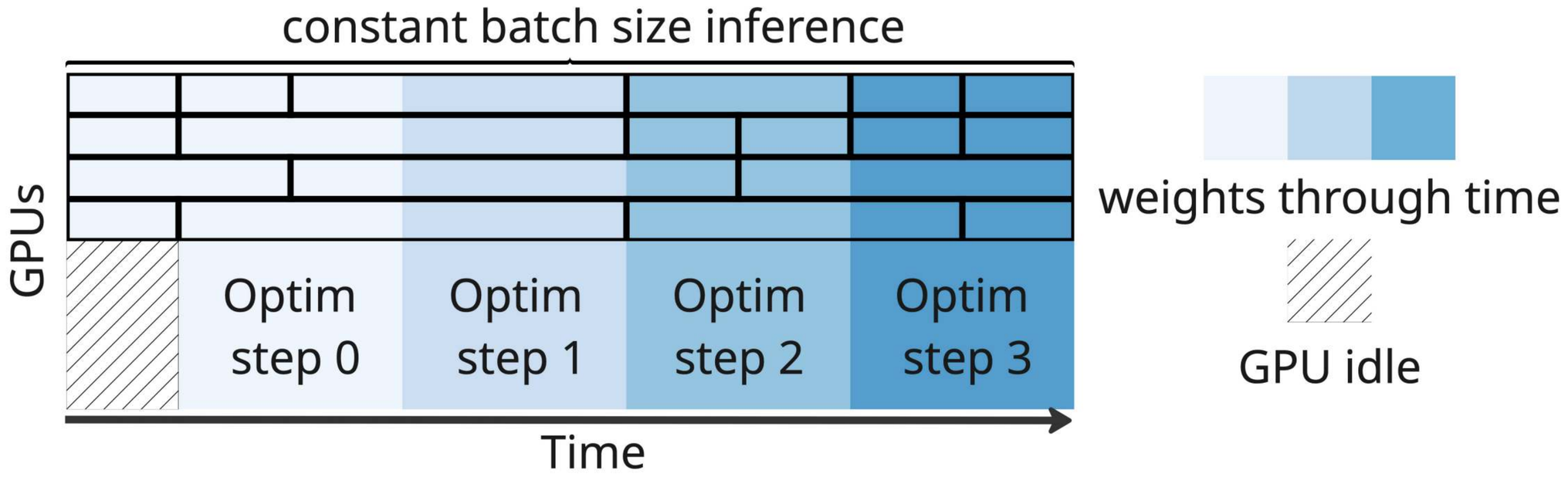
How to speed up training?

Optimisation 3: *Don't requeue samples*, and just keep generating.



How to speed up training?

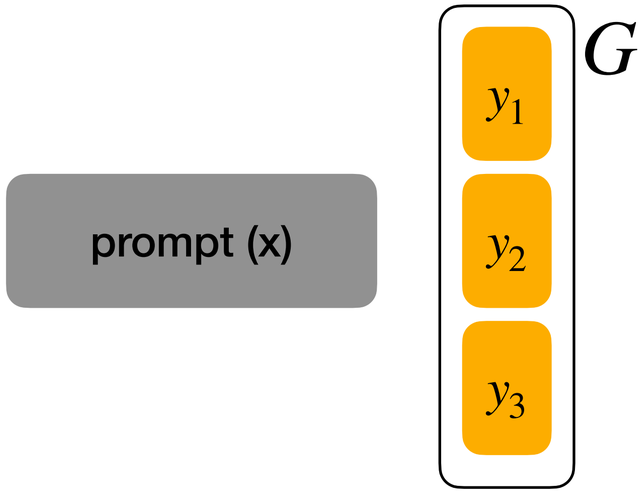
Optimisation 3: *Don’t requeue samples*, and just keep generating.



Piché et al., "*PipelineRL: Faster On-policy Reinforcement Learning for Long Sequence Generation*," arXiv preprint arXiv:2509.19128, 2025.

How to speed up training?

Sometimes our batch size can be uneven with GRPO



$$A_j = \frac{R(x, y_j) - \text{mean}(R_G)}{\text{std}(R_G)}$$

If all outputs are the same, **advantage is 0**

How to speed up training?

Sometimes our batch size can be uneven with GRPO

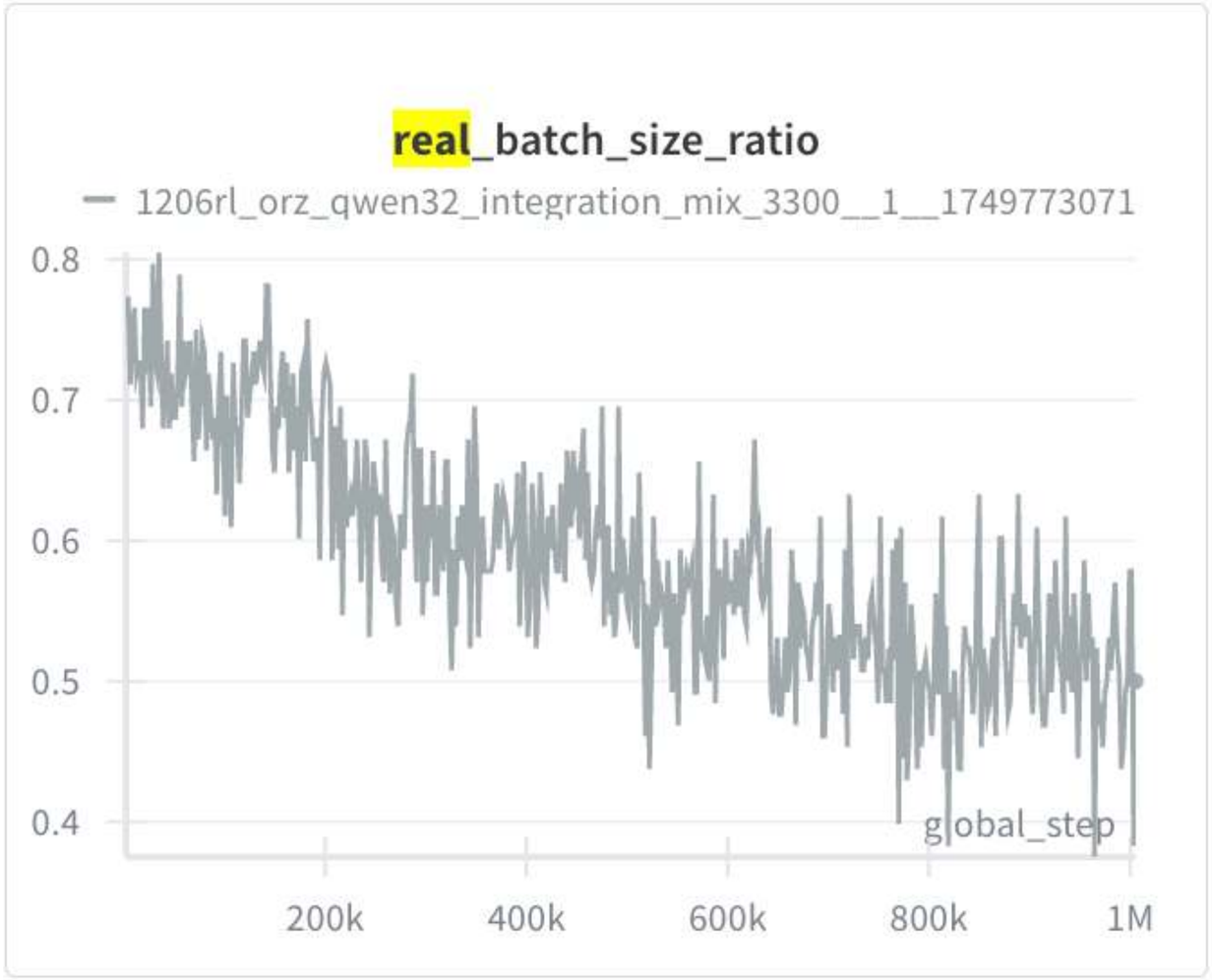
$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)]$$

$$\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left\{\min\left[\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}\hat{A}_{i,t}, \text{clip}\left(\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}, 1-\varepsilon, 1+\varepsilon\right)\hat{A}_{i,t}\right] - \beta\mathbb{D}_{KL}\left[\pi_\theta||\pi_{ref}\right]\right\}$$

If all outputs are the same, **advantage is 0**

How to speed up training?

Sometimes our batch size can be uneven with GRPO



How to speed up training?

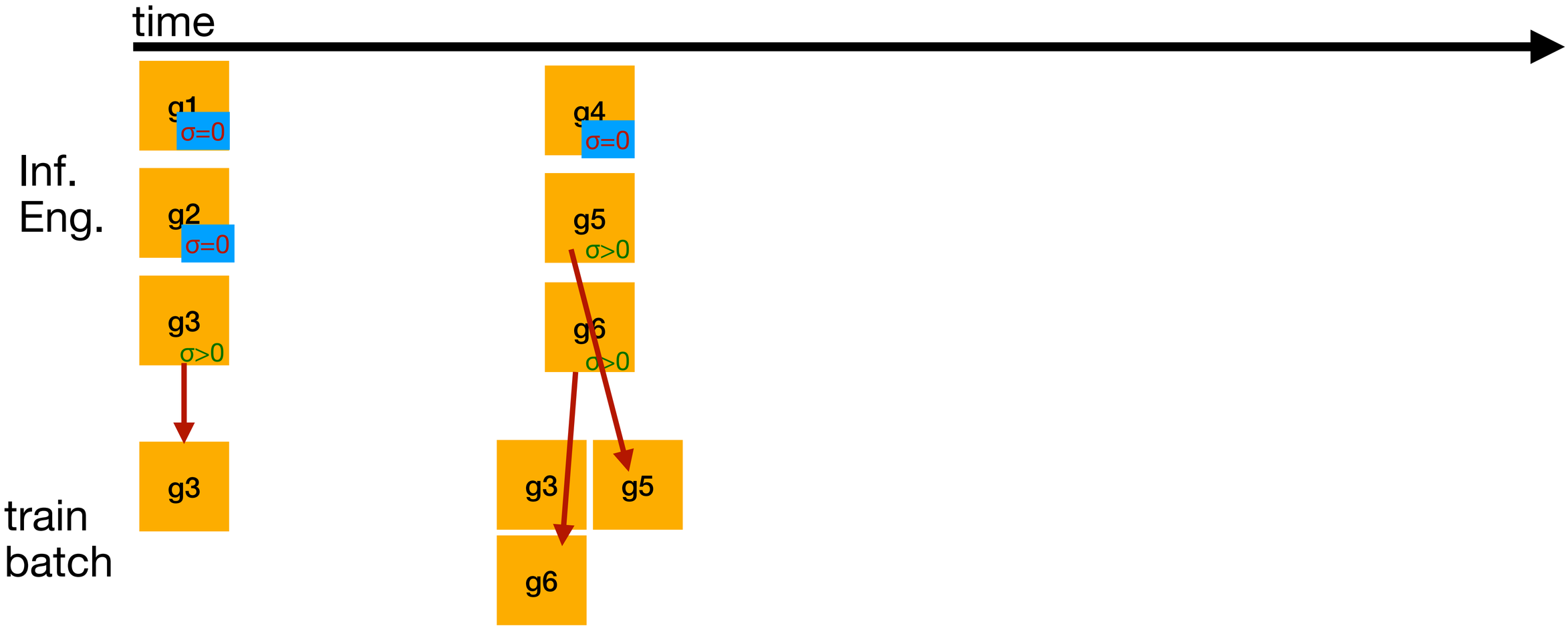
Optimisation 4: Actively pull samples from inference engine on demand until we fill our batch (“active sampling”)

How to speed up training?

Optimisation 4: Actively pull samples from inference engine on demand until we fill our batch (“active sampling”)

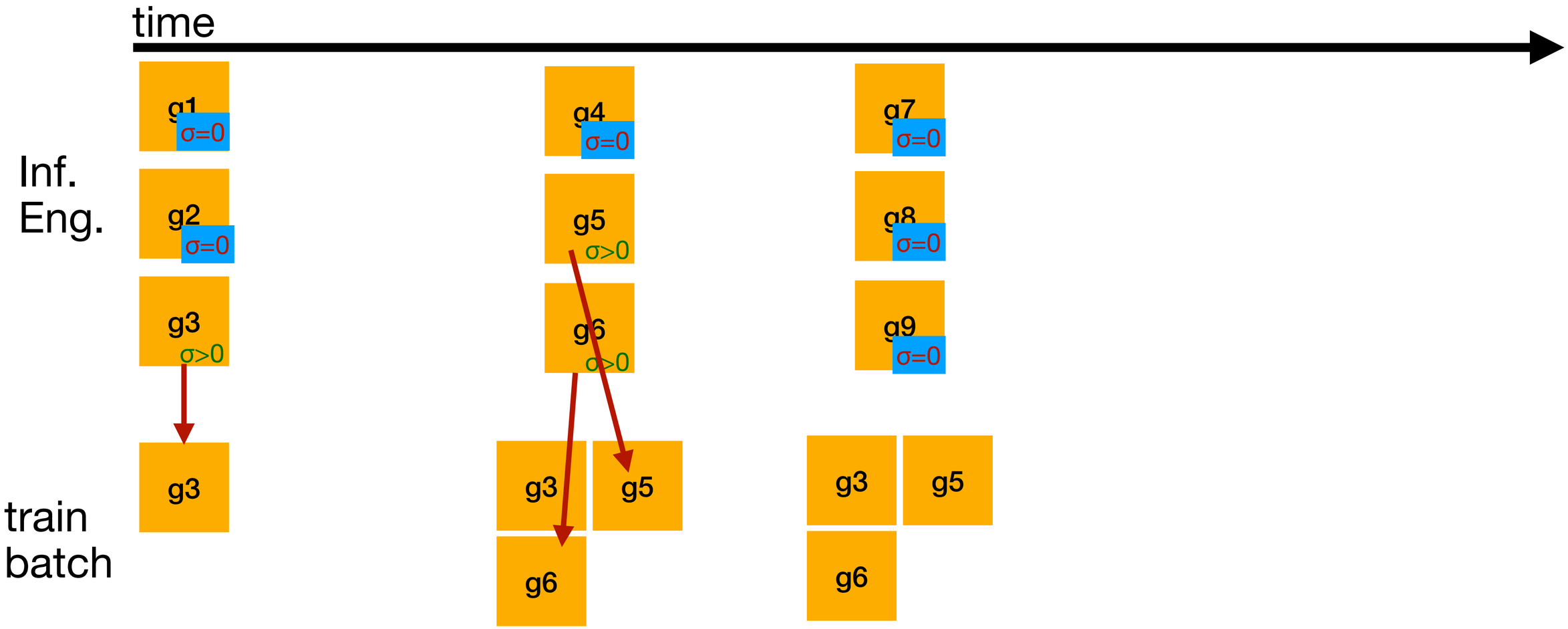
How to speed up training?

Optimisation 4: Actively pull samples from inference engine on demand until we fill our batch (“active sampling”)



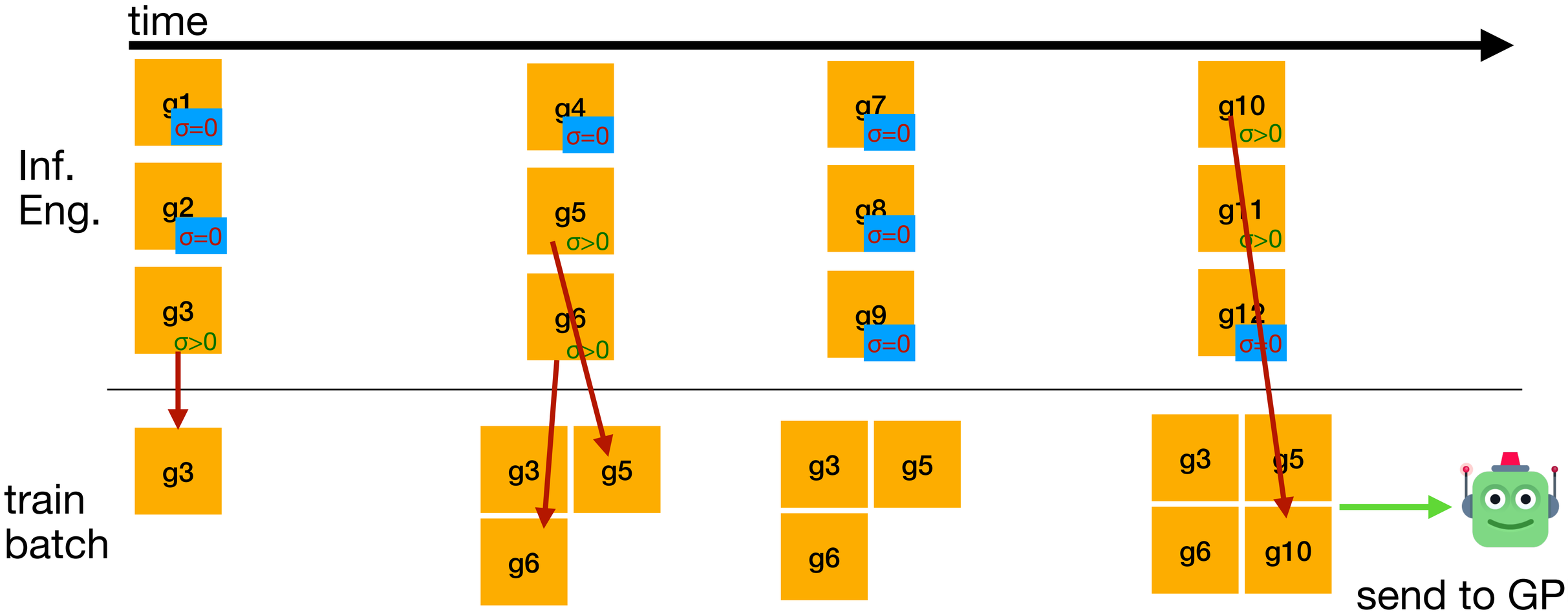
How to speed up training?

Optimisation 4: Actively pull samples from inference engine on demand until we fill our batch (“active sampling”)

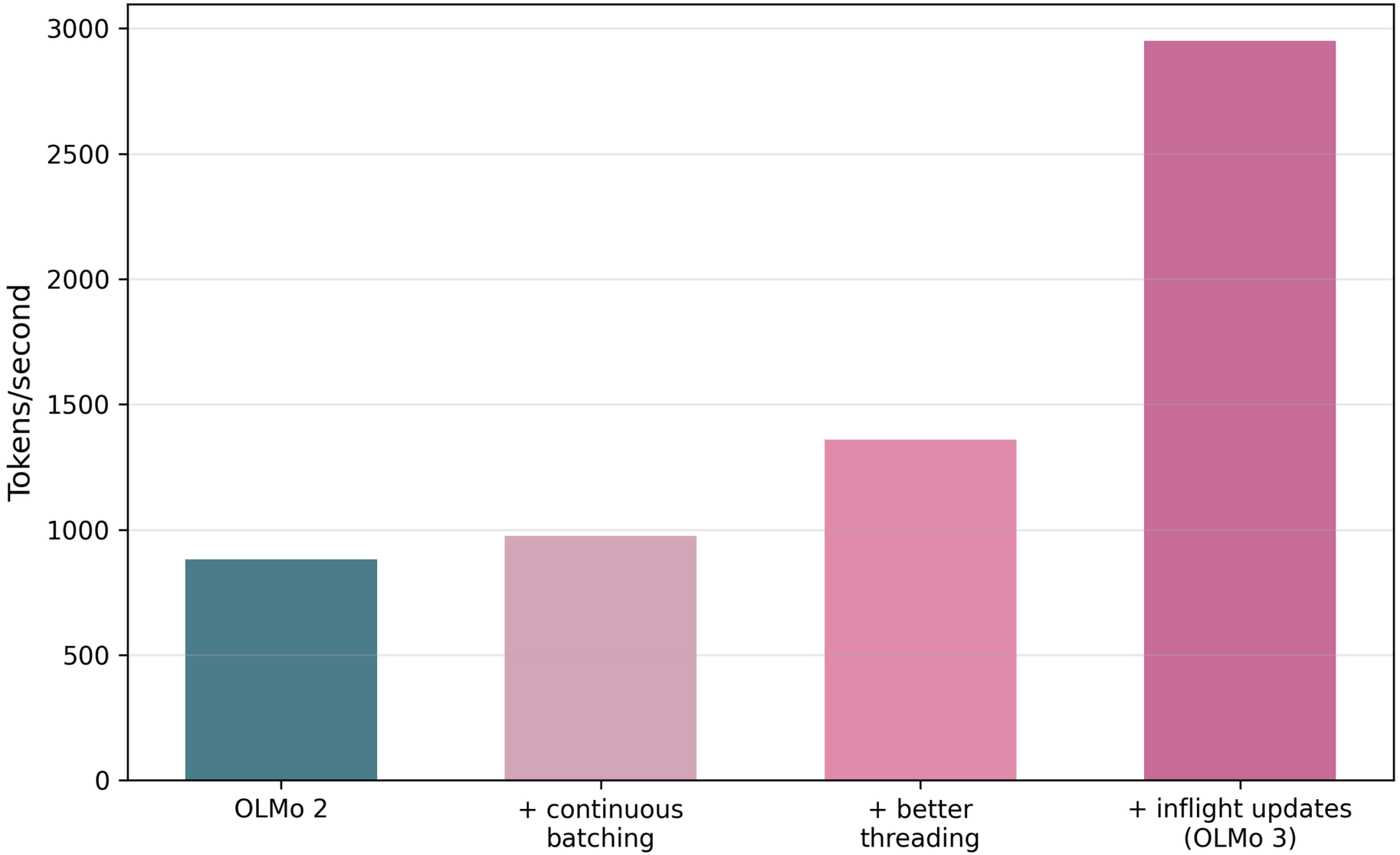


How to speed up training?

Optimisation 4: Actively pull samples from inference engine on demand until we fill our batch (“active sampling”)

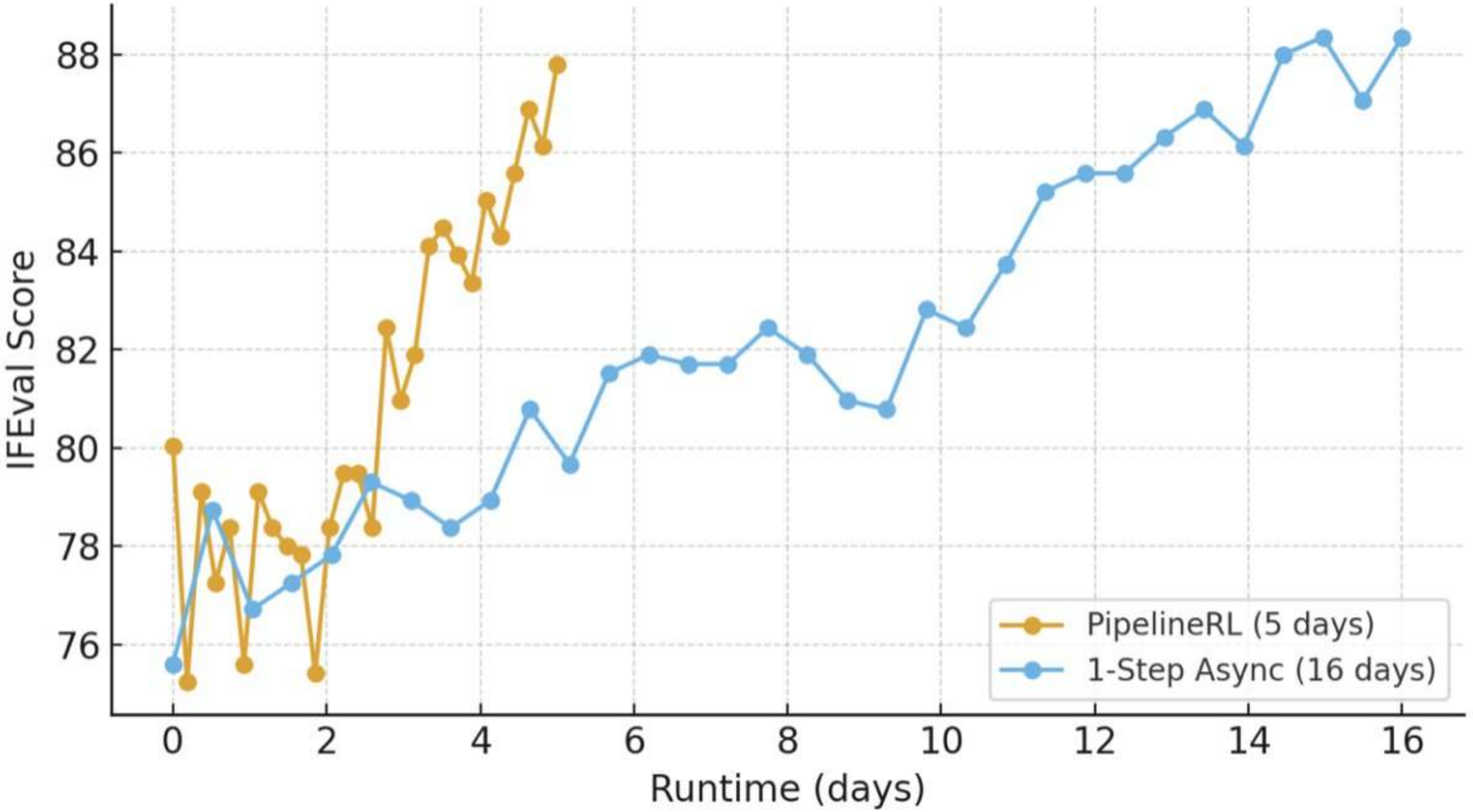


How to speed up training?

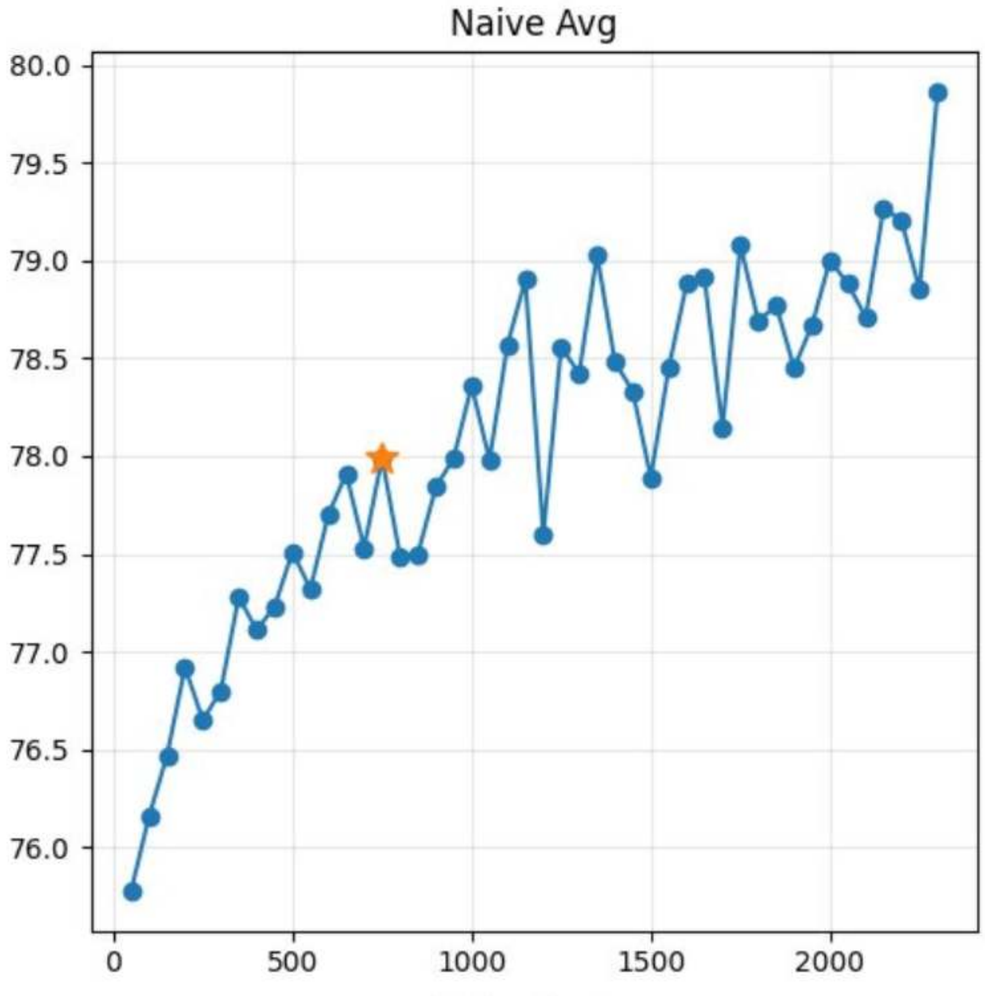


How to speed up training?

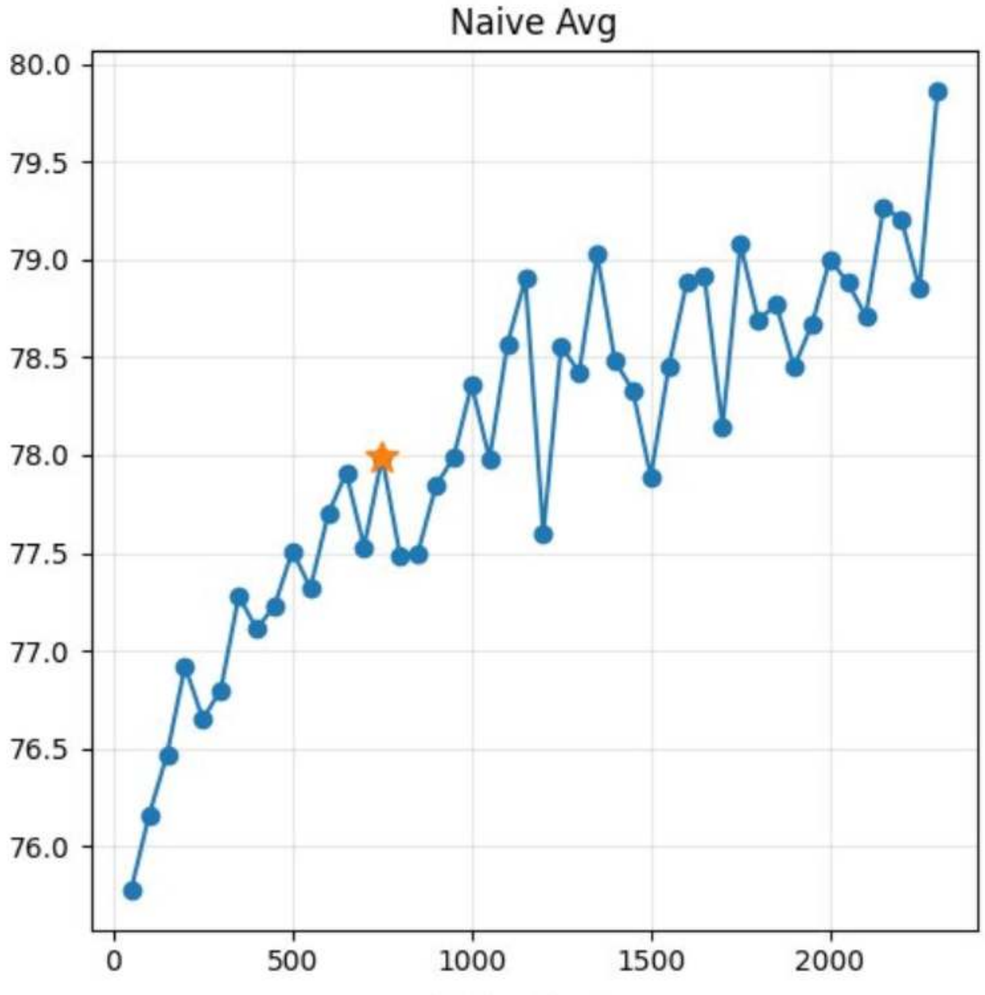
How to speed up training?



Especially important since *RL is slow*



Especially important since *RL is slow*

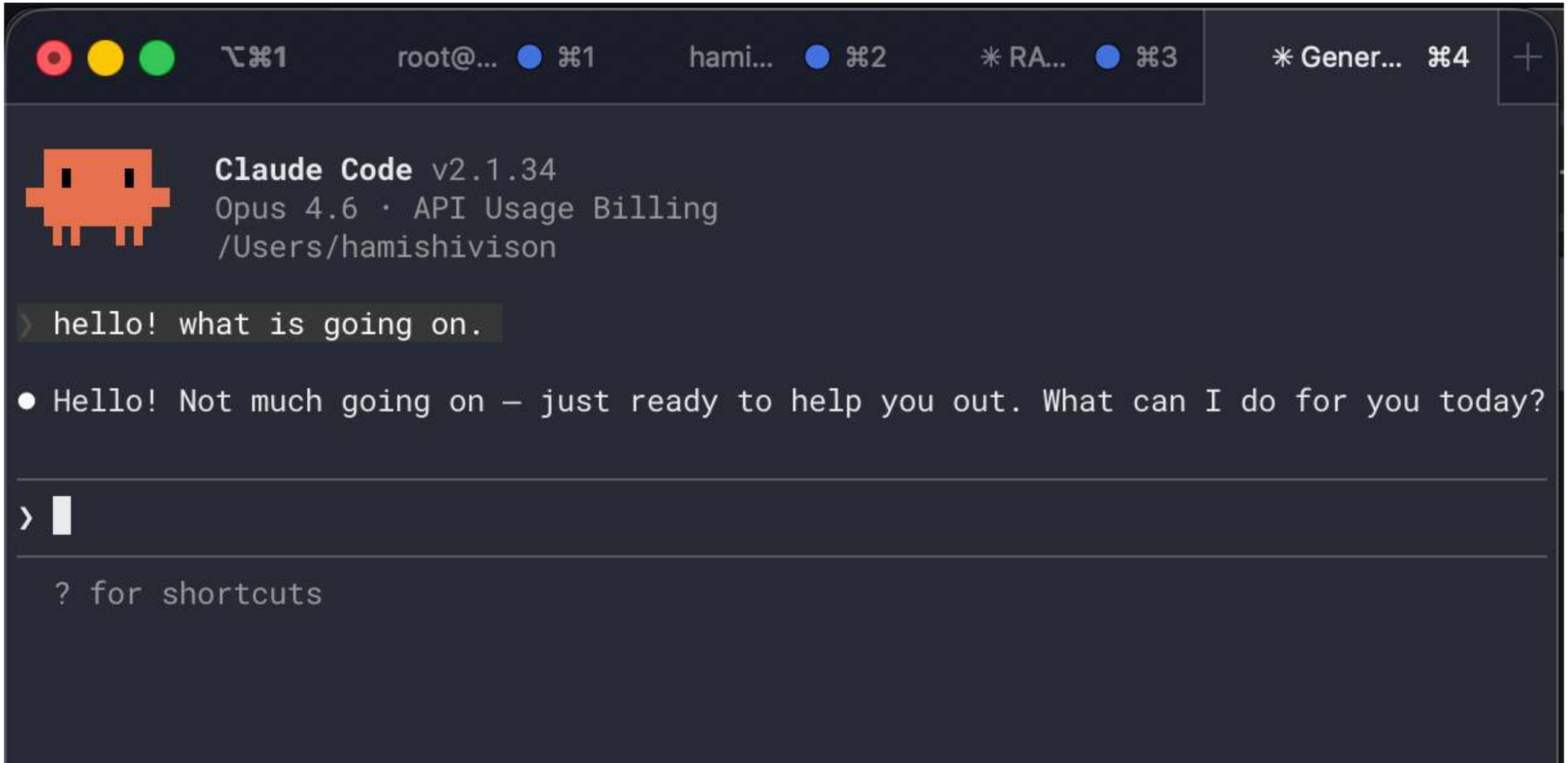


~1 month of training on 9 nodes

RL Training with tools and environments

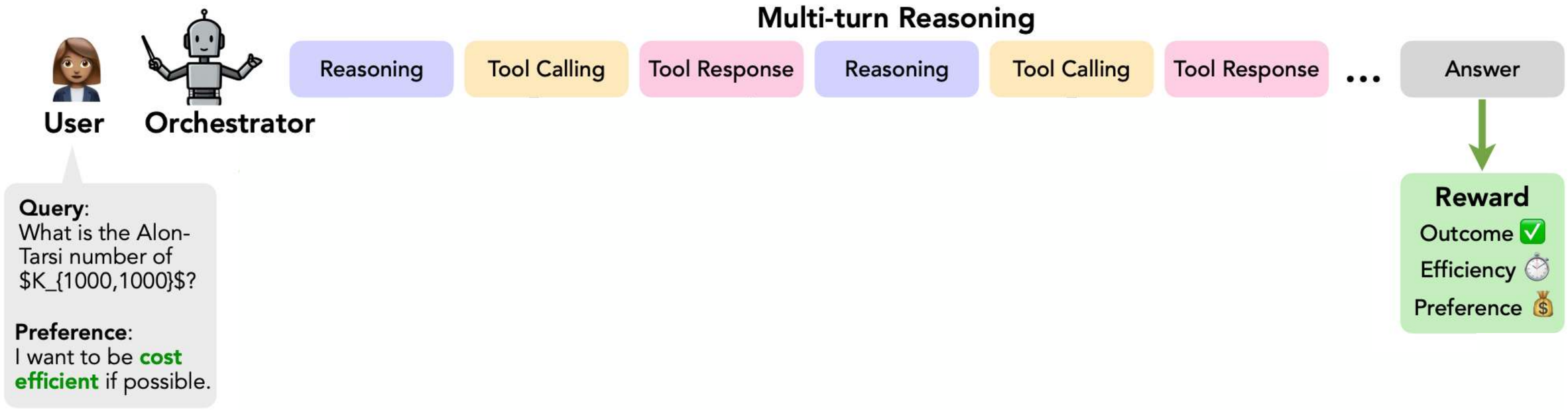
Tools & RL Environments

These days we expect use models as part of larger systems!



Tools & RL Environments

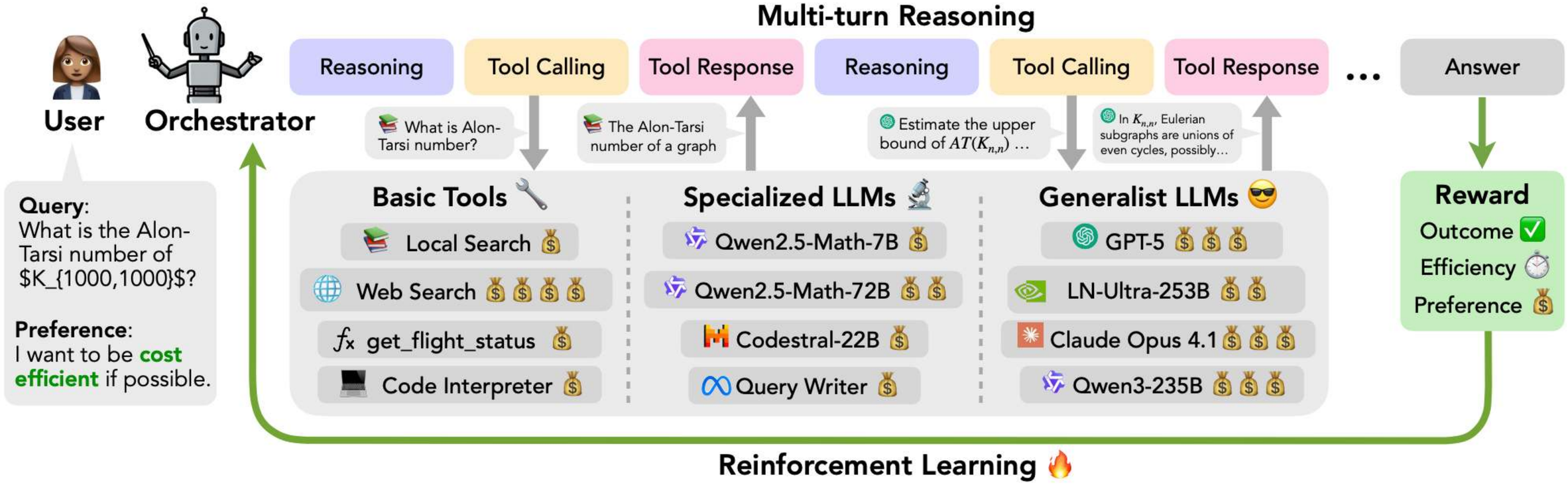
We can think of RL with tool calls as a *multi-turn environment*



Su et al., "*ToolOrchestra: Elevating Intelligence via Efficient Model and Tool Orchestration*," arXiv preprint arXiv:2511.21689, 2025.

Tools & RL Environments

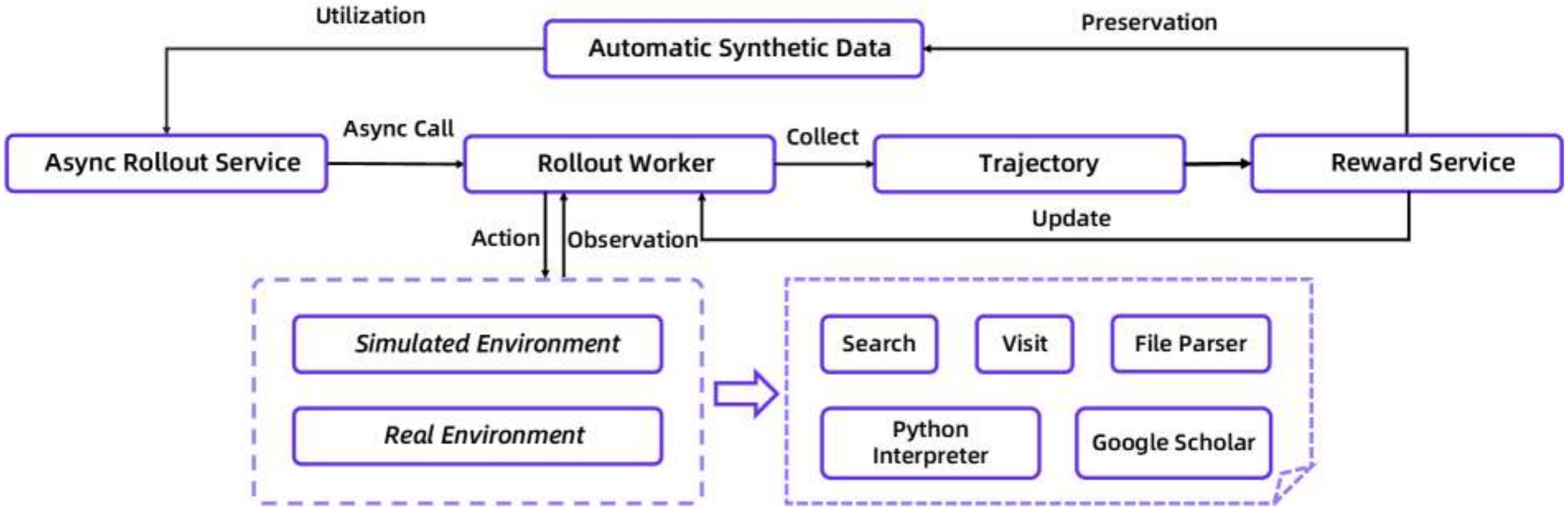
We can think of RL with tool calls as a *multi-turn environment*



Su et al., "*ToolOrchestra: Elevating Intelligence via Efficient Model and Tool Orchestration*," arXiv preprint arXiv:2511.21689, 2025.

Tools & RL Environments

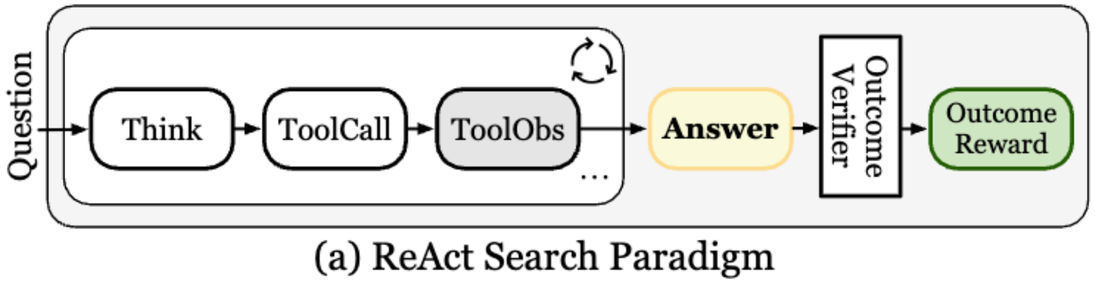
Far more infrastructure complexity!



Tongyi DeepResearch Team et al., "*Tongyi DeepResearch Technical Report,*" arXiv preprint arXiv:2510.24701, 2025.

Tools & RL Environments

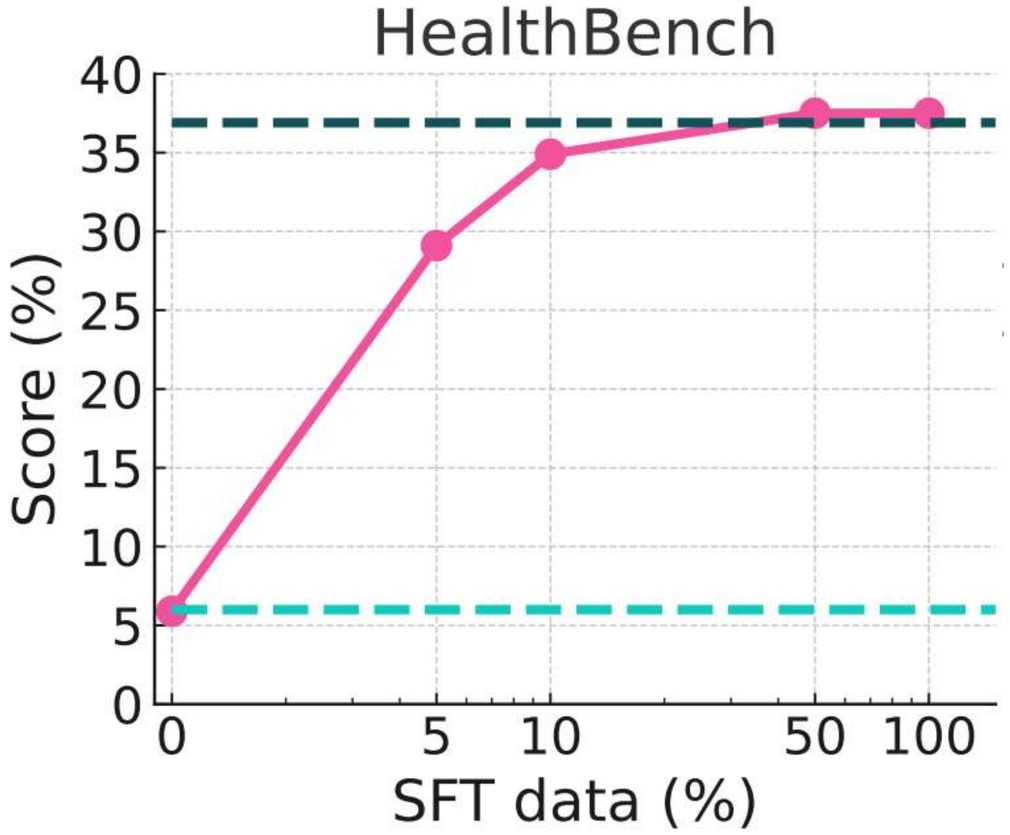
We can also often improve the algorithm in tool-use specific ways



(a) ReAct Search Paradigm

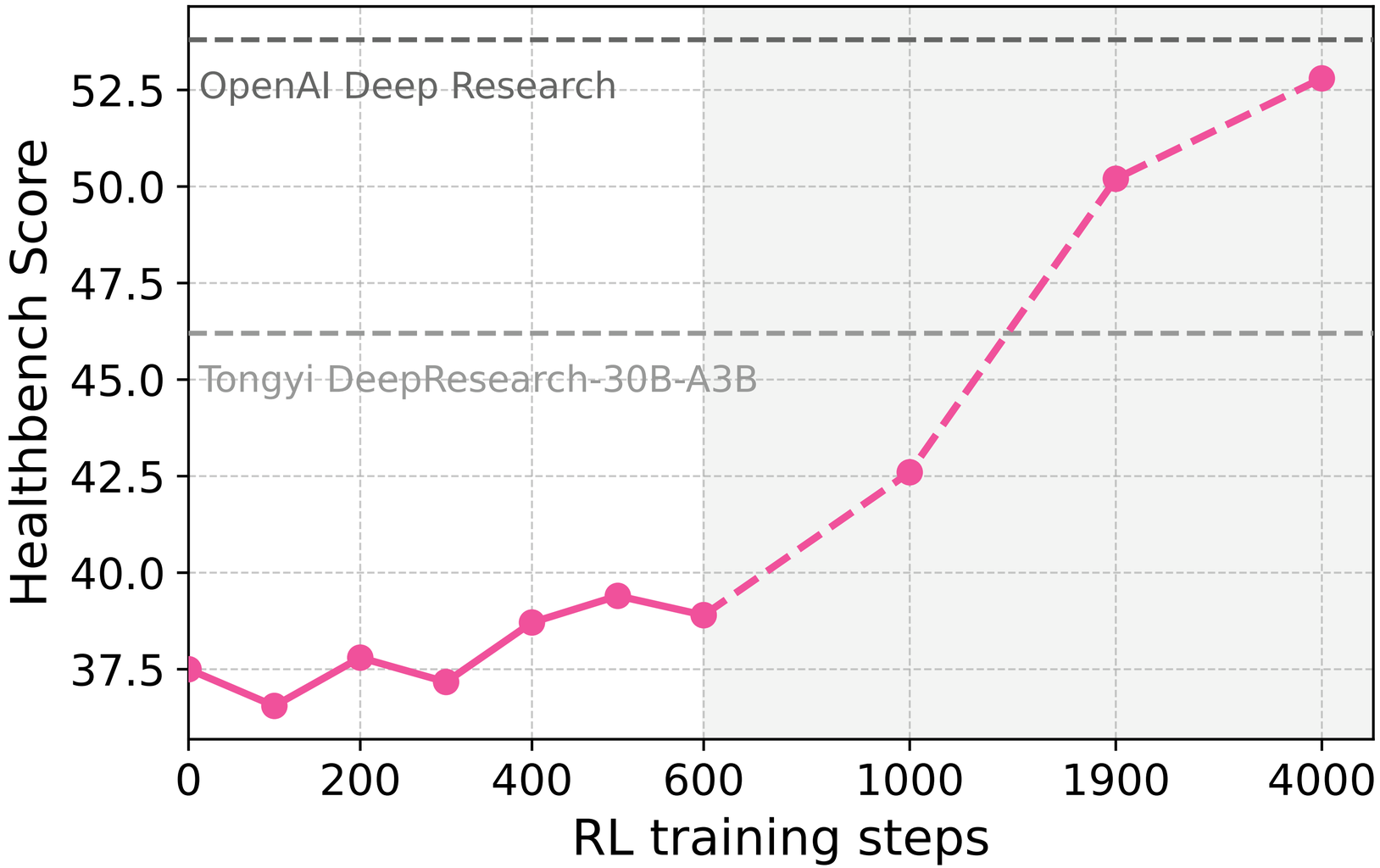
Tools & RL Environments

Using SFT alone can plateau…



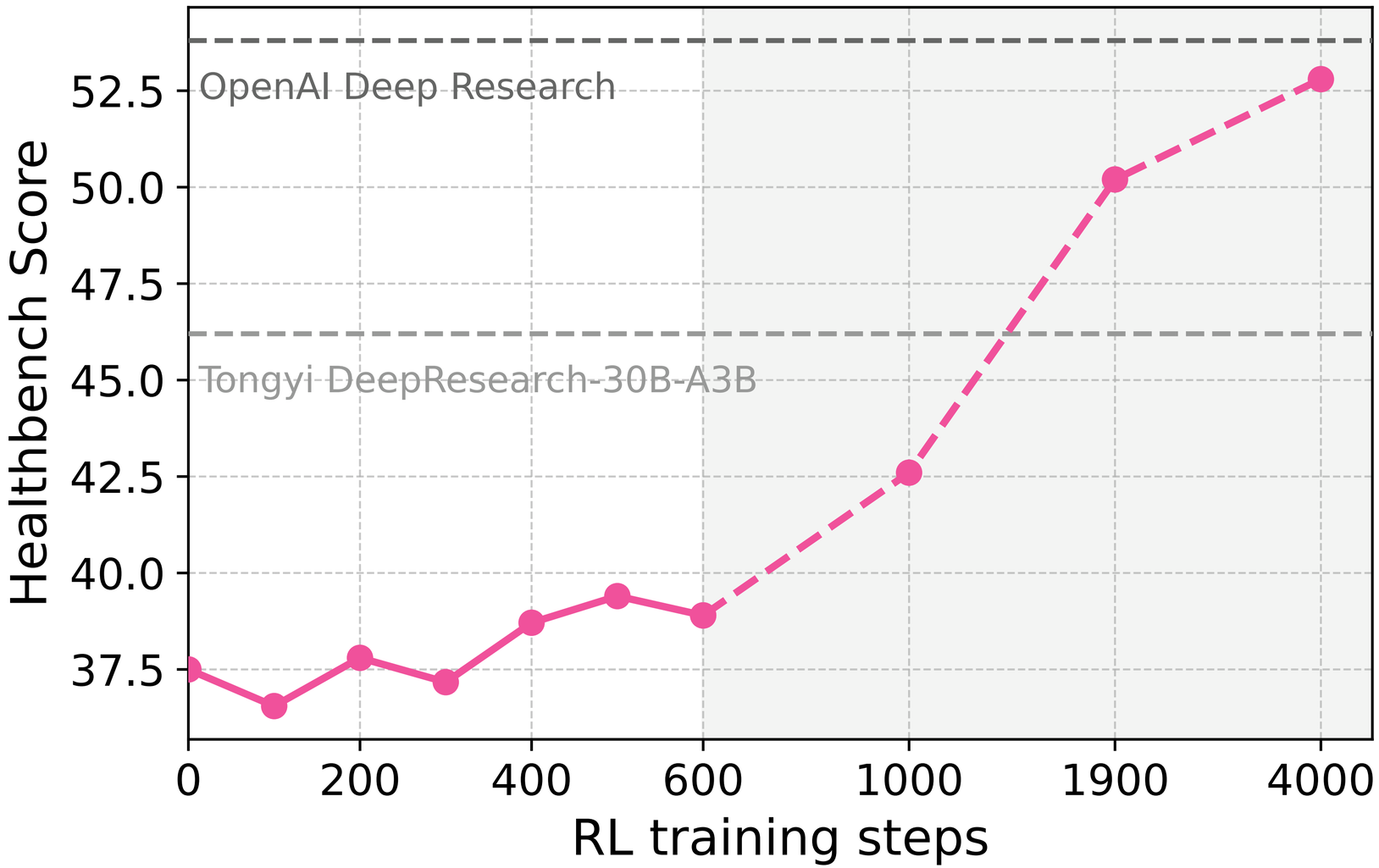
Tools & RL Environments

But then further RL can further improve.



Tools & RL Environments

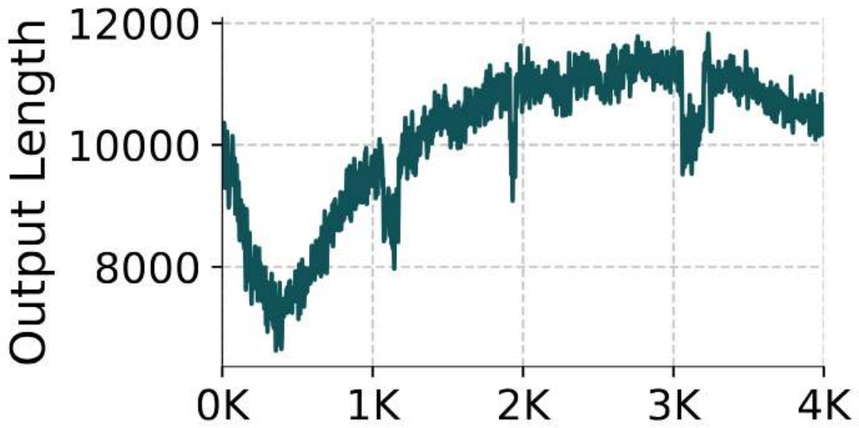
But then further RL can further improve.

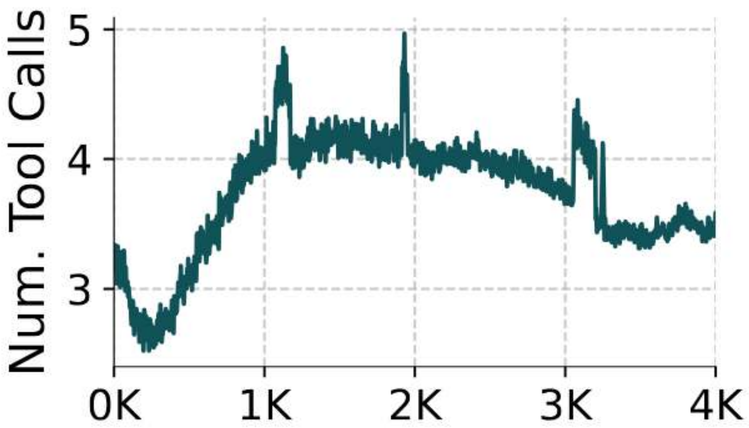


~2 months of training on 2 nodes

Tools & RL Environments

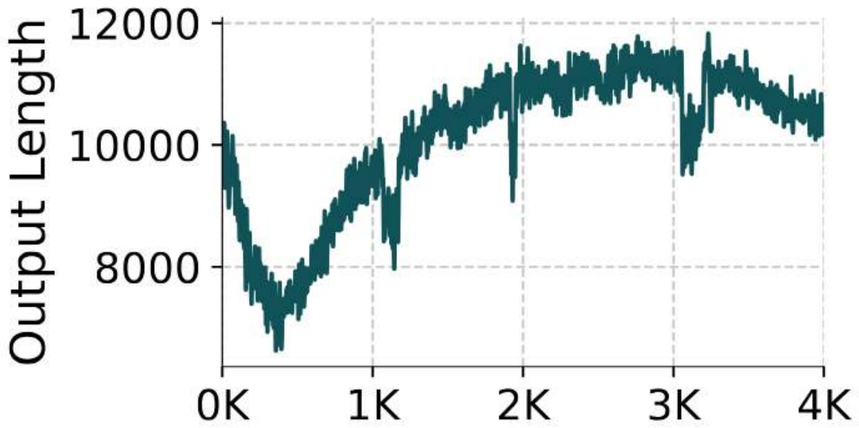
But some issues are recoverable!

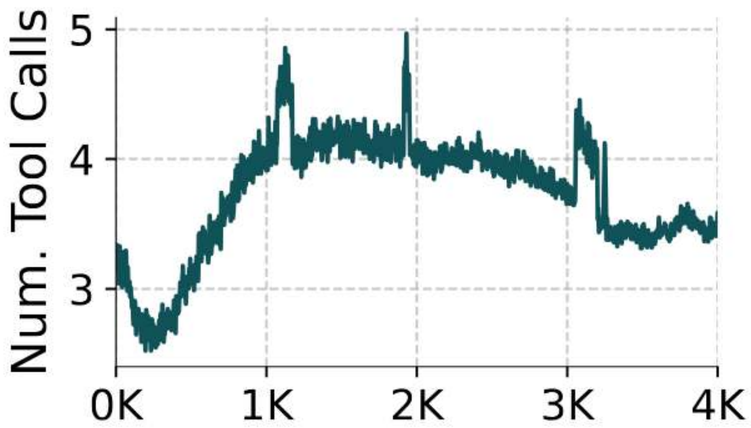


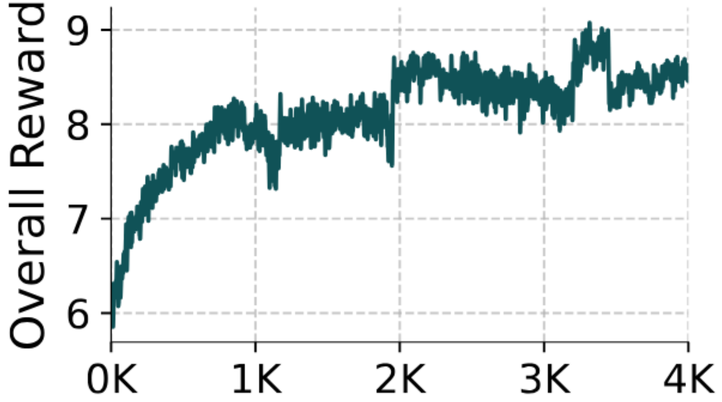


Tools & RL Environments

But some issues are recoverable!

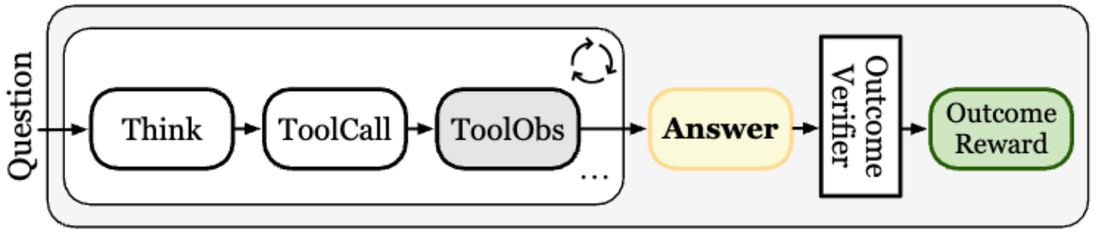




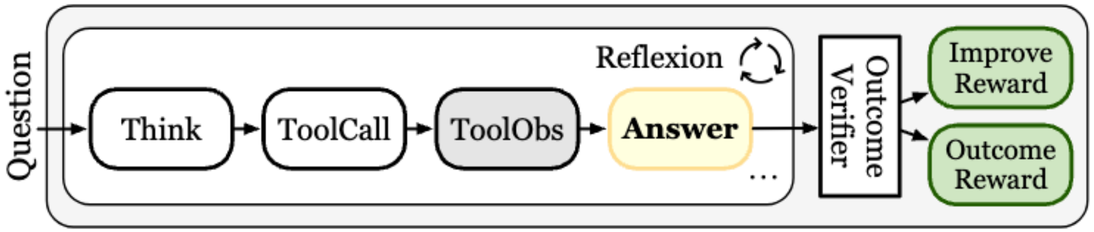


Tools & RL Environments

We can also often improve the algorithm in tool-use specific ways



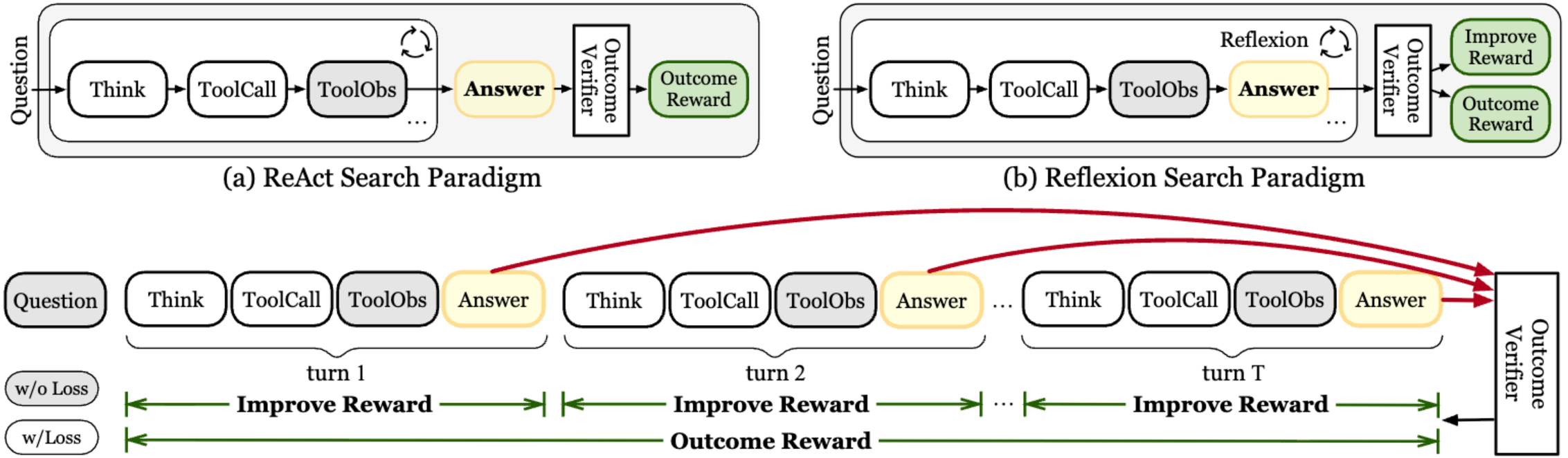
(a) ReAct Search Paradigm



(b) Reflexion Search Paradigm

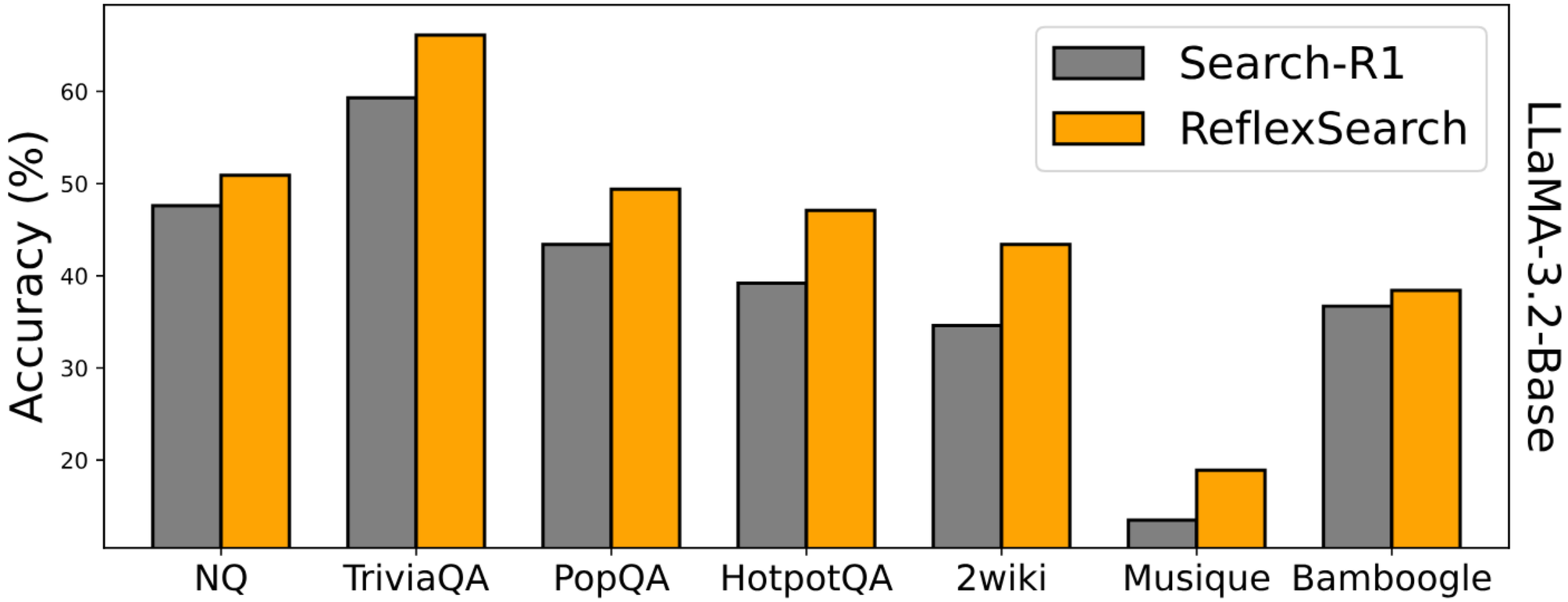
Tools & RL Environments

We can also often improve the algorithm in tool-use specific ways



Tools & RL Environments

We can also often improve the algorithm in tool-use specific ways



Tools & RL Environments

One recent example: DR Tulu

DR Tulu: Reinforcement Learning with Evolving Rubrics for Deep Research

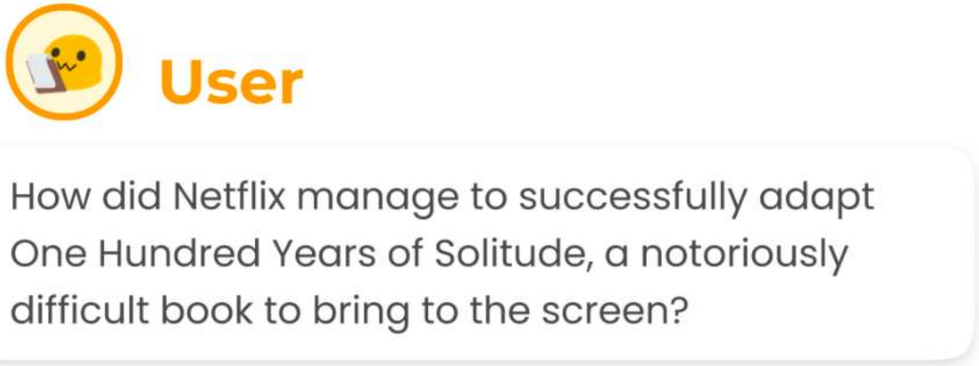
Rulin Shao[1♡†], **Akari Asai**[2,3♡†], **Shannon Zejiang Shen**[4♡†], **Hamish Ivison**[1,2♡†], **Varsha Kishore**[1,2†], **Jingming Zhuo**[1†], **Xinran Zhao**[3], **Molly Park**[1], **Samuel G. Finlayson**[1,5], **David Sontag**[4], **Tyler Murray**[2], **Sewon Min**[2,6], **Pradeep Dasigi**[2], **Luca Soldaini**[2], **Faeze Brahman**[2], **Wen-tau Yih**[1], **Tongshuang Wu**[3], **Luke Zettlemoyer**[1], **Yoon Kim**[4], **Hannaneh Hajishirzi**[1,2], **Pang Wei Koh**[1,2]

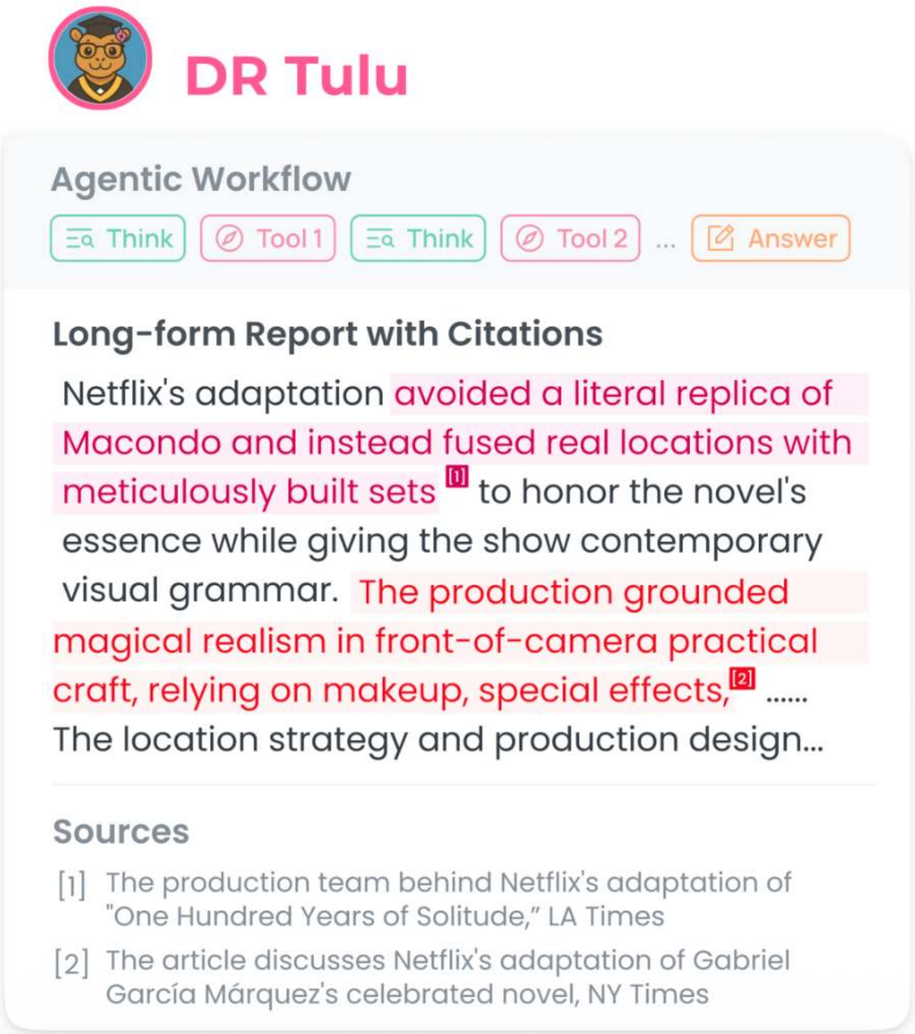
[1]University of Washington, [2]Allen Institute for AI, [3]Carnegie Mellon University
[4]Massachusetts Institute of Technology, [5]Seattle Children's Hospital, [6]University of California, Berkeley

[♡]Joint first authors, [†]Core contributors. See full author contributions here.

Tools & RL Environments

One recent example: DR Tulu





DR Tulu

“Deep Research”: Multi -paragraph answers that integrate external evidence andprovide fine-grained attributions to sources

Our goal: Unified model that iteratively plans, searches, integrates

`think` (`<think></think>`) uses the LM itself to plan next steps given the current state and information.

`tool` (`<call_tool></call_tool>`) invokes one of multiple search-related tools. The specific tool is chosen by setting the `name` attribute and tool-specific arguments. *Example:* `<call_tool name="google_-search" k="10" lang="en">query</call_tool>`. We append the tool's output, in plain text, to the context for subsequent steps.

`answer` (`<answer></answer>`) produces the final response and stops. Inside the answer, the LM may enclose claims in `<cite id="SOURCE_ID"></cite>` tags to provide references to supporting source identifiers.

DR Tulu

Prior work mainly focussed on:

1. Developing more complex scaffolds *without model training*
2. Training on extremely hard short-form QA problems requiring many (>10) searches

`think` (`<think></think>`) uses the LM itself to plan next steps given the current state and information.

`tool` (`<call_tool></call_tool>`) invokes one of multiple search-related tools. The specific tool is chosen by setting the `name` attribute and tool-specific arguments. *Example:* `<call_tool name="google_-search" k="10" lang="en">query</call_tool>`. We append the tool's output, in plain text, to the context for subsequent steps.

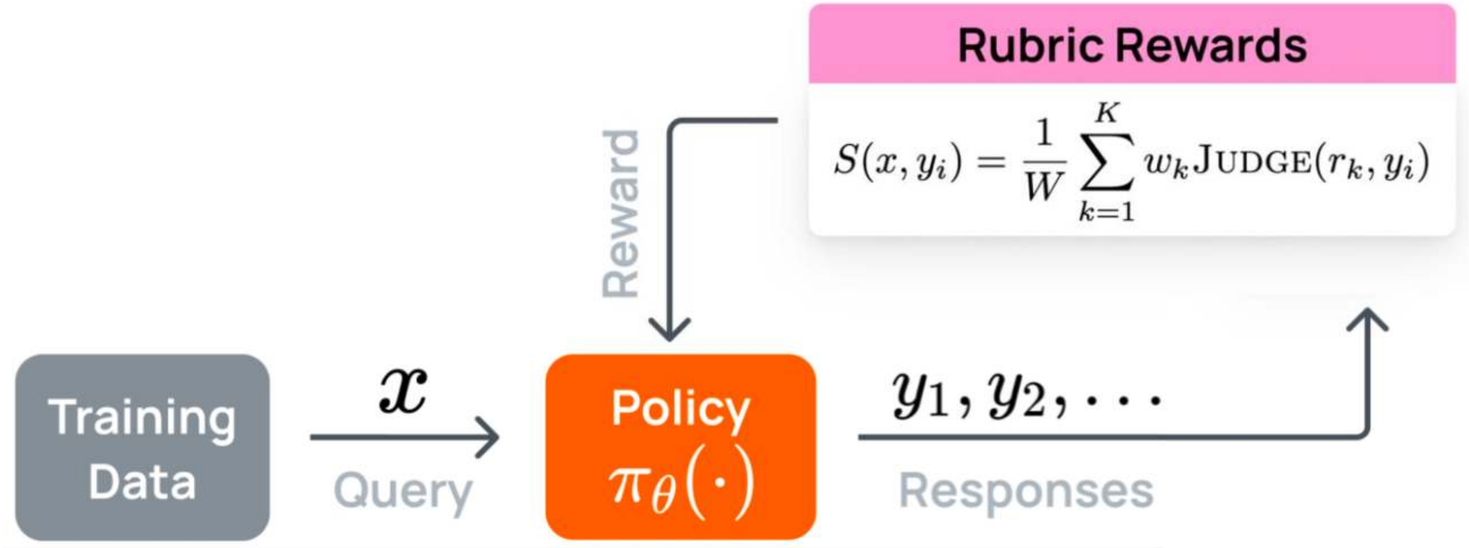
`answer` (`<answer></answer>`) produces the final response and stops. Inside the answer, the LM may enclose claims in `<cite id="SOURCE_ID"></cite>` tags to provide references to supporting source identifiers.

Why is DR hard to train?

1. **High-quality** answers are hard to come by!
2. **Responses are not verifiable** (long-form text).

Why is DR hard to train?

1. **High-quality** answers are hard to come by!
2. **Responses are not verifiable** (long-form text).



Is rubrics enough for DR?

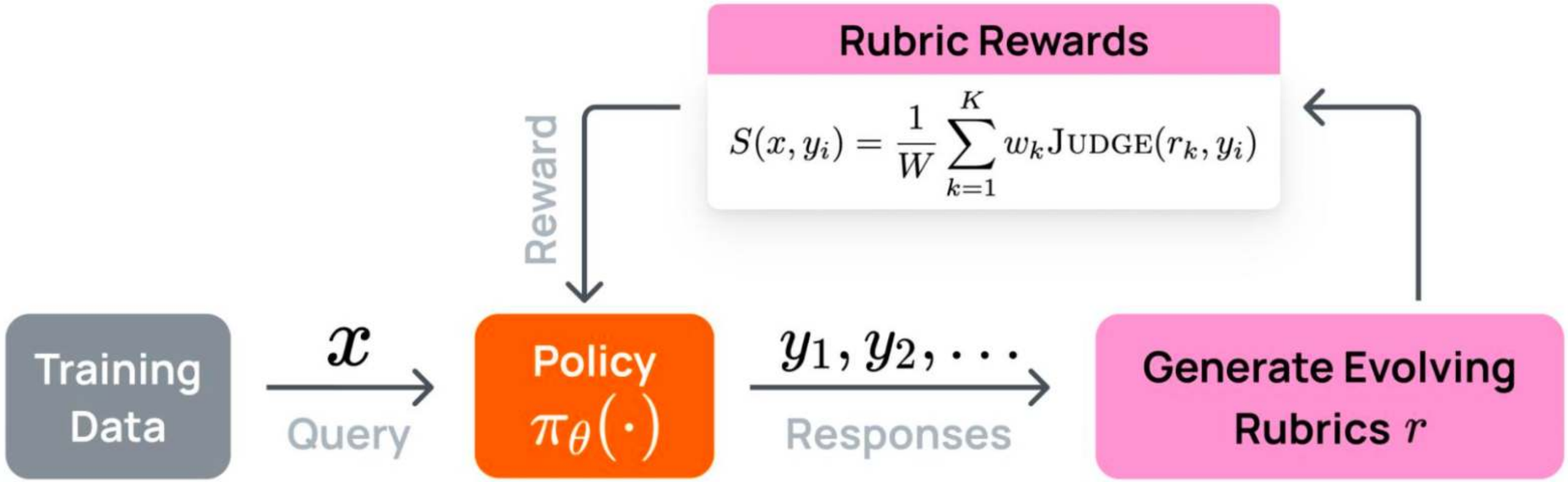
1. **Rubrics need to be specific!**
    - Need to reference up-to-date information
    - e.g. “The response should be factually correct” - good rubric for a human judge, bad for a LM!
2. **May not be discriminative for current model**!

Is rubrics enough for DR?

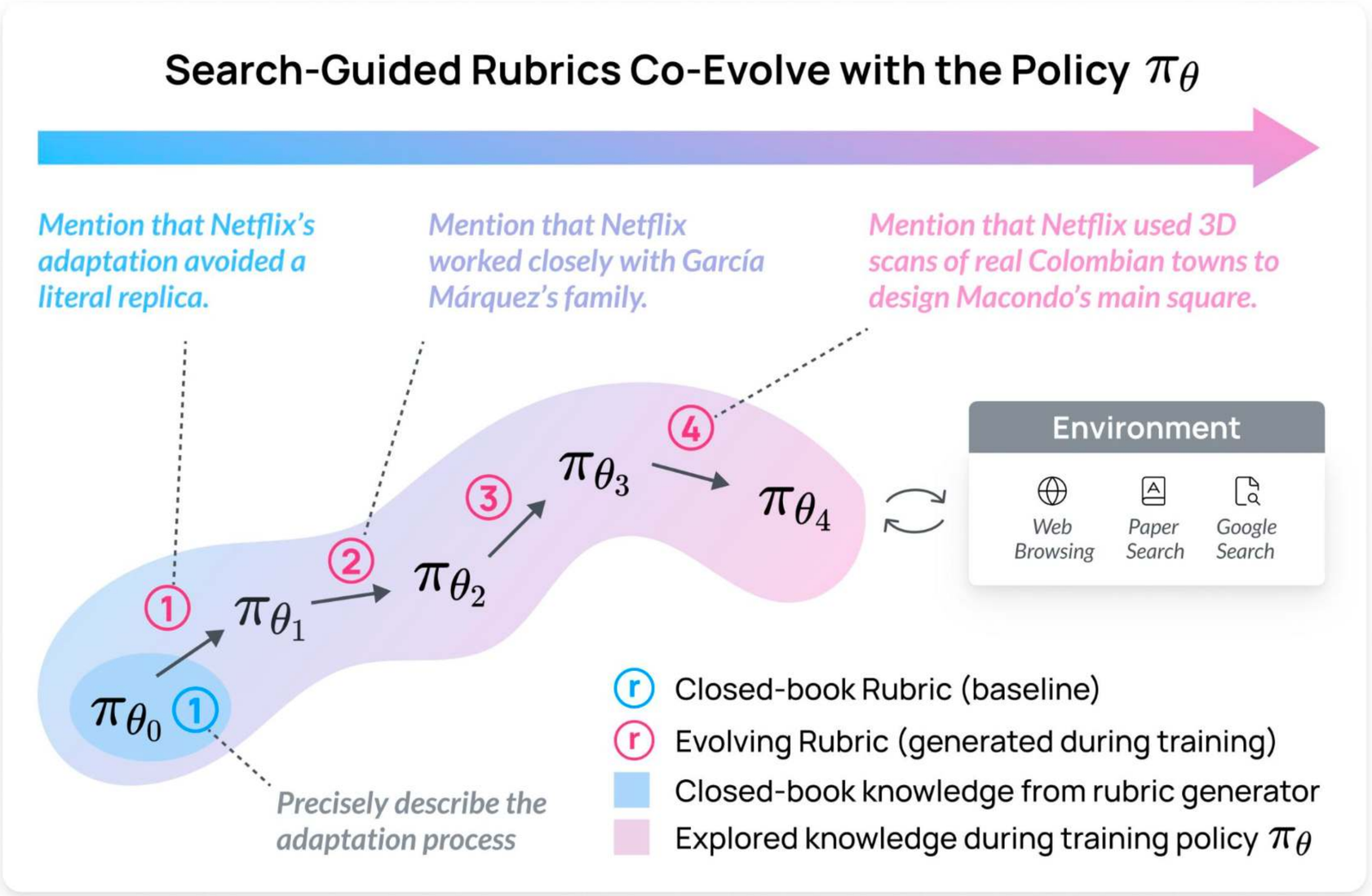
1. **Rubrics need to be specific!**
    - Need to reference up-to-date information
    - e.g. “The response should be factually correct” - good rubric for a human judge, bad for a LM!
2. **May not be discriminative for current model**!

Our solution: *Evolving Rubrics*

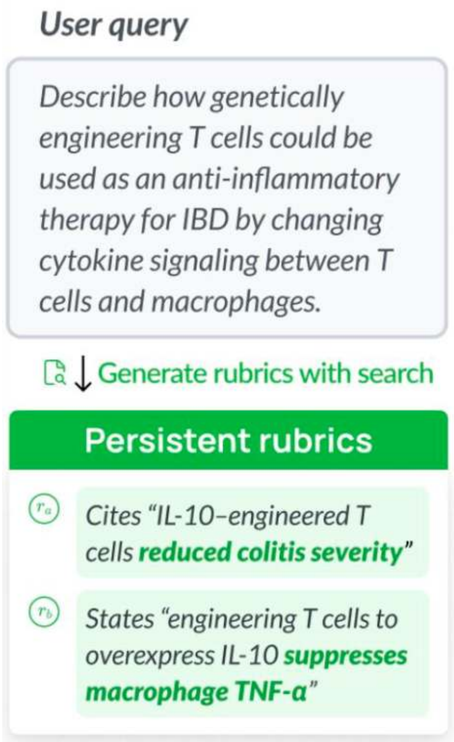
Iteratively update rubrics **based on current model generations**



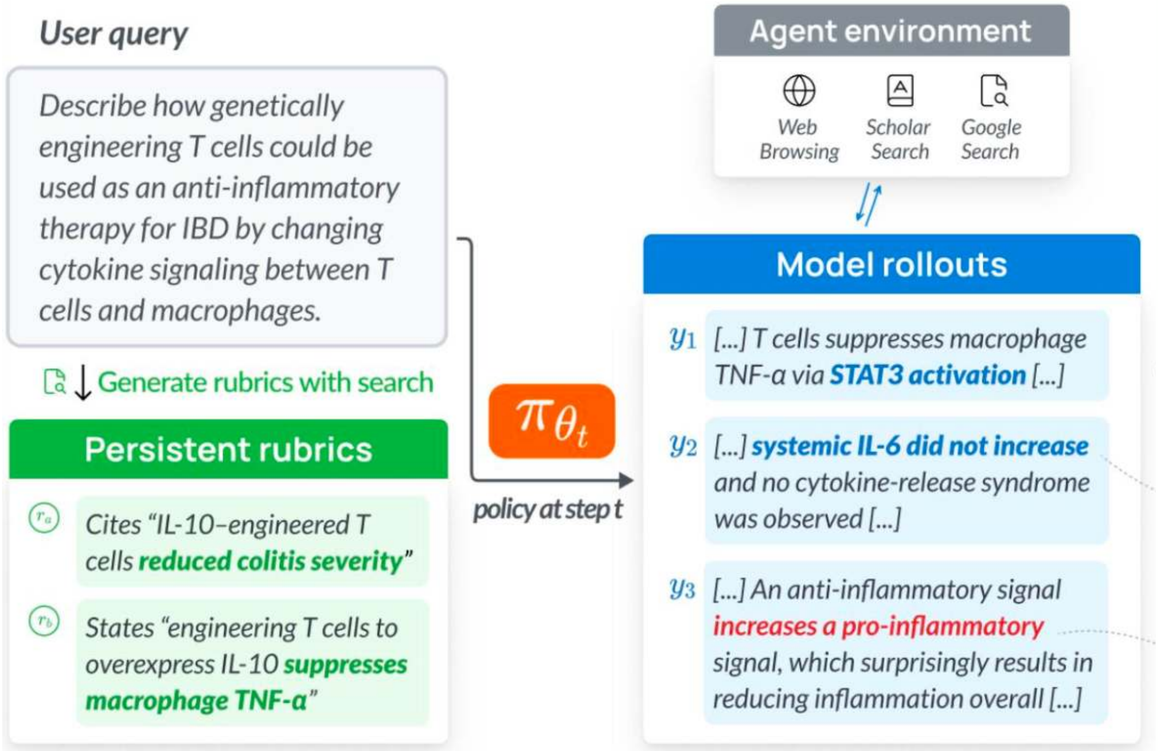
Our solution: *Evolving Rubrics*



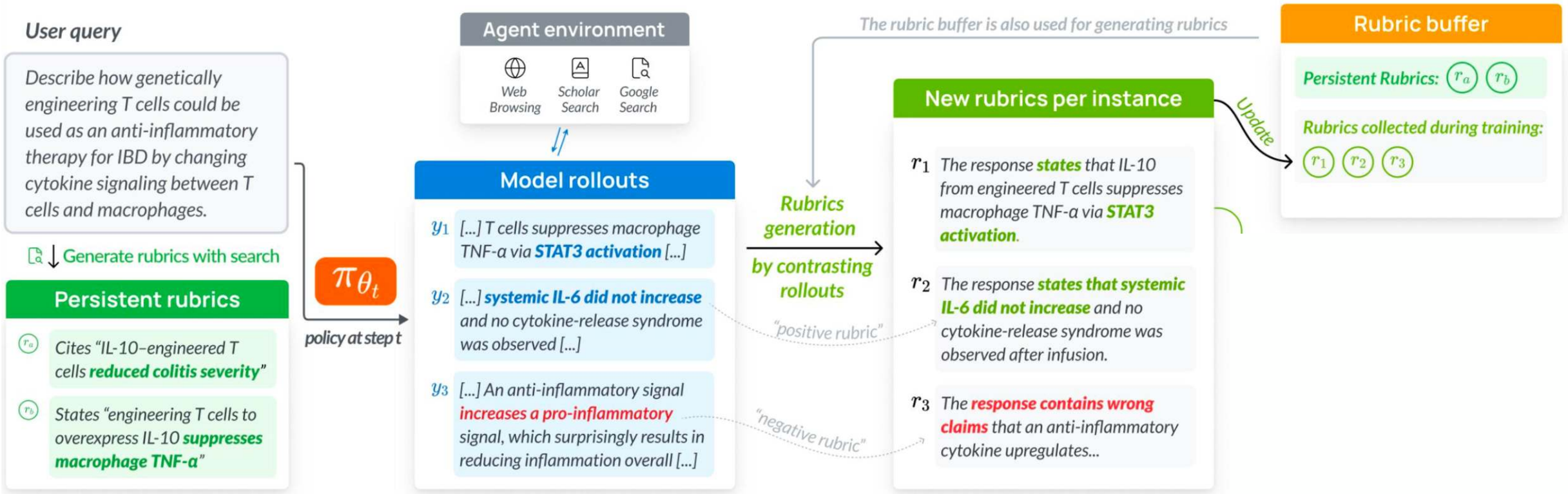
DR Tulu Training



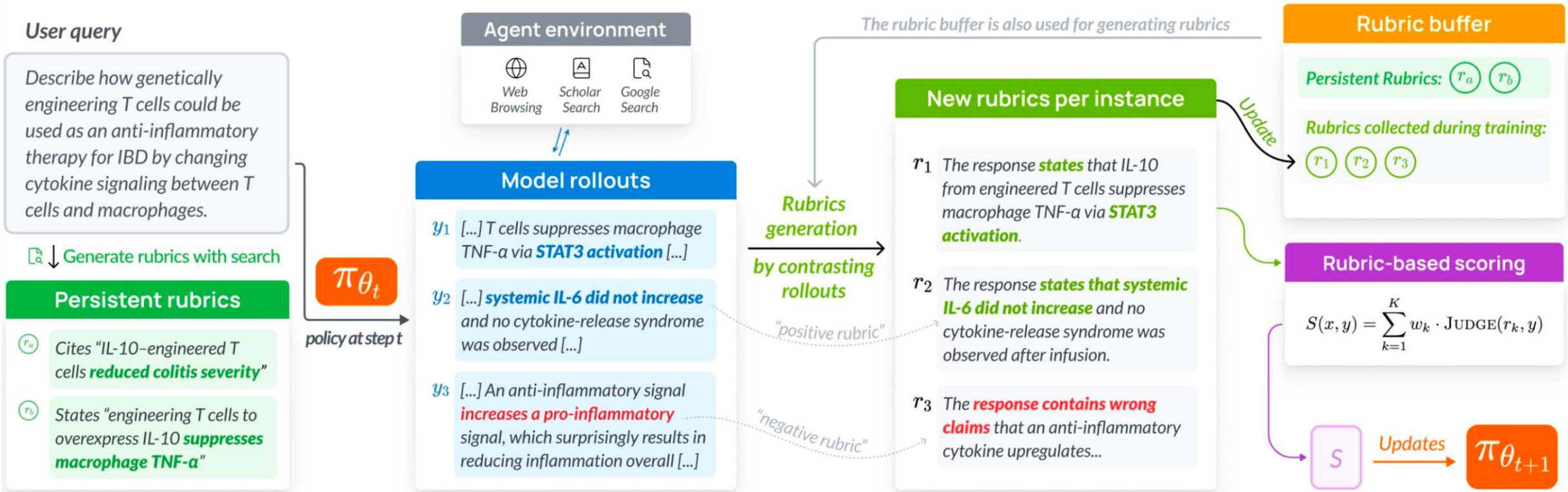
DR Tulu Training



DR Tulu Training



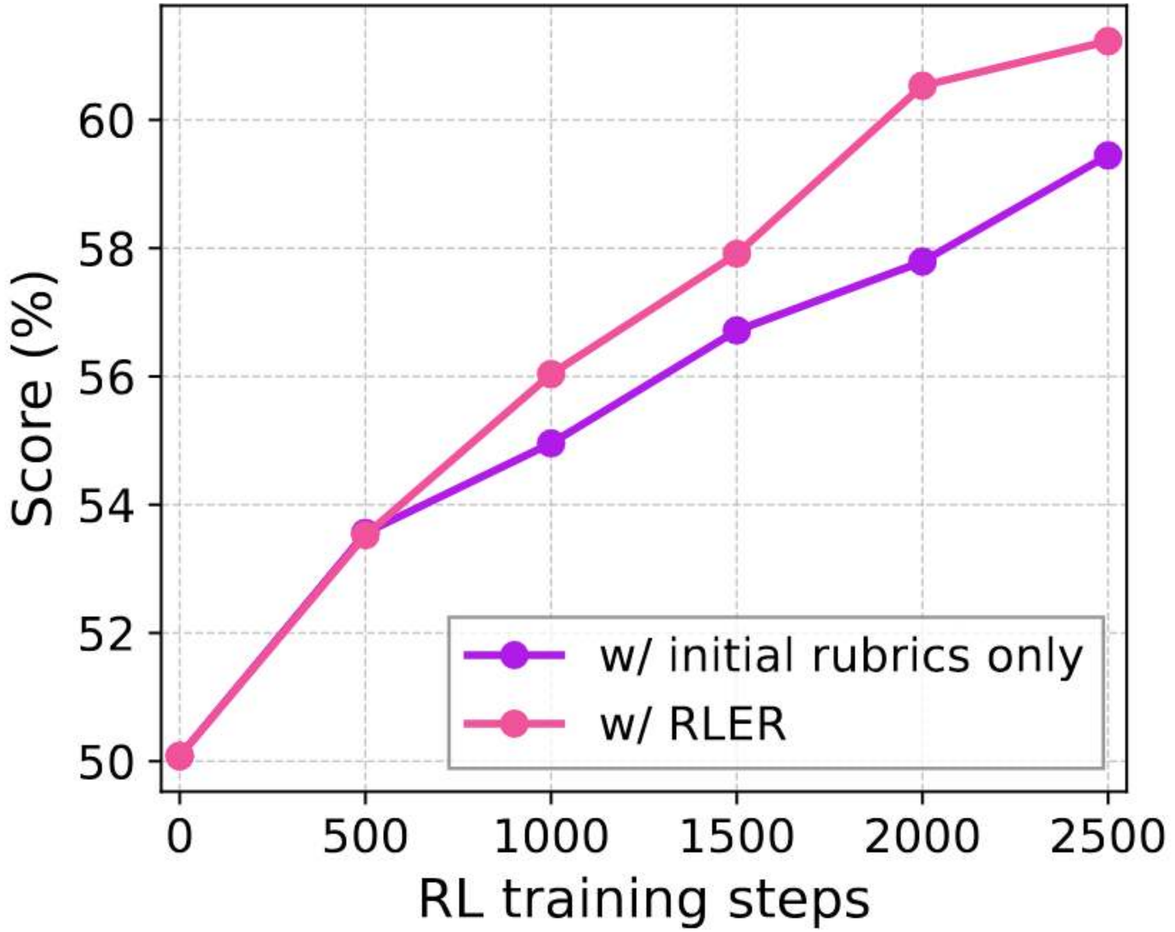
DR Tulu Training



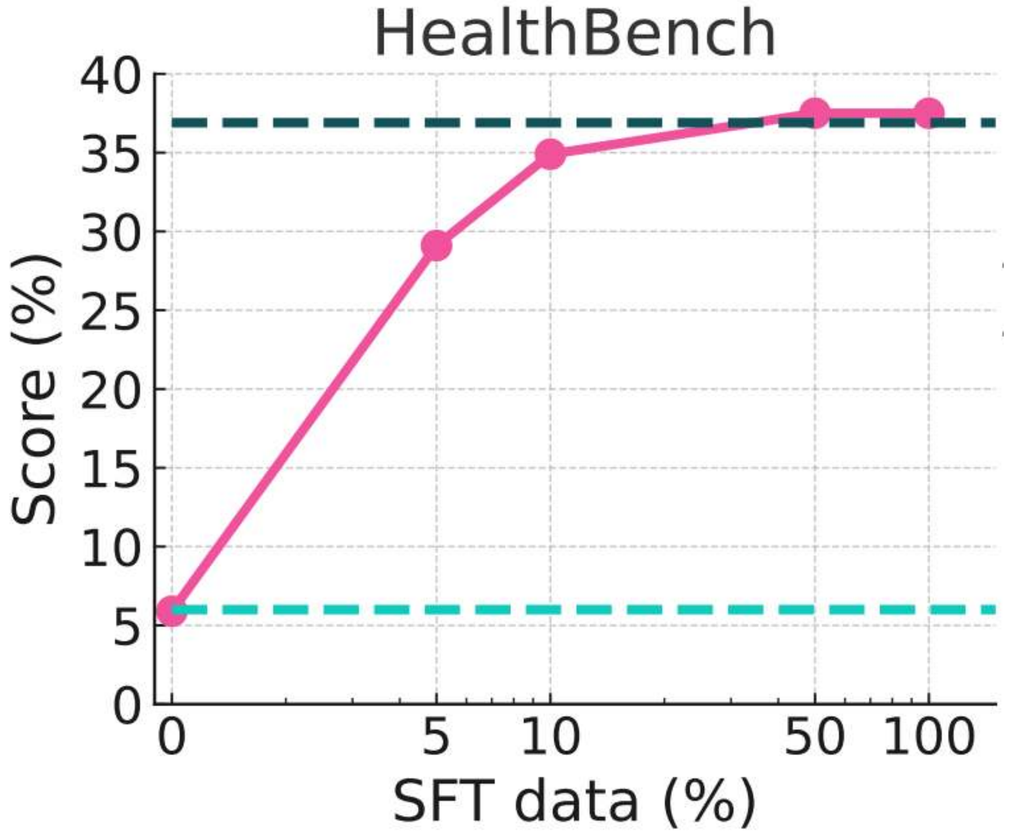
Using more specific rubrics helps!

	SQAv2	Health	Research	DRB	Avg.
DR Tulu (SFTv0.1)	73.4	37.5	68.6	39.4	54.7
General rubrics	80.6	36.0	65.0	37.5	54.8
Closed-book rubrics	83.2	34.8	66.6	37.6	55.6
Initial search-based rubrics	82.8	37.9	66.9	39.3	56.7

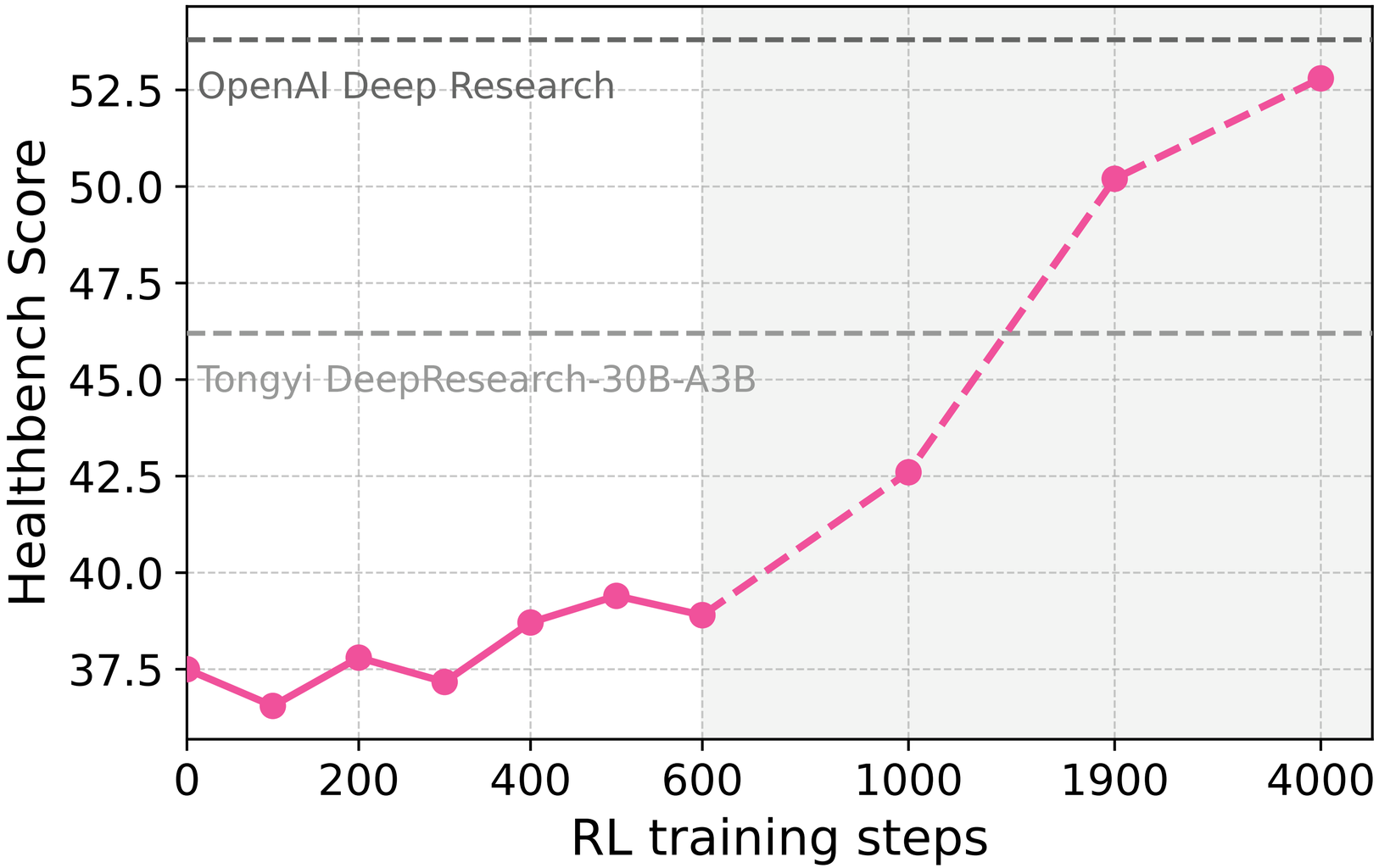
Using evolving rubrics helps even more!



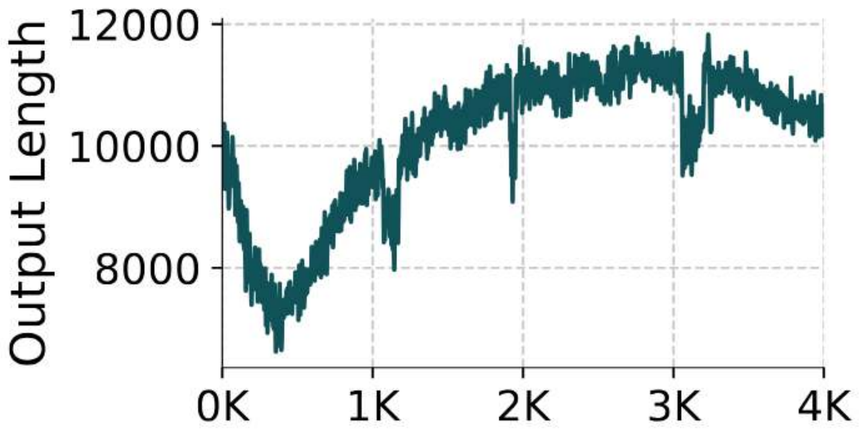
Using SFT alone is not enough

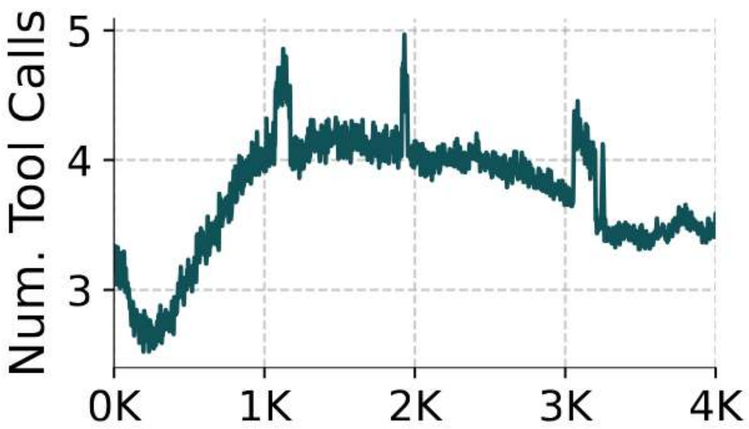


RL takes a while to show clear improvements

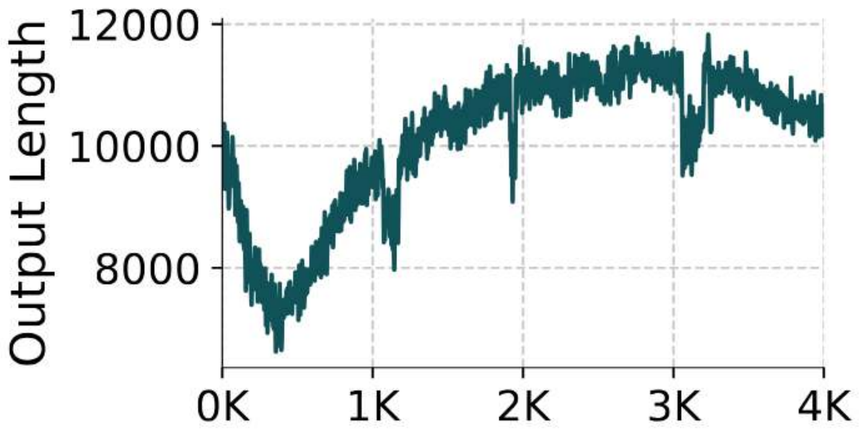


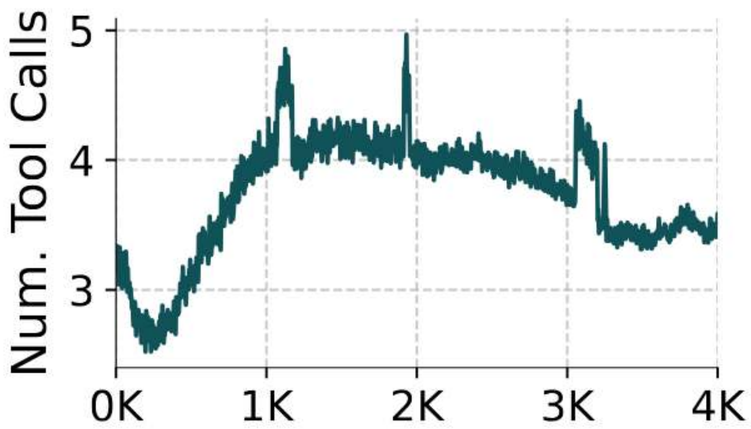
RL is (somewhat) robust to infra problems

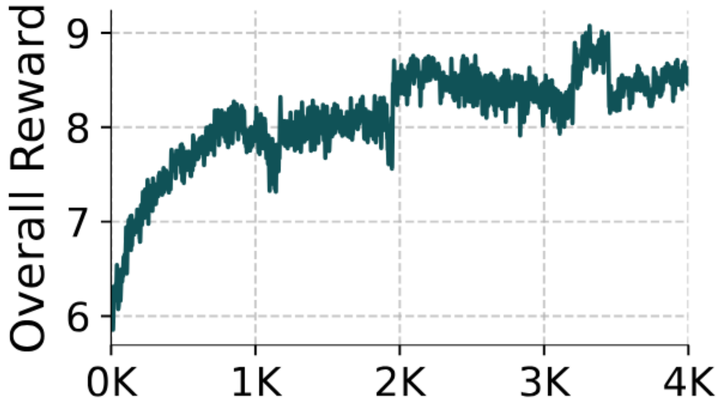




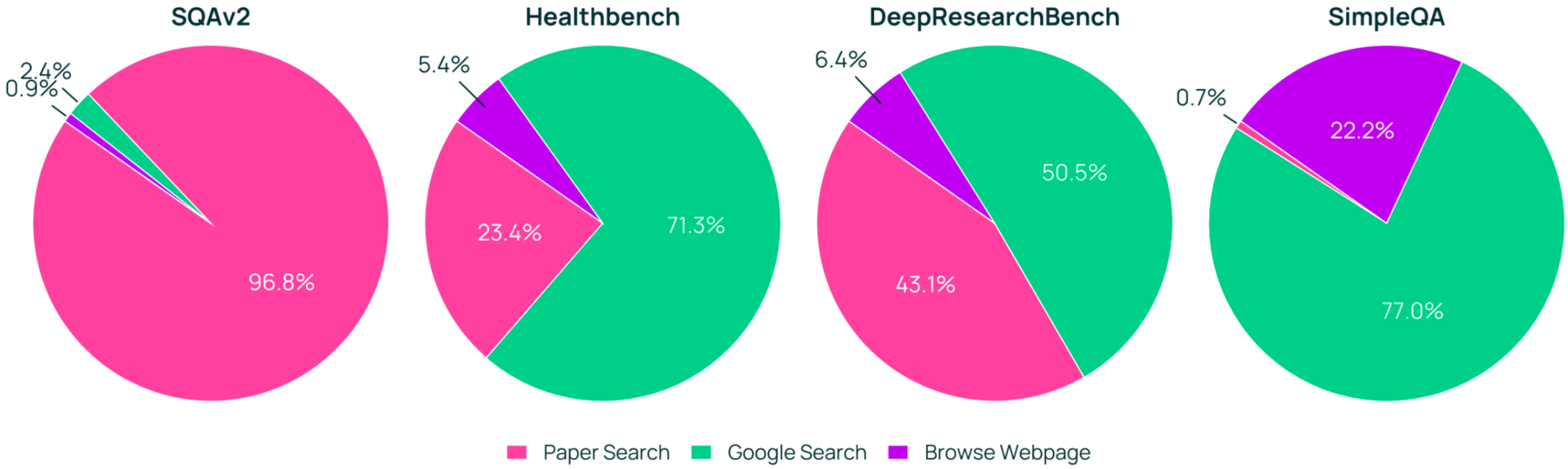
RL is (somewhat) robust to infra problems







RL learns the most appropriate tool to use

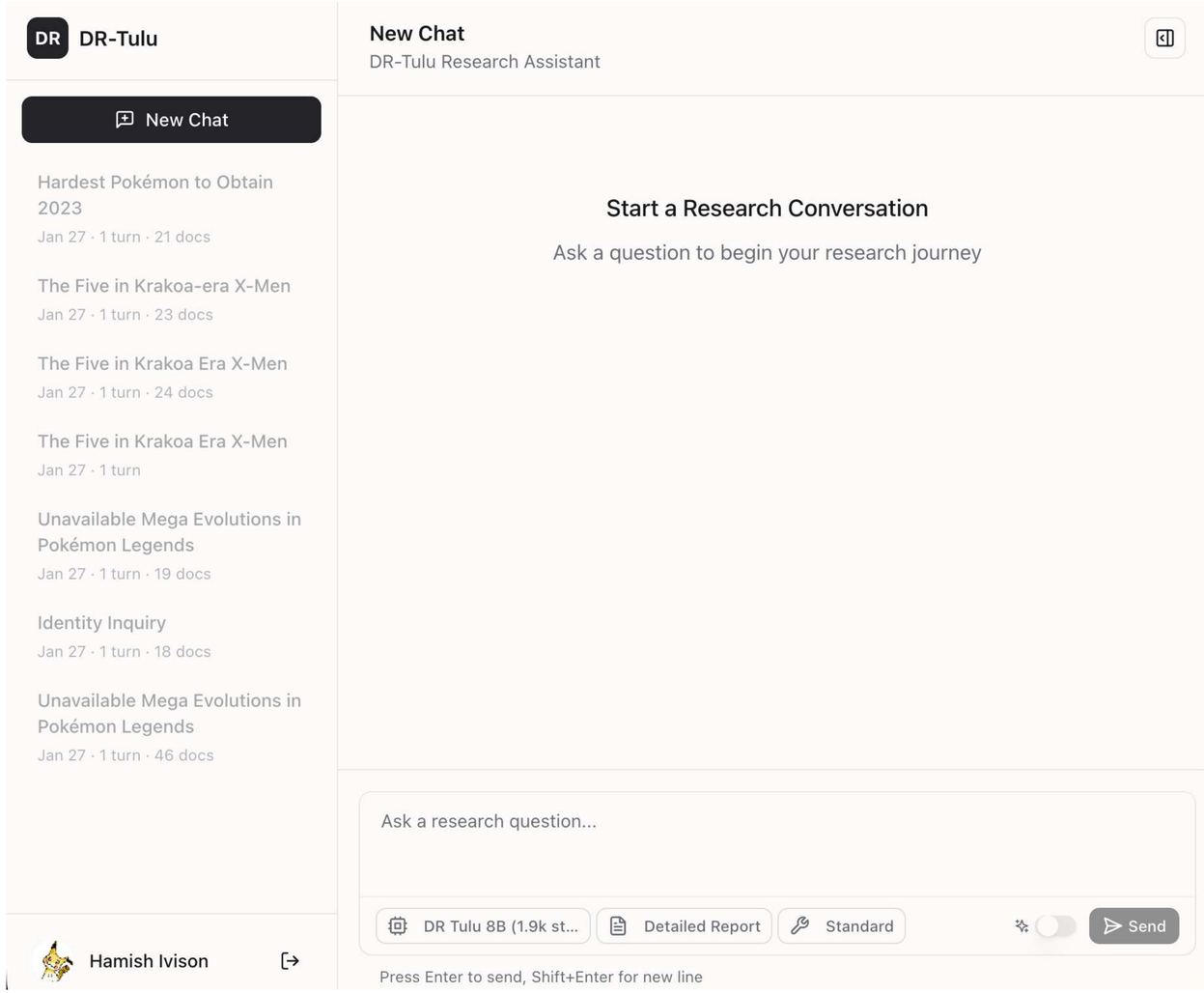


See the paper for more!

Code and data all available!

Includes:

- Custom agent infrastructure library (dr-agent-lib)
- one-turn async tool calling GRPO implementation (open-instruct)



dr-tulu.org

Training a ‘reasoner’ LM with RL

Training a ‘reasoner’ LM with RL

What is Olmo?

- Fully open (data, code, weights) models -> Useful for researchers and AI community to inspect, reproduce, innovate on.
- Demonstrates that properly developed open models can match or exceed proprietary alternatives (no secret sauce)

Pre-training
Open data,
New data paradigm

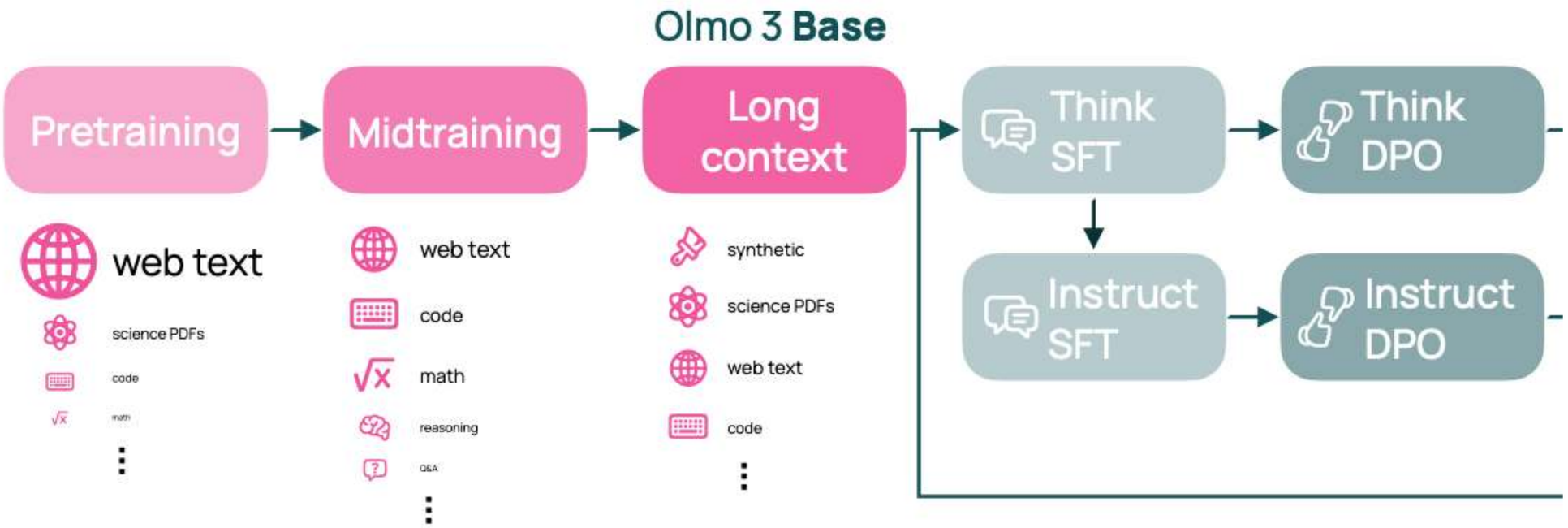
Post-training
RLVR
Thinking, Tool Use

Multimodal

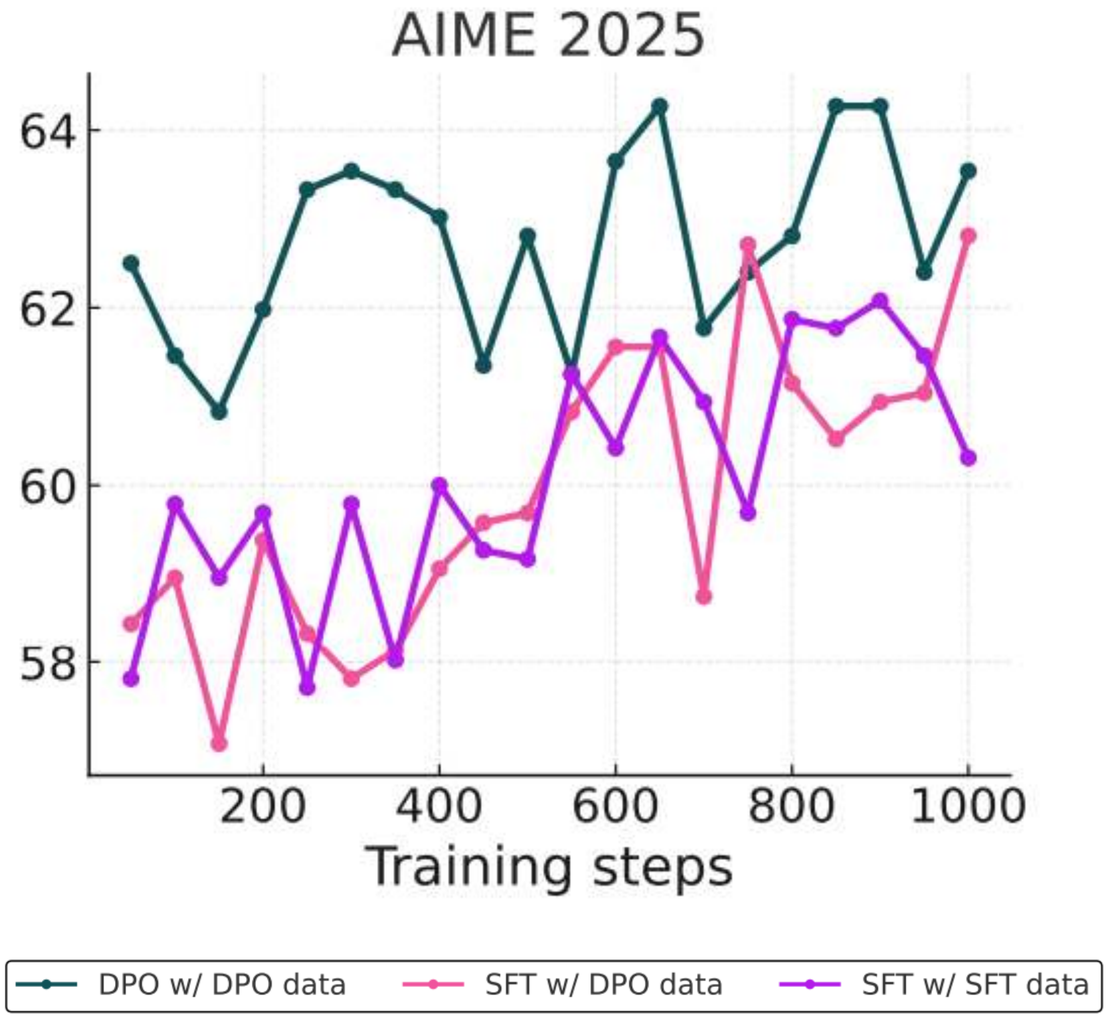
Olmo FlexOlmo OlmoThink Tülu

Dolma Molmo DR Tulu

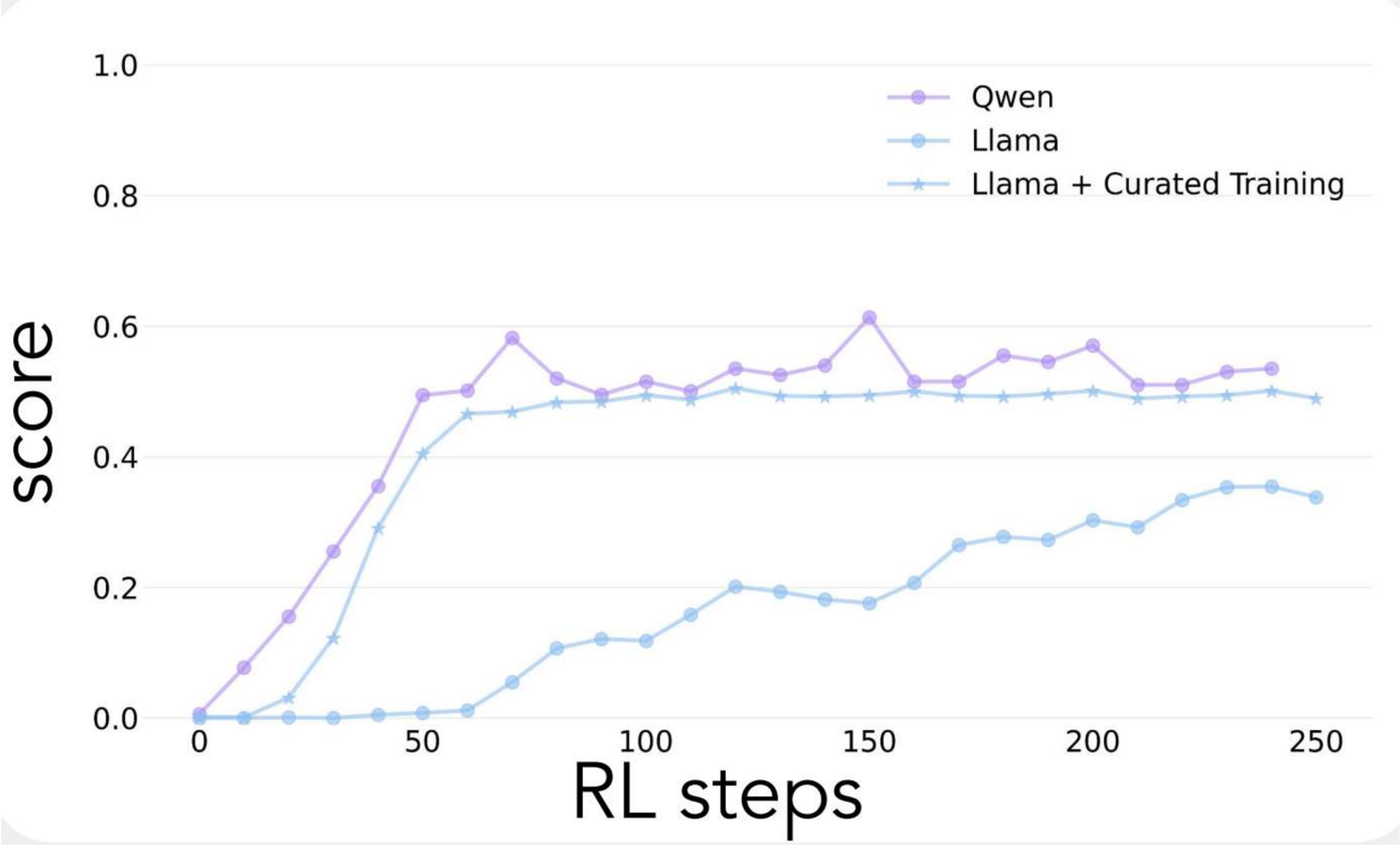
Start with a strong base



Start with a strong base

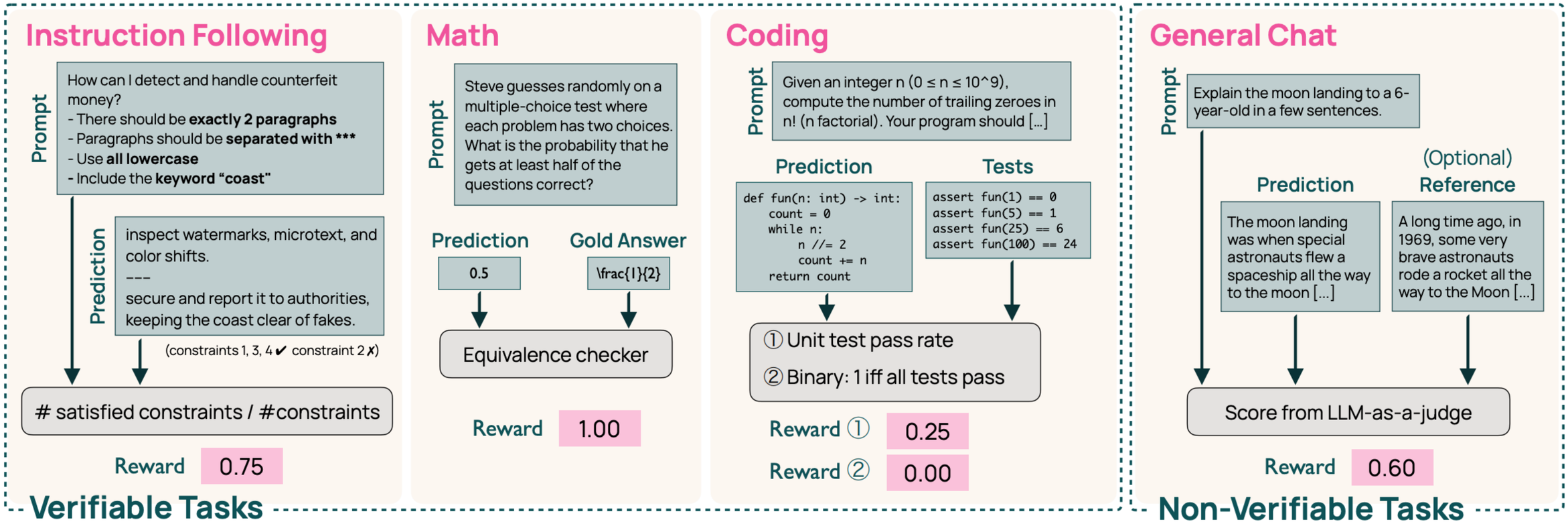


Start with a strong base *midtrained on reasoning data*

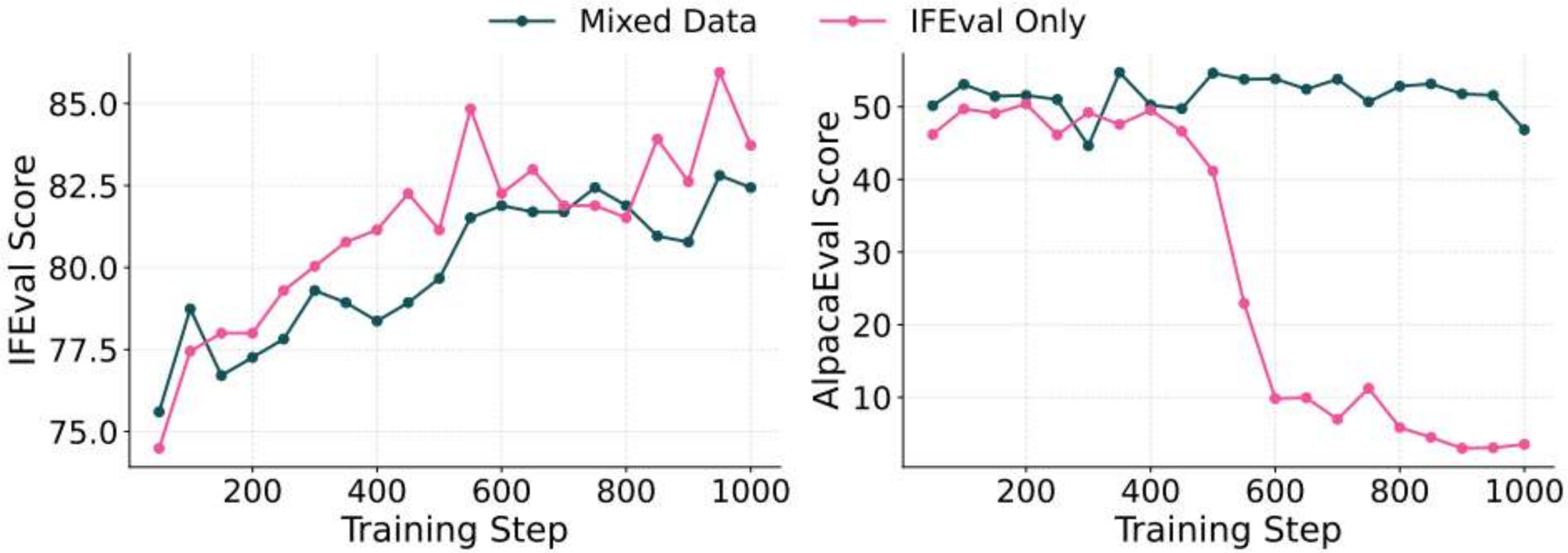


Gandhi et al., "*Cognitive Behaviors that Enable Self-Improving Reasoners, or, Four Habits of Highly Effective STaRs*," COLM 2025, arXiv:2503.01307

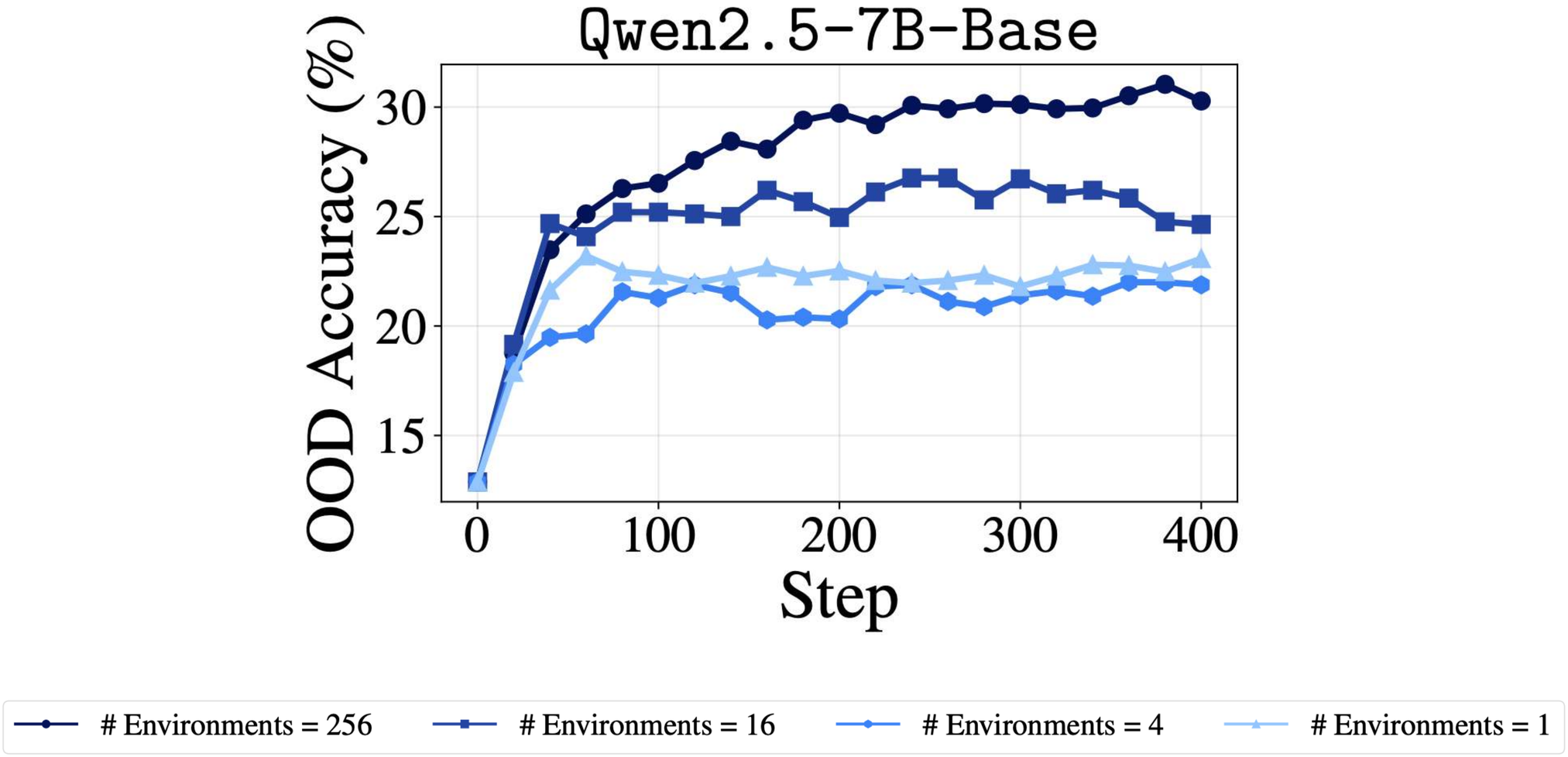
Prepare a diverse mix of RL data



Prepare a diverse mix of RL data



More RL diversity -> more generalisation!



Filter your data aggressively

Step 2: offline difficulty filtering As stated previously, to improve the sample efficiency of RL for our reasoner model, we generate eight rollouts for each prompt from the initial checkpoint of the model we train (e.g., if starting from the DPO-trained model, we generate from the DPO checkpoint). We then remove all samples that the model easily solves (that is, those with a pass rate greater than 62.5%). We sample with a temperature of 1.0 and top-p of 1.0, matching how we sample during RL training. We used offline filtering for the 7B OLMO 3 THINK to filter out RL problems that are too easy for our models' training. For the 32B, we rely on active sampling, which fills RL batches only on samples with a non-zero GRPO group gradient, and re-using the 7B DPO-filtered data as the starting point for the model due to compute and time constraints.

Adjust your RL algorithm!

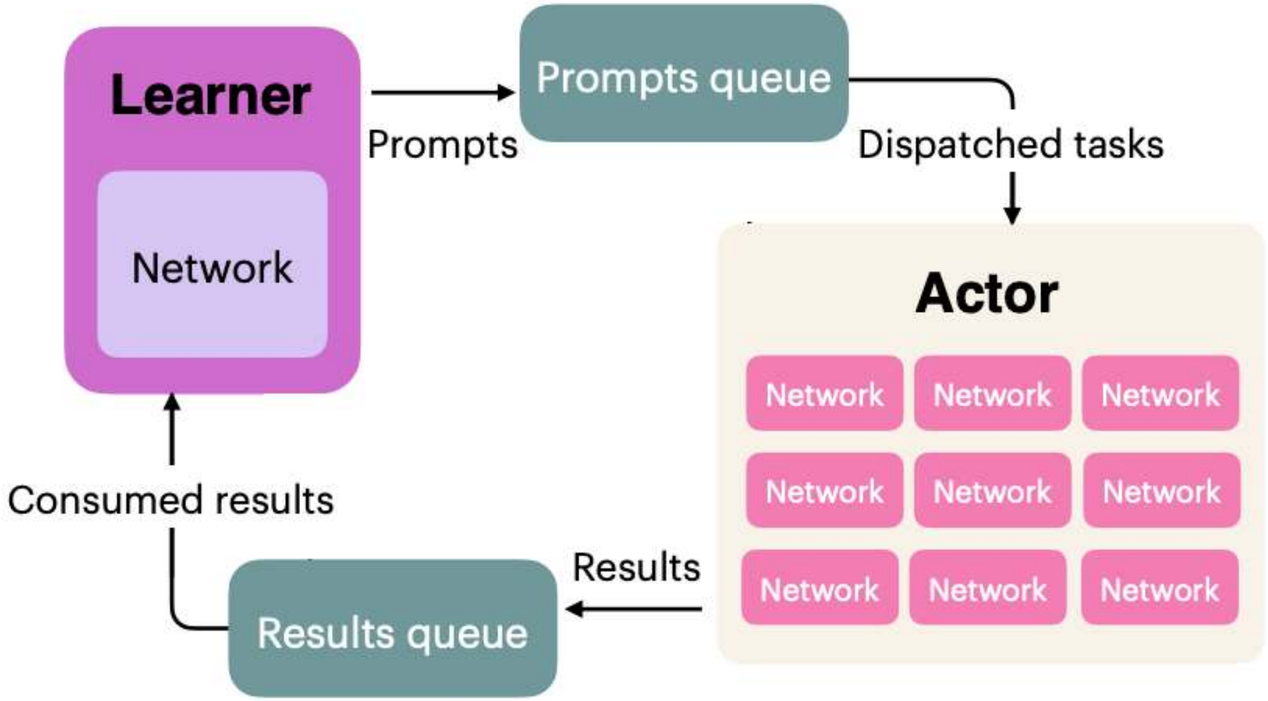
OlmoRL formulation Our final objective function includes a token-level loss, truncated importance sampling, clip-higher, and no standard deviation in the advantage calculation:

$$\mathcal{J}(\theta) = \frac{1}{\sum_{i=1}^{G}|y_i|}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|}\min\left(\frac{\pi\left(y_{i,t} \mid x, y_{i,<t}; \theta_{\text{old}}\right)}{\pi_{\text{vllm}}\left(y_{i,t} \mid x, y_{i,<t}; \theta_{\text{old}}\right)}, \rho\right)\min\Big(r_{i,t}\,A_{i,t},\ \text{clip}\left(r_{i,t},\ 1-\varepsilon_{\text{low}},\ 1+\varepsilon_{\text{high}}\right)A_{i,t}\Big), \quad (1)$$

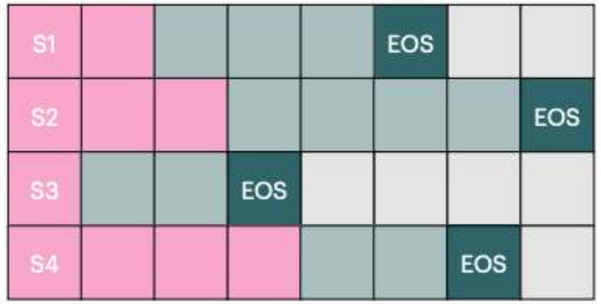
where $r_{i,t} = \frac{\pi(y_{i,t}|x,y_{i,<t};\theta)}{\pi(y_{i,t}|x,y_{i,<t};\theta_{\text{old}})}$, ε_{low} and $\varepsilon_{\text{high}}$ are the clipping hyperparameters. Here, $y_i \sim \pi_{\text{vllm}}(\cdot \mid x; \theta_{\text{old}})$ and $\pi_{\text{vllm}}(\cdot \mid x; \theta_{\text{old}})$ are the token probabilities returned from vLLM, ρ is the truncated importance sampling cap value (Yao et al., 2025), and the advantage $A_{i,t}$ for the t-th token t in the response y_i is calculated within the group G based on the relative reward of the outputs inside each group:

$$A_{i,t} = \left(r\left(x, y_i\right) - \text{mean}\left(\left\{r\left(x, y_i\right)\right\}_{i=1}^{G}\right)\right). \quad (2)$$

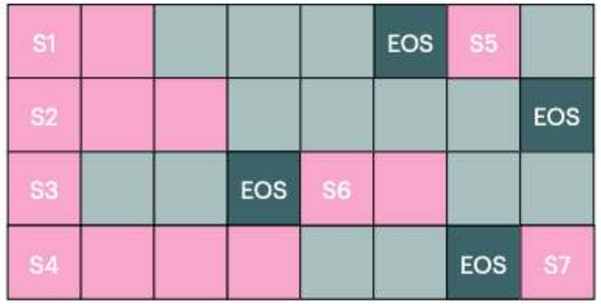
Ensure you are using an asynchronous RL framework



(a) Distributed reinforcement learning architecture



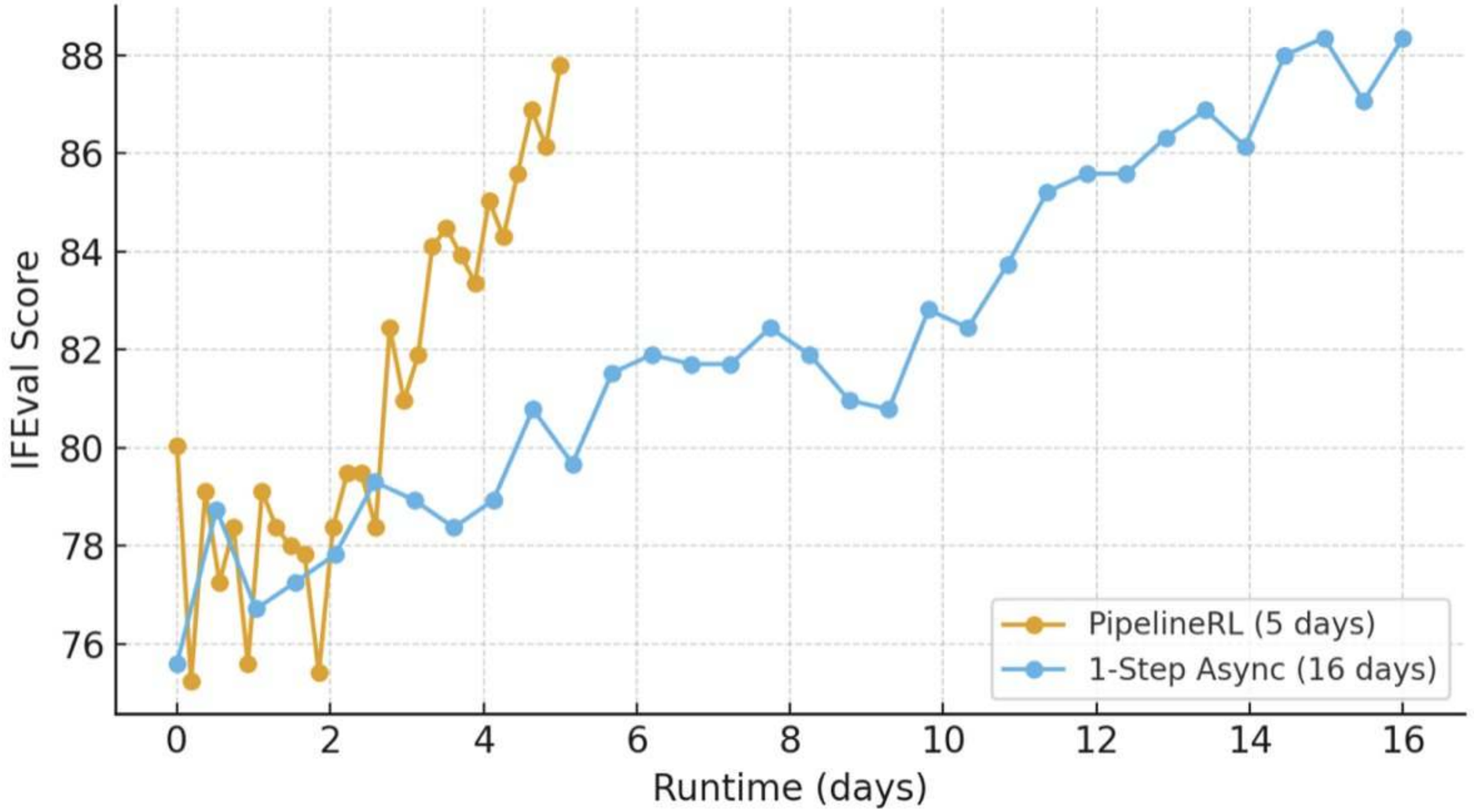
(b) Static batching



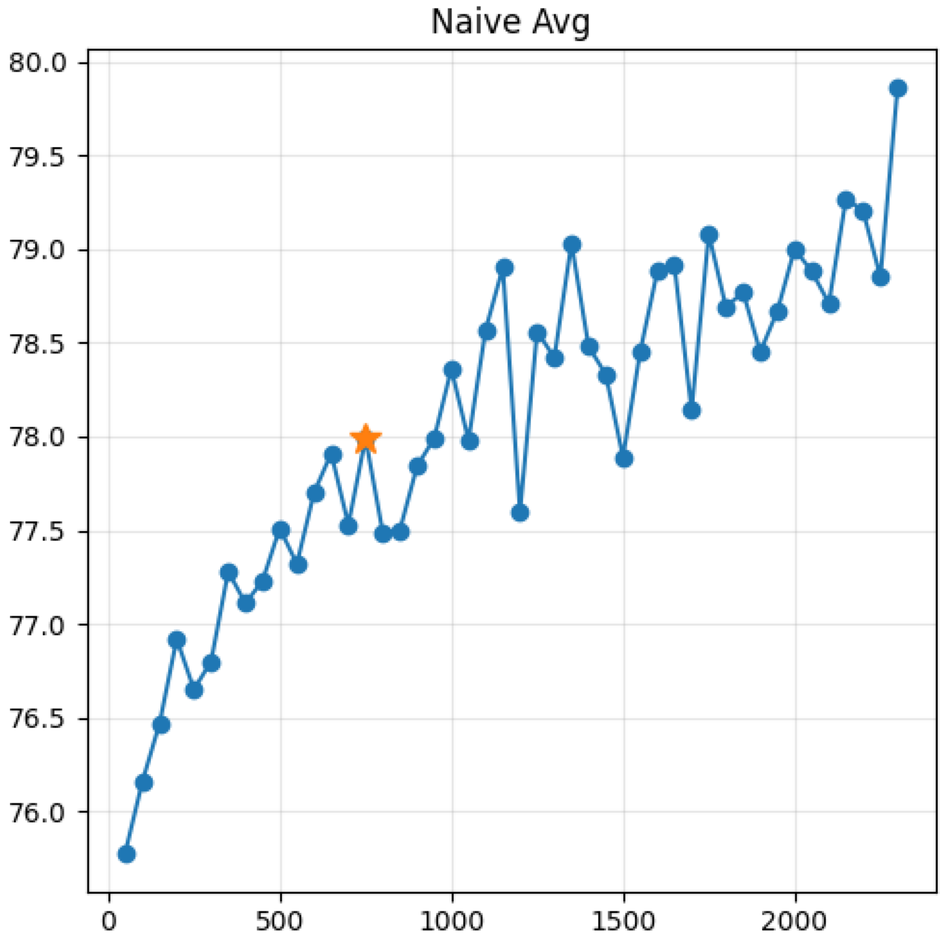
(c) Continuous batching

open-instruct Public

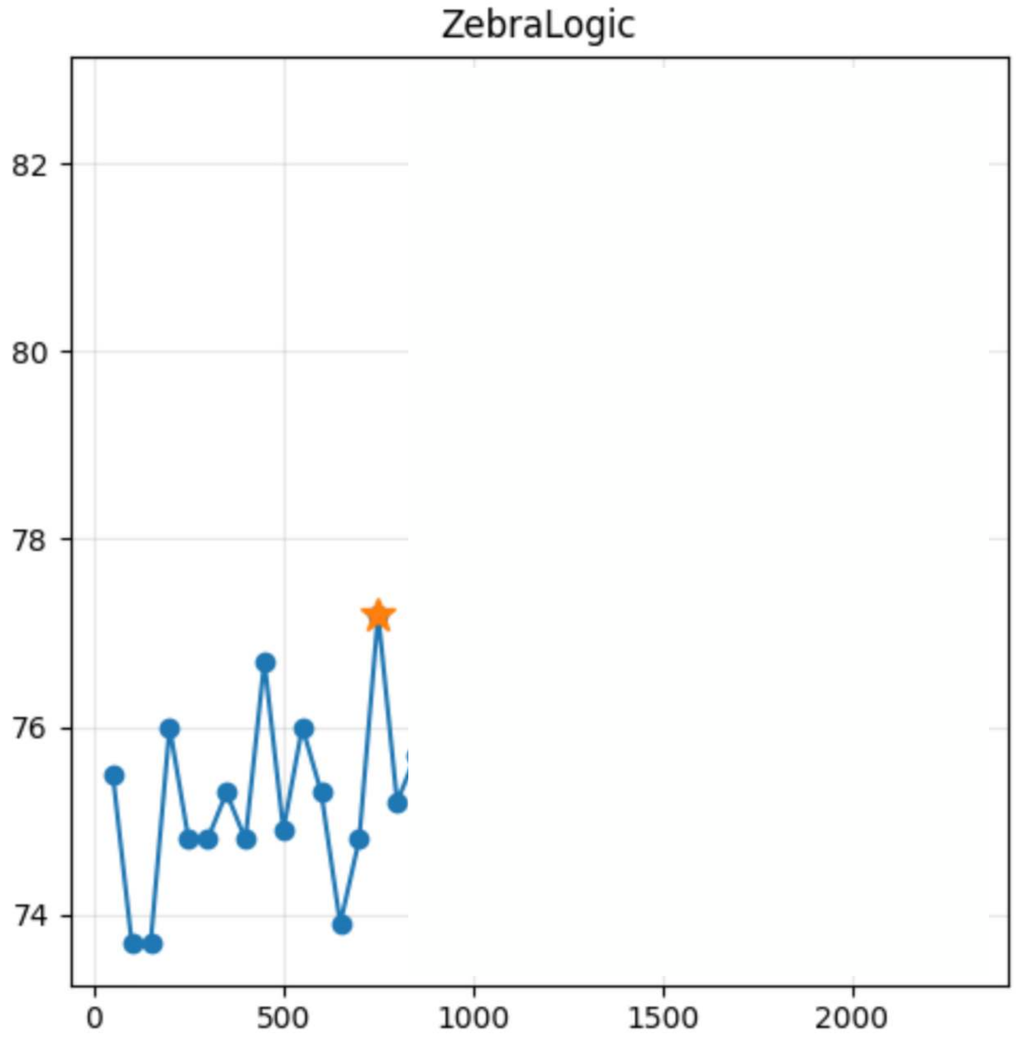
Ensure you are using an asynchronous RL framework



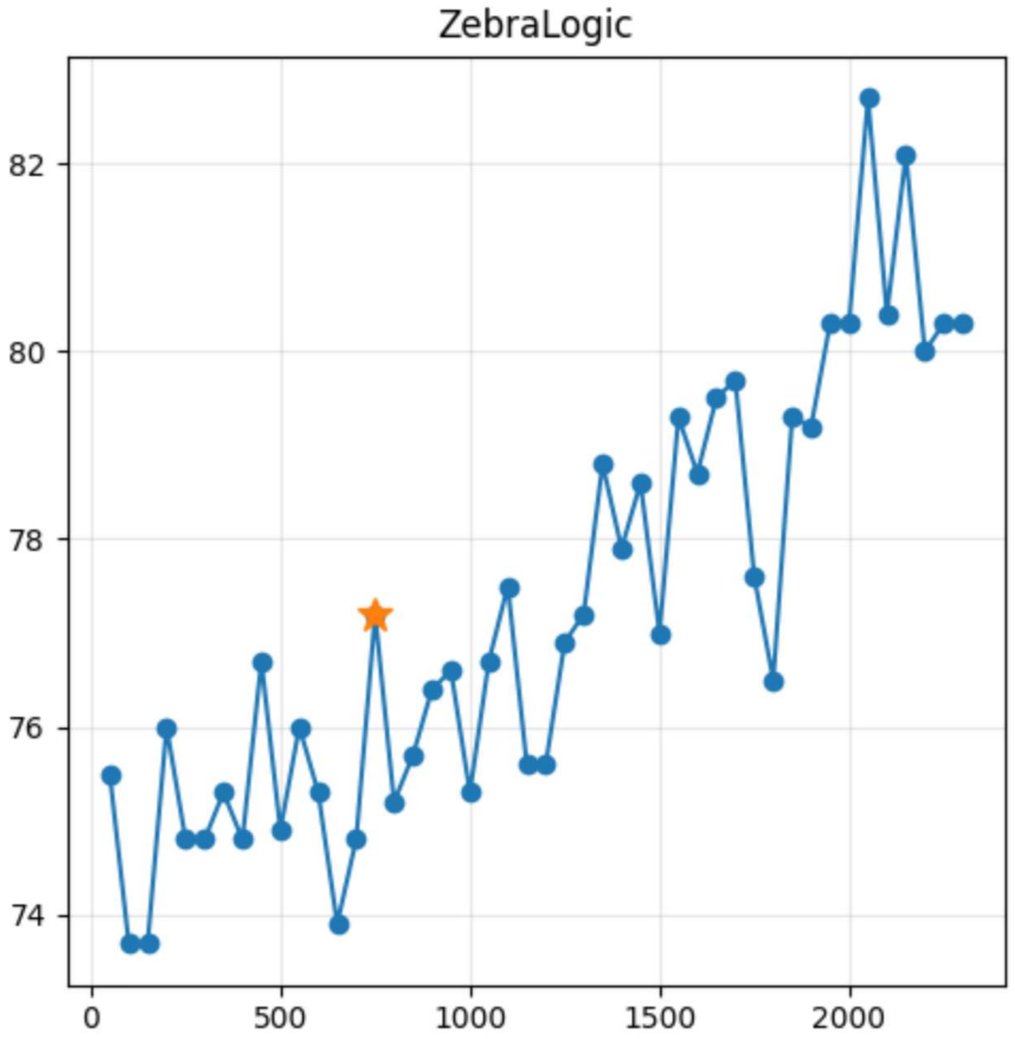
But still be prepared to wait a while



And never give up hope



And never give up hope



Thanks for listening!

Models and training code for Olmo 3:

https://github.com/allenai/open-instruct

https://huggingface.co/collections/allenai/olmo-3

Olmo 3, DR Tulu, and other of my work can be found at:

https://ivison.id.au/

Happy to answer further questions or generally chat:

hamishiv@cs.washington.edu

@hamishivi (and most other things)

Thanks for listening!

Citations for the work discussed:

Stiennon et al., "*Learning to summarize from human feedback*," NeurIPS 2020.

Guo et al., "DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning," *Nature* 645, 633–638 (2025).

Yuksekgonul et al., "*Learning to Discover at Test Time*", arXiv preprint arXiv:2601.16175, 2026.

MiniMax. *MiniMax-M1: Scaling Test-Time Compute Efficiently with Lightning Attention*. arXiv preprint arXiv:2506.13585, 2025.

Shao et al., "*DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models*," arXiv preprint arXiv:2402.03300, 2024.

Khatri et al., "*The Art of Scaling Reinforcement Learning Compute for LLMs*," arXiv preprint arXiv:2510.13786, 2025.

Yao et al., "*Your Efficient RL Framework Secretly Brings You Off-Policy RL Training*," Feng Yao's Notion, Aug. 2025.

Piché et al., "*PipelineRL: Faster On-policy Reinforcement Learning for Long Sequence Generation*," arXiv preprint arXiv:2509.19128, 2025.

Tongyi DeepResearch Team et al., "*Tongyi DeepResearch Technical Report,*" arXiv preprint arXiv:2510.24701, 2025.

Gandhi et al., "*Cognitive Behaviors that Enable Self-Improving Reasoners, or, Four Habits of Highly Effective STaRs*," COLM 2025, arXiv:2503.01307

Team OLMo et al., "*OLMo 3*," arXiv preprint arXiv:2512.13961, 2025.

Shao et al., "*DR Tulu: Reinforcement Learning with Evolving Rubrics for Deep Research*," arXiv preprint arXiv:2511.19399, 2025.

Xiao et al., "*ReflxSearch: Simple and Free Process Rewards for Agentic Search*," *work in progress.*

Zeng et al., "*RLVE: Scaling Up Reinforcement Learning for Language Models with Adaptive Verifiable Environments*," arXiv preprint arXiv:2511.07317, 2025.

Su et al., "*ToolOrchestra: Elevating Intelligence via Efficient Model and Tool Orchestration*," arXiv preprint arXiv:2511.21689, 2025.