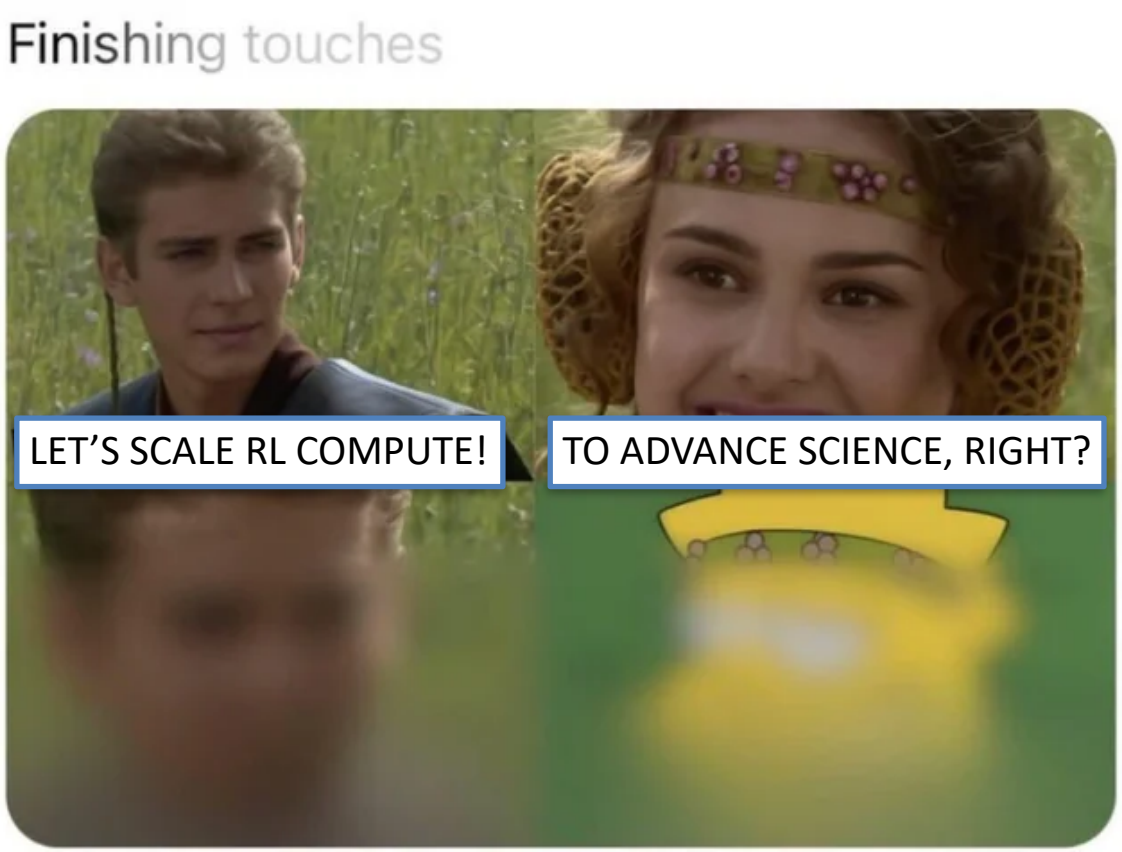
The Hitchhiker’s Guide to Frontier Reinforcement Learning

Rishabh Agarwal
Reinforcement Learner @ Periodic Labs
ICLR 2026 (Workshop on Scaling Post-Training for LLMs)

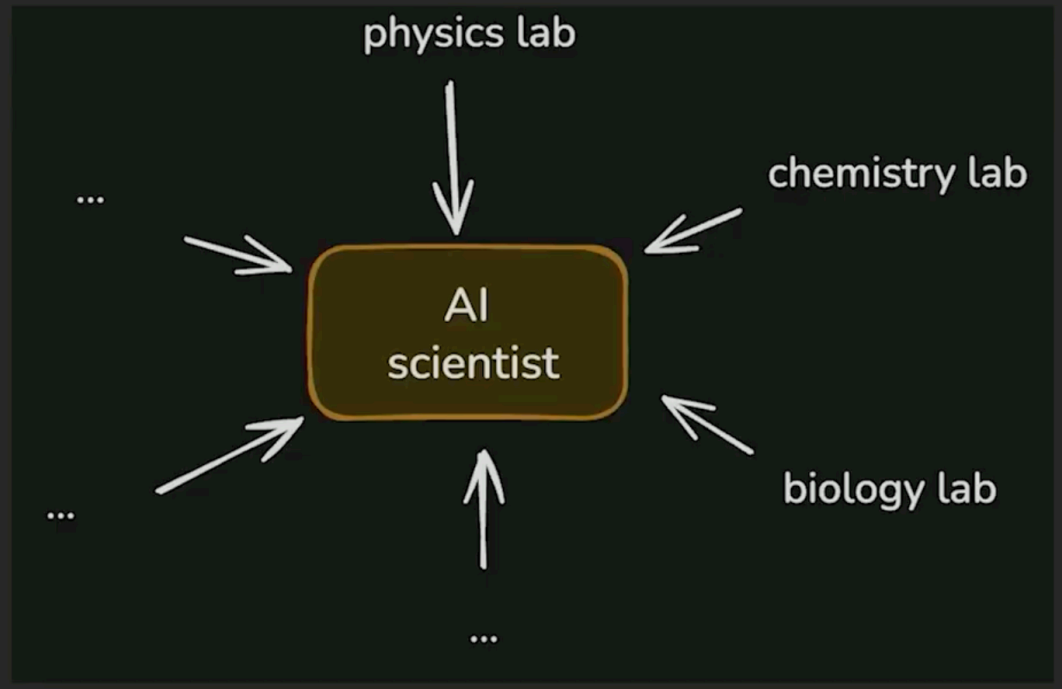
Accelerating science is the most valuable use of AI



Periodic Labs: RL in the physical sciences

To create an AI scientist, you have to *do* science.

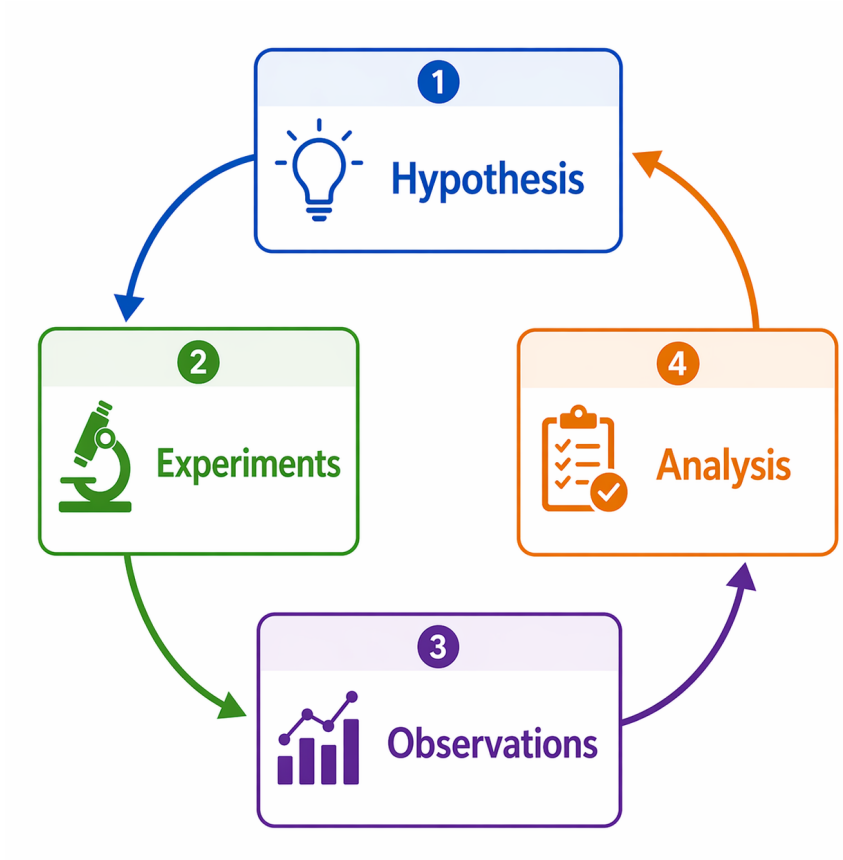
In our case, nature is the RL environment



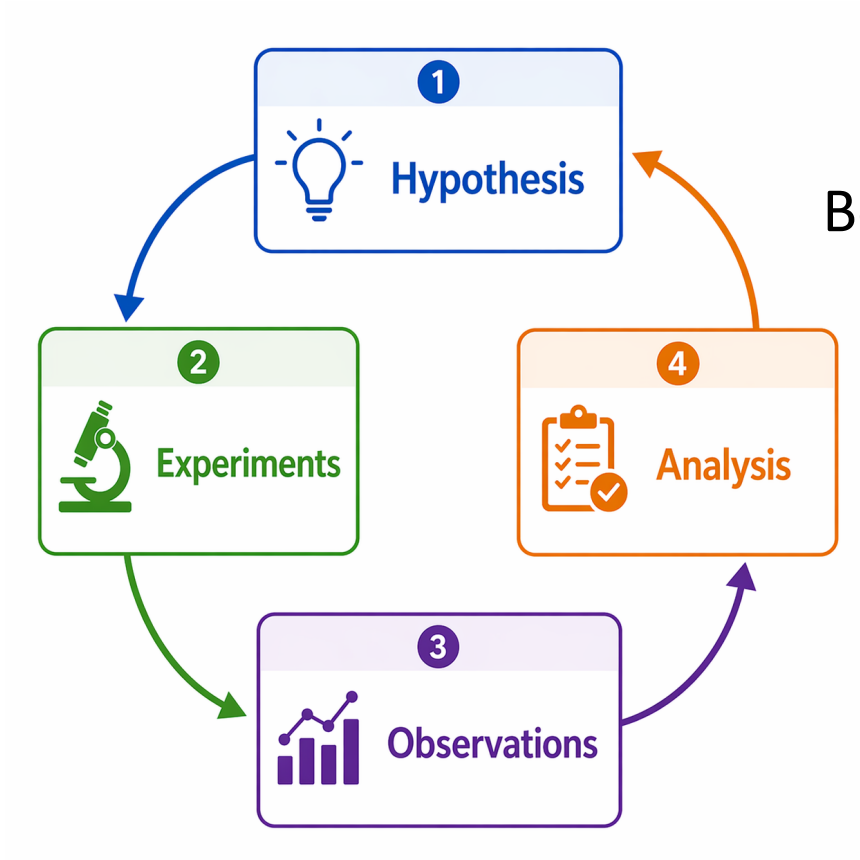
Technological progress is limited by our ability to design the physical world.

GPU Mode Keynote, Liam Fedus

Scientific Discovery in the Physical Sciences

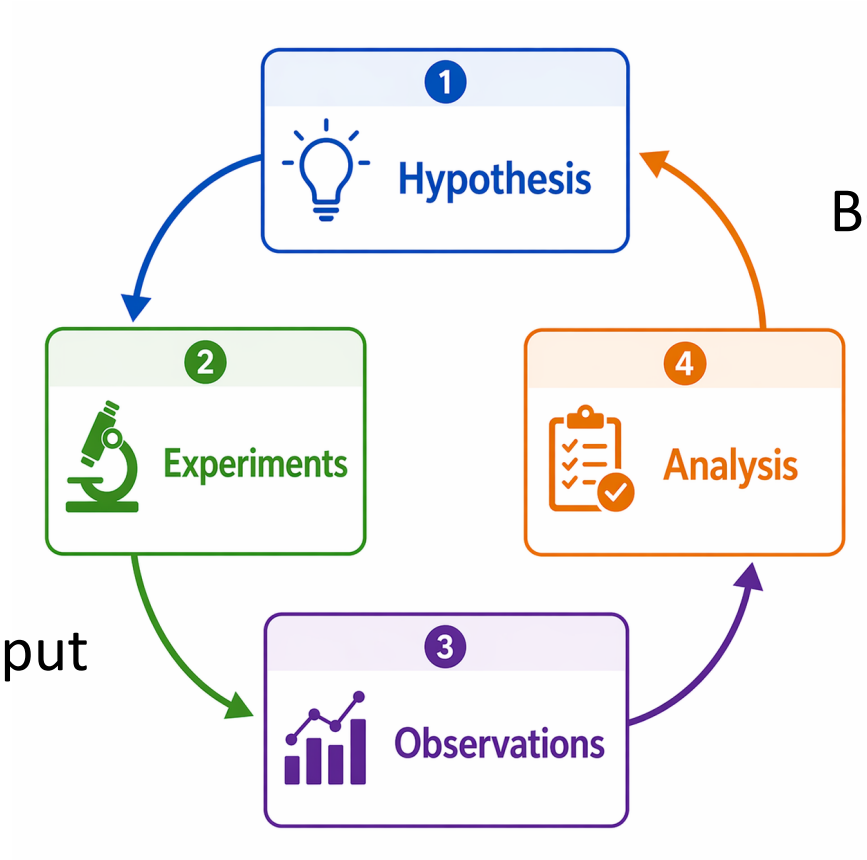


Scientific Discovery in the Physical Sciences



Bottlenecked by “intelligence”

Scientific Discovery in the Physical Sciences

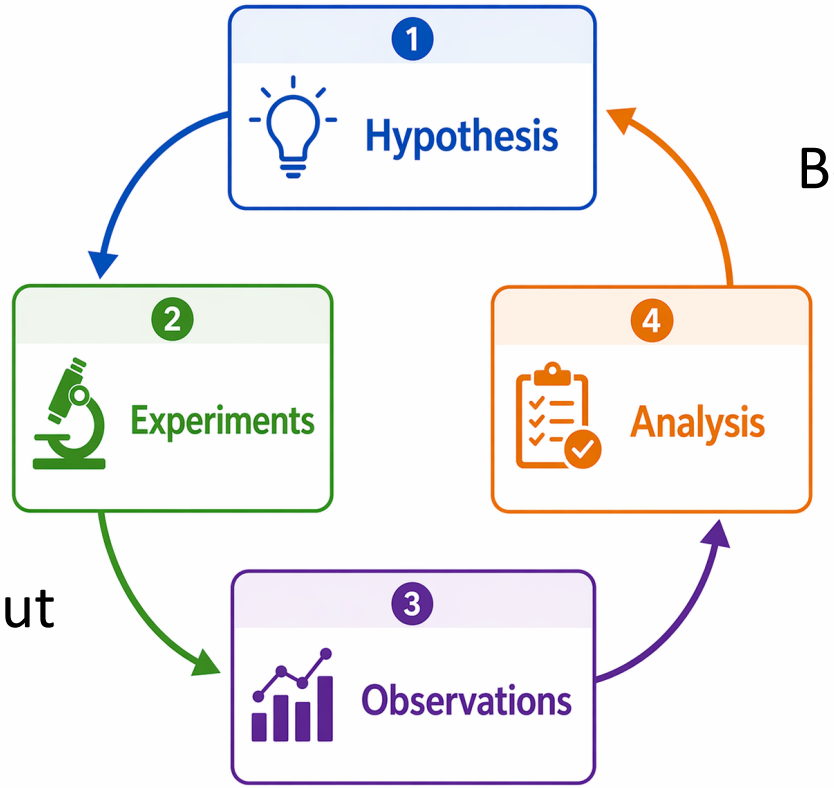


Bottlenecked by “intelligence”

Bottlenecked by high-throughput autonomous laboratories

Scientific Discovery in the Physical Sciences

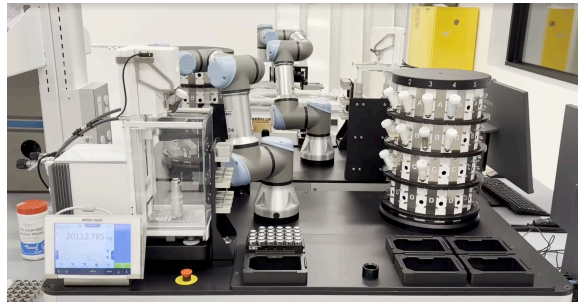
Bottlenecked by high-throughput autonomous laboratories



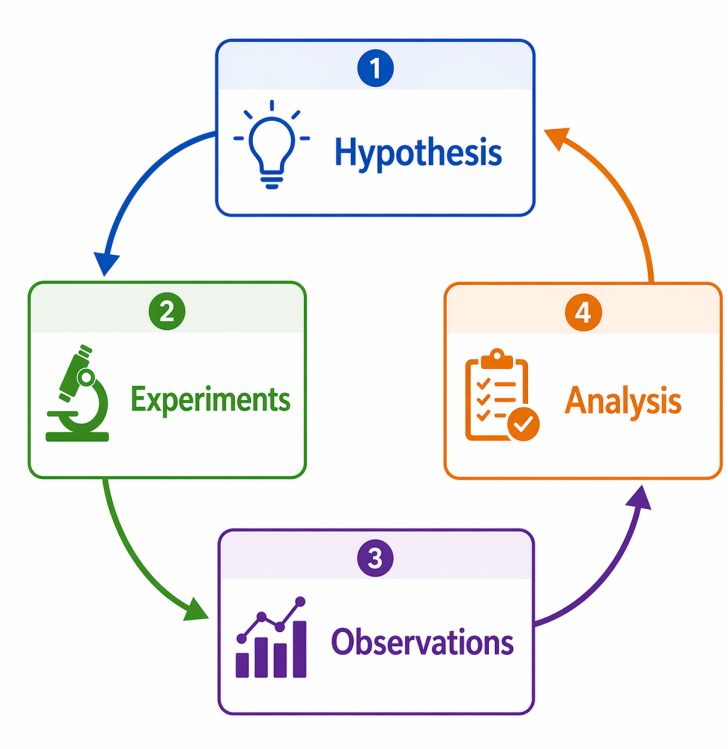
Bottlenecked by “intelligence”

Intelligence is necessary, but not sufficient. New knowledge is created when ideas are found to be consistent with reality. And so, at Periodic, we are building AI scientists *and* the autonomous laboratories for them to operate.

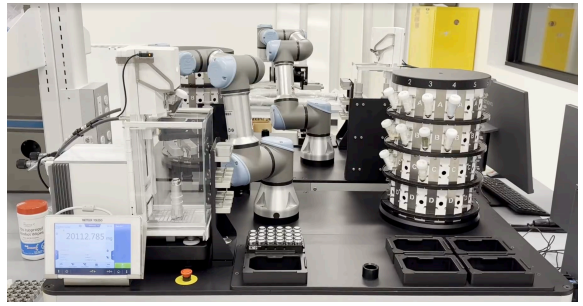
Scientific Discovery @ Periodic Labs



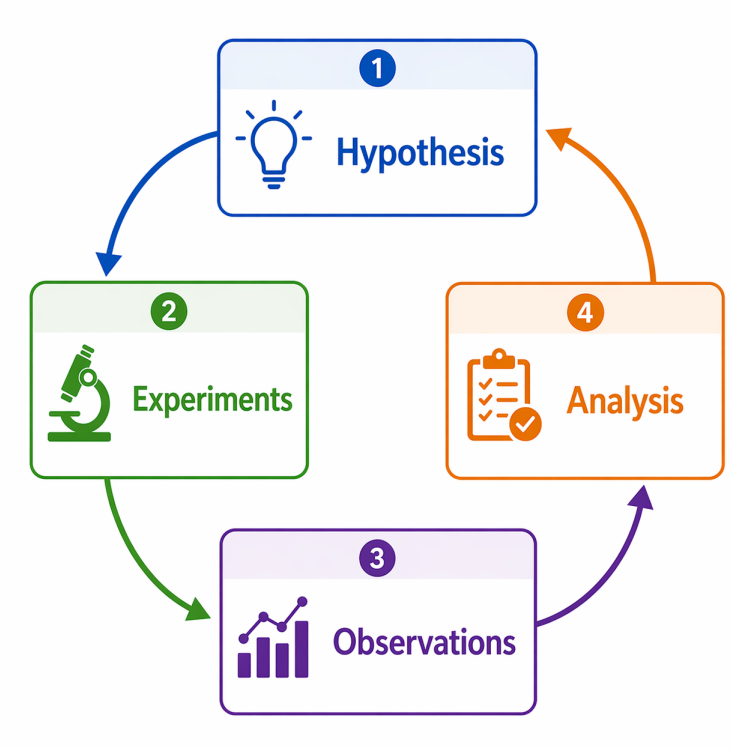
Autonomous laboratory

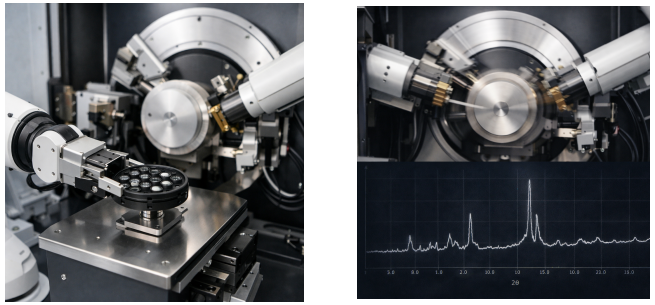


Scientific Discovery @ Periodic Labs



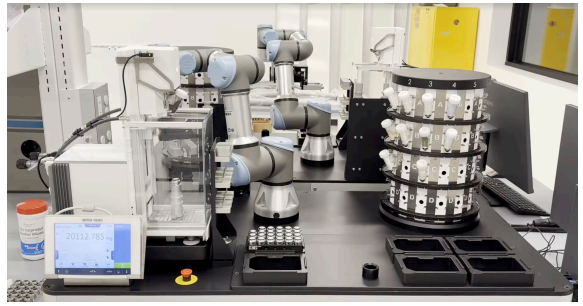
Autonomous laboratory



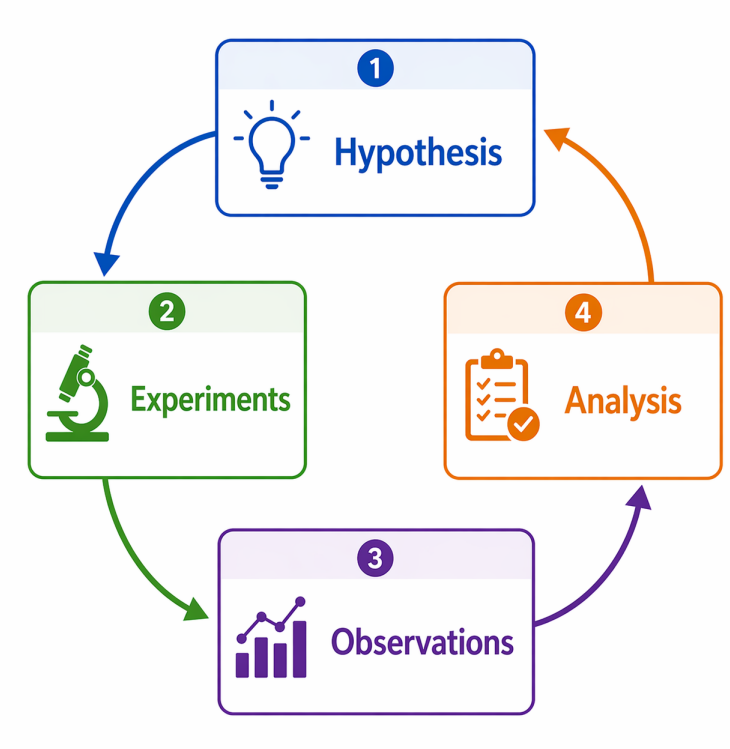


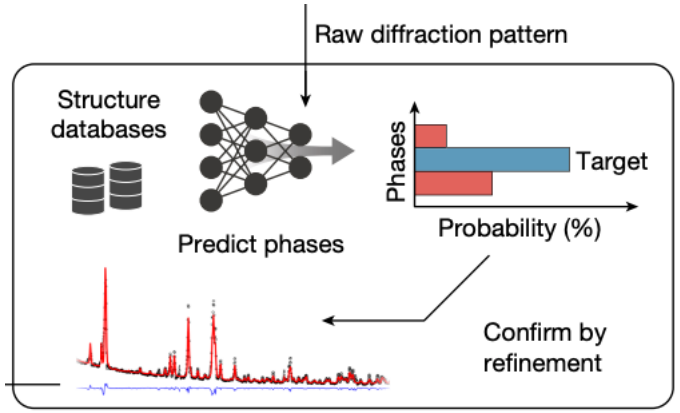
Materials Characterization (e.g., XRD)

Scientific Discovery @ Periodic Labs

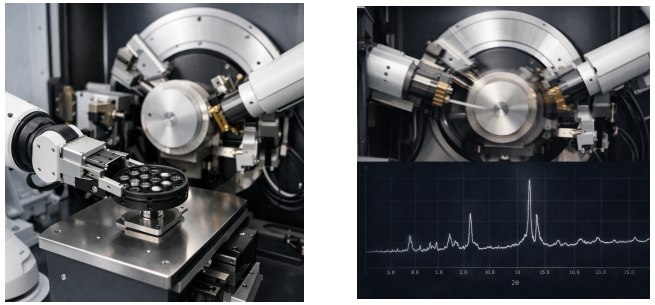


Autonomous laboratory



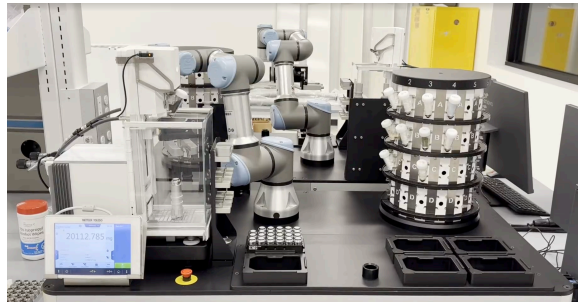


Automated Phase Identification



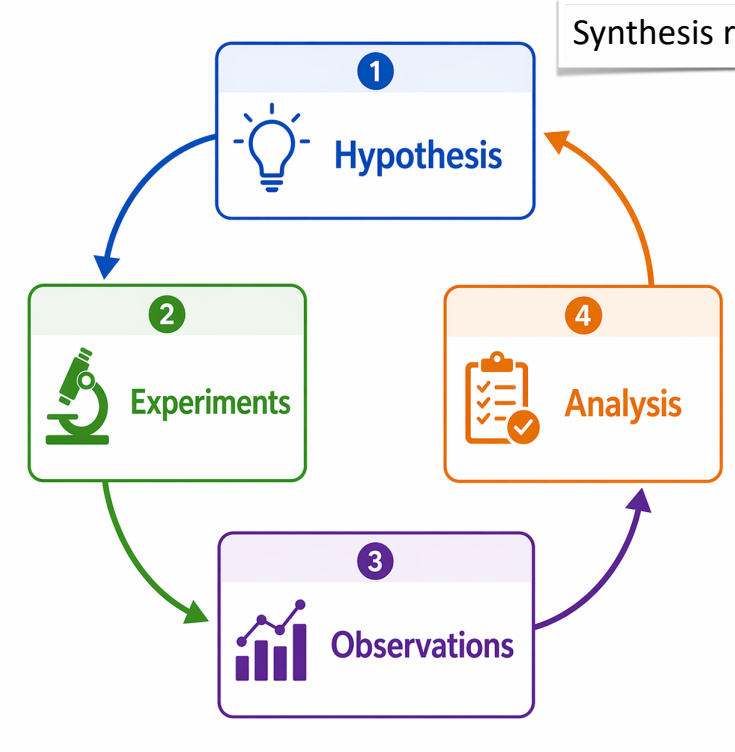
Materials Characterization (e.g., XRD)

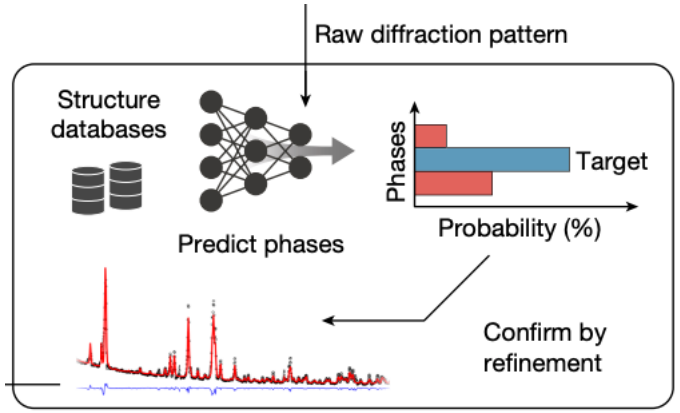
Scientific Discovery @ Periodic Labs



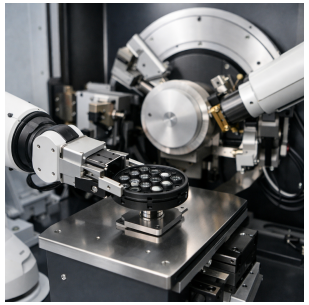
Autonomous laboratory

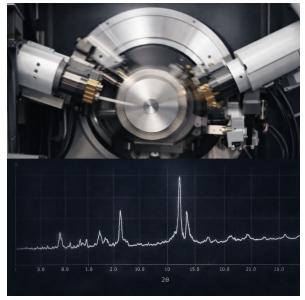
Synthesis recipe for a “new” superconductor





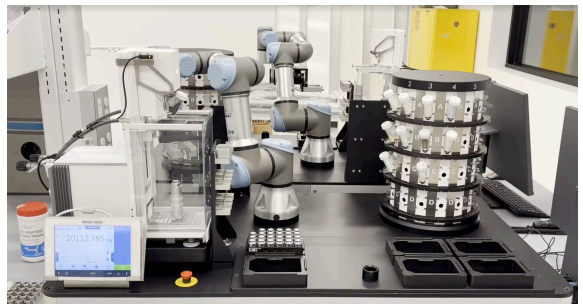
Automated Phase Identification





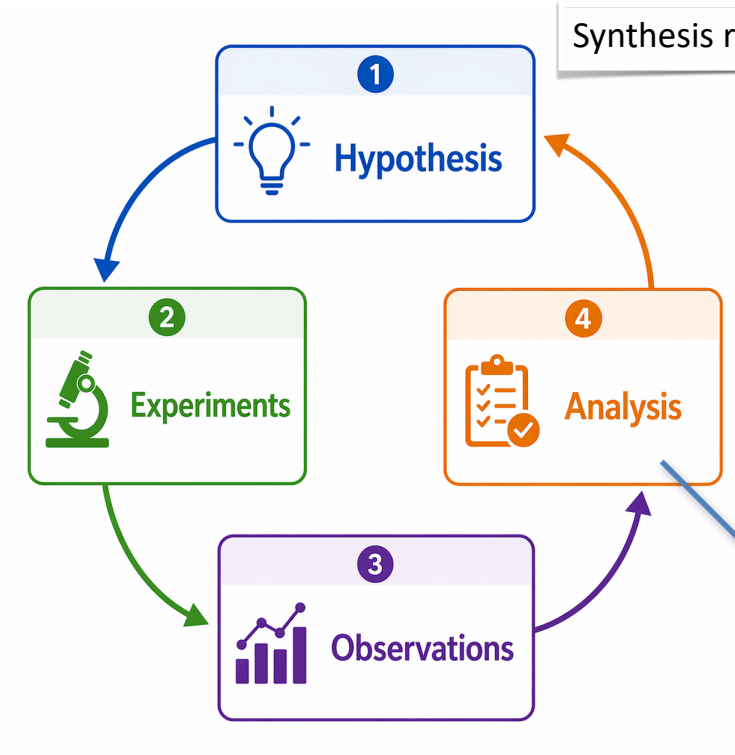
Materials Characterization (e.g., XRD)

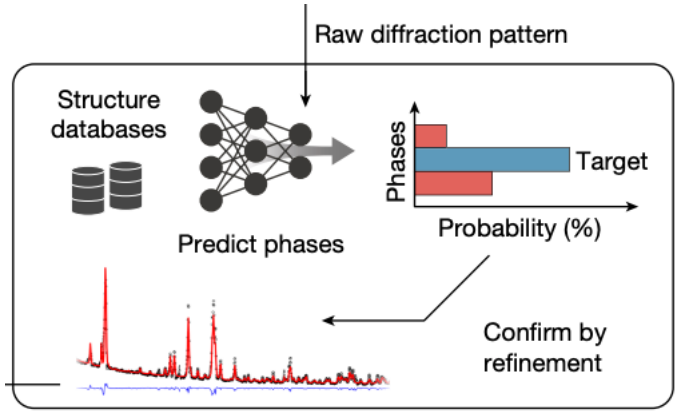
Scientific Discovery @ Periodic Labs



Autonomous laboratory

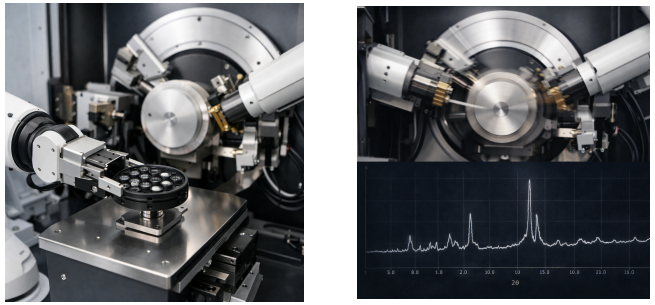
Synthesis recipe for a “new” superconductor





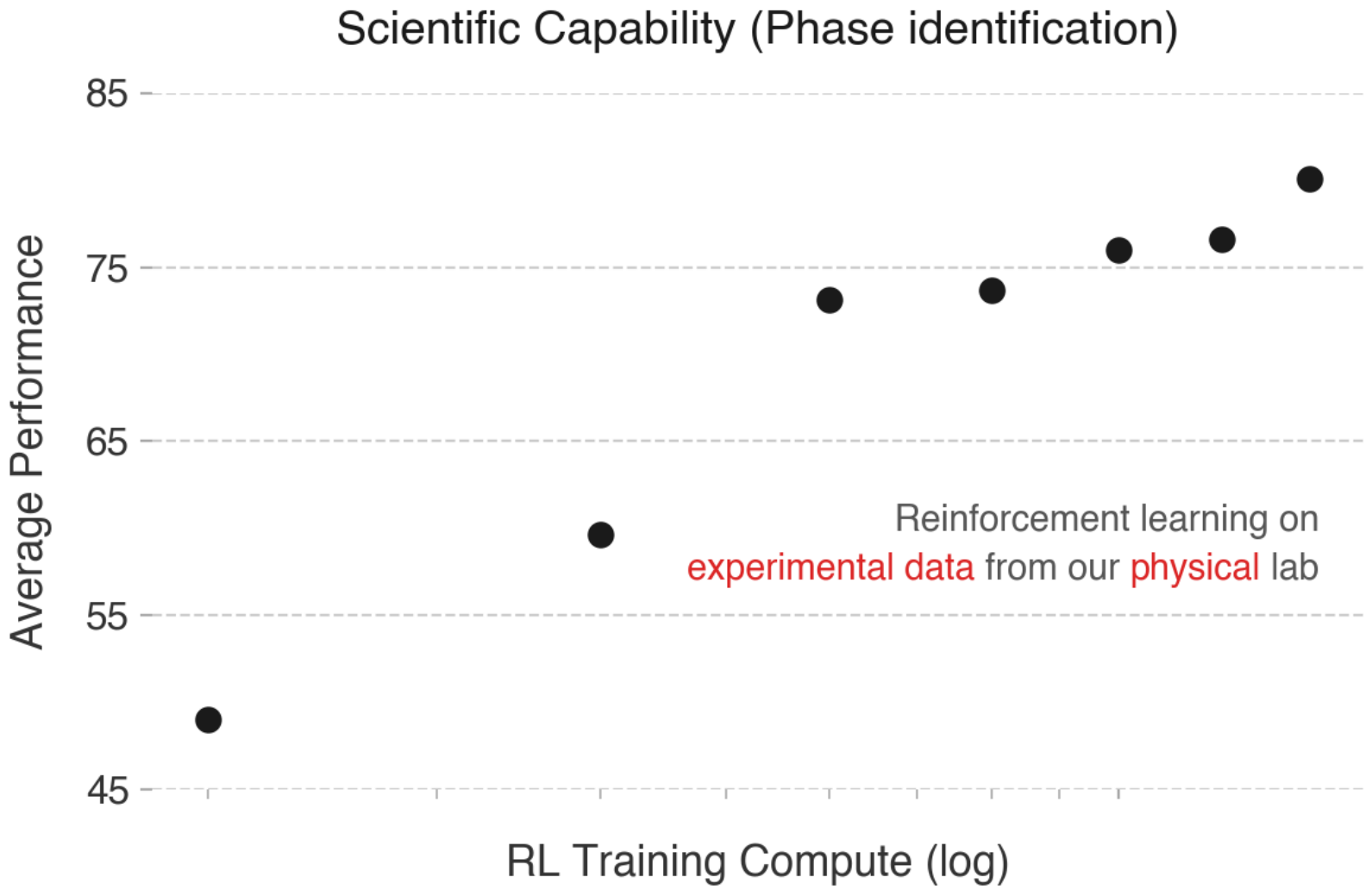
Automated Phase Identification

Let’s start here, important bottleneck and (weakly) verifiable!

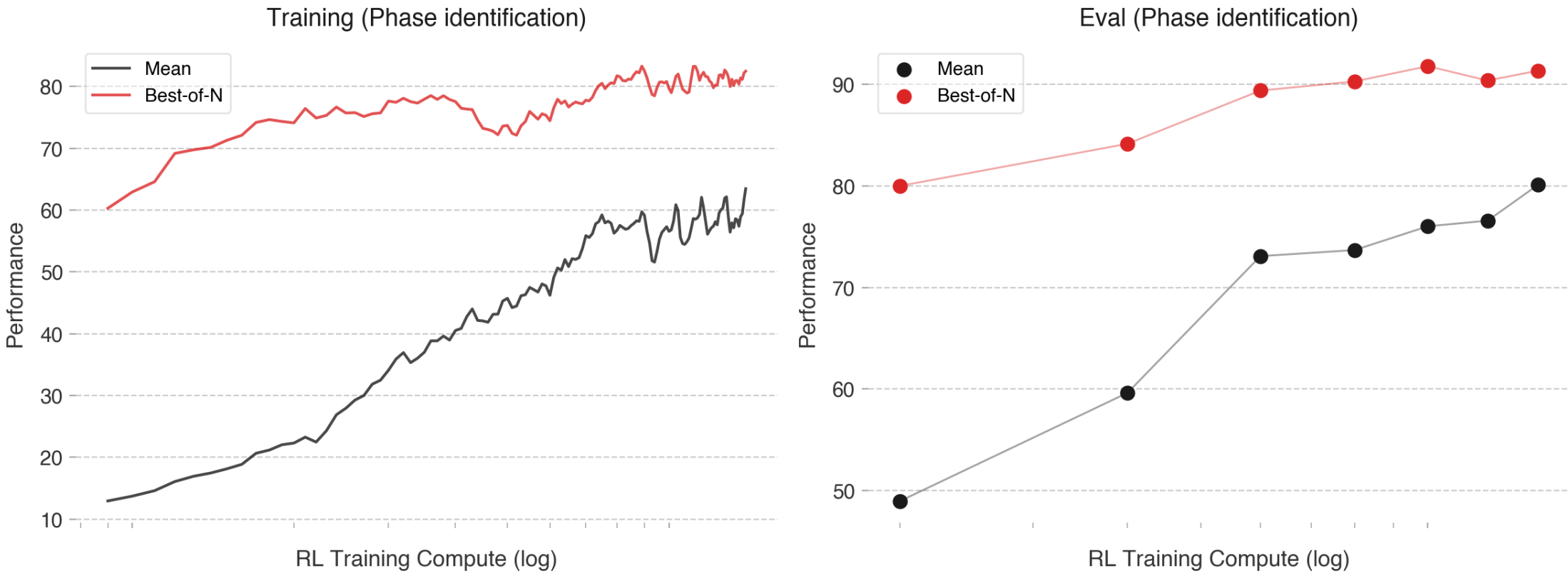


Materials Characterization (e.g., XRD)

Scaling RL Compute @ Periodic Labs

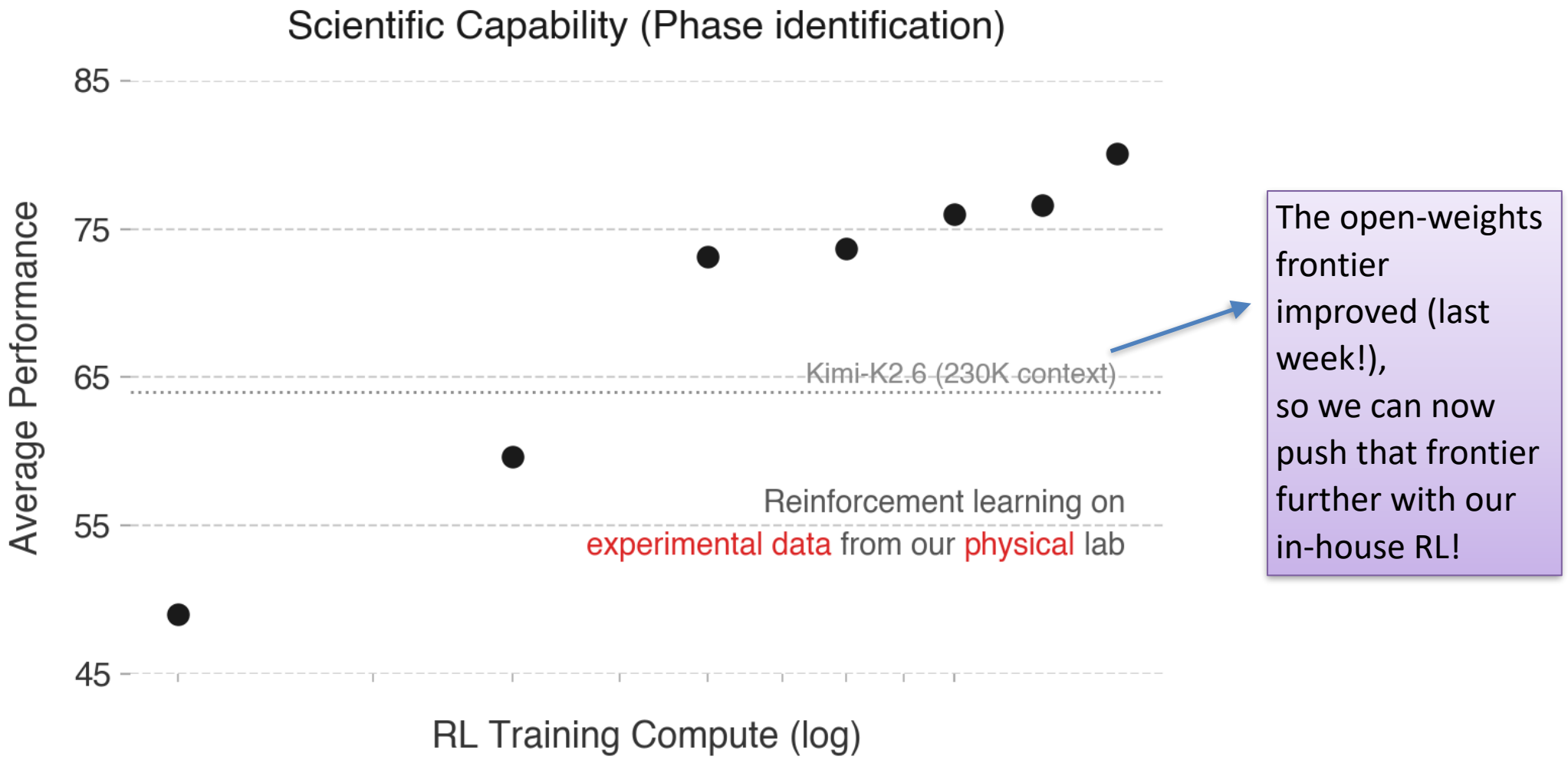


Scaling RL expands the model's capabilities



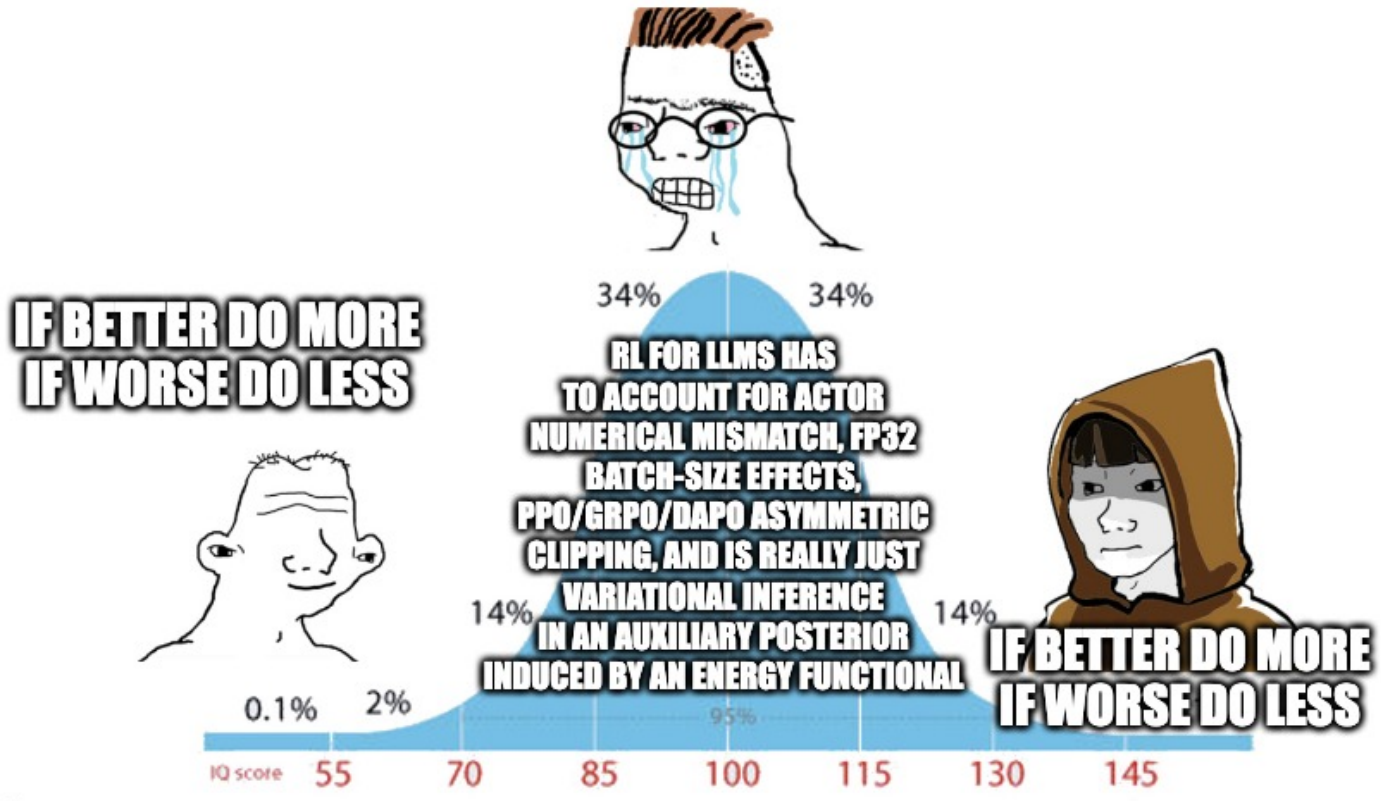
RL improves the model's effective coverage of correct solutions under repeated sampling (see Composer2 for another example)

Scaling RL Compute @ Periodic Labs



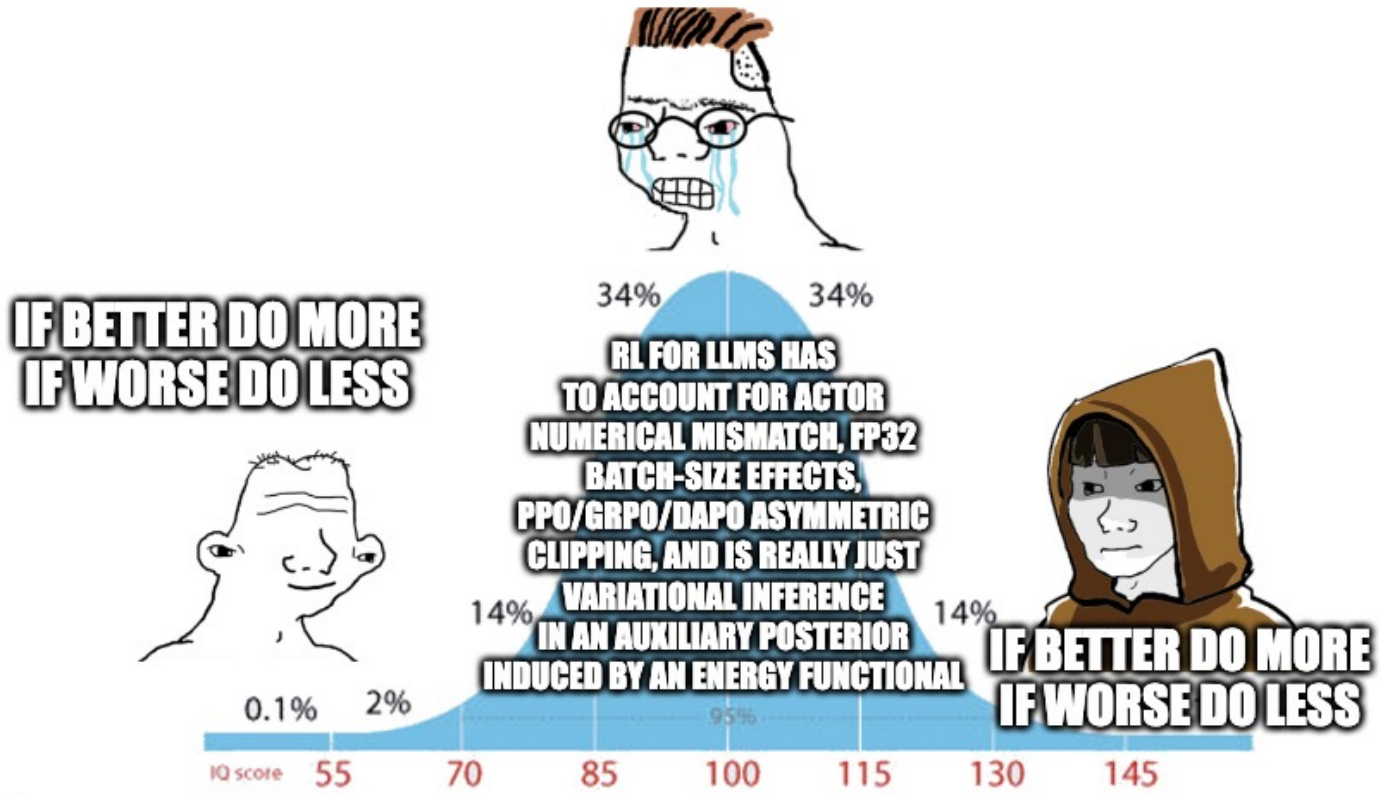
The open-weights frontier improved (last week!),
so we can now push that frontier further with our in-house RL!

This Talk: Scaling RL at the Frontier



Source: Ian Osband

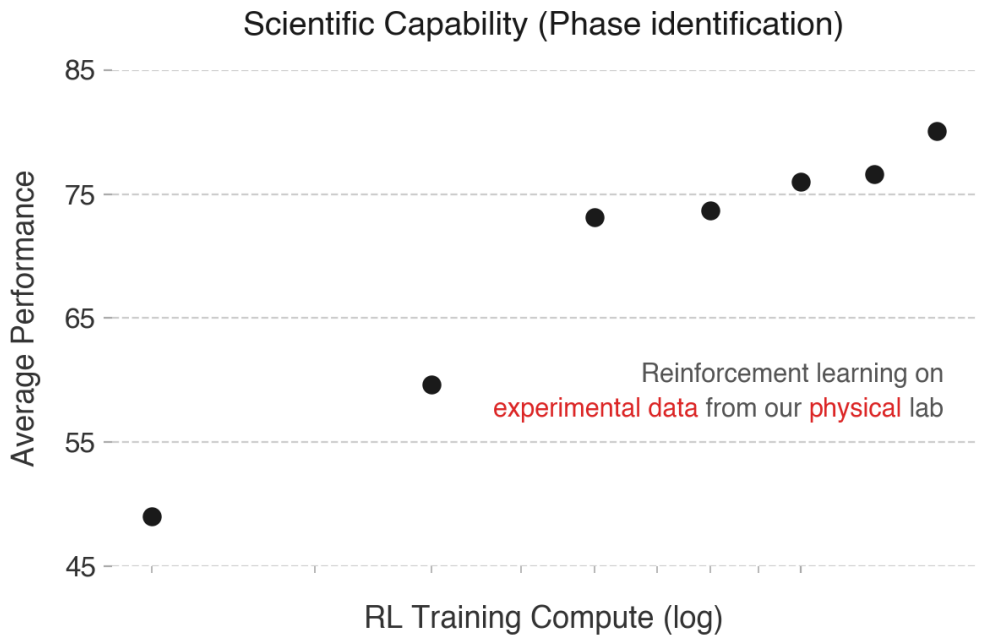
This Talk: Scaling RL at the Frontier



Source: Ian Osband

The core algorithm is simple. The engineering to make it work reliably at scale is hard.

This Talk: Scaling RL at the Frontier

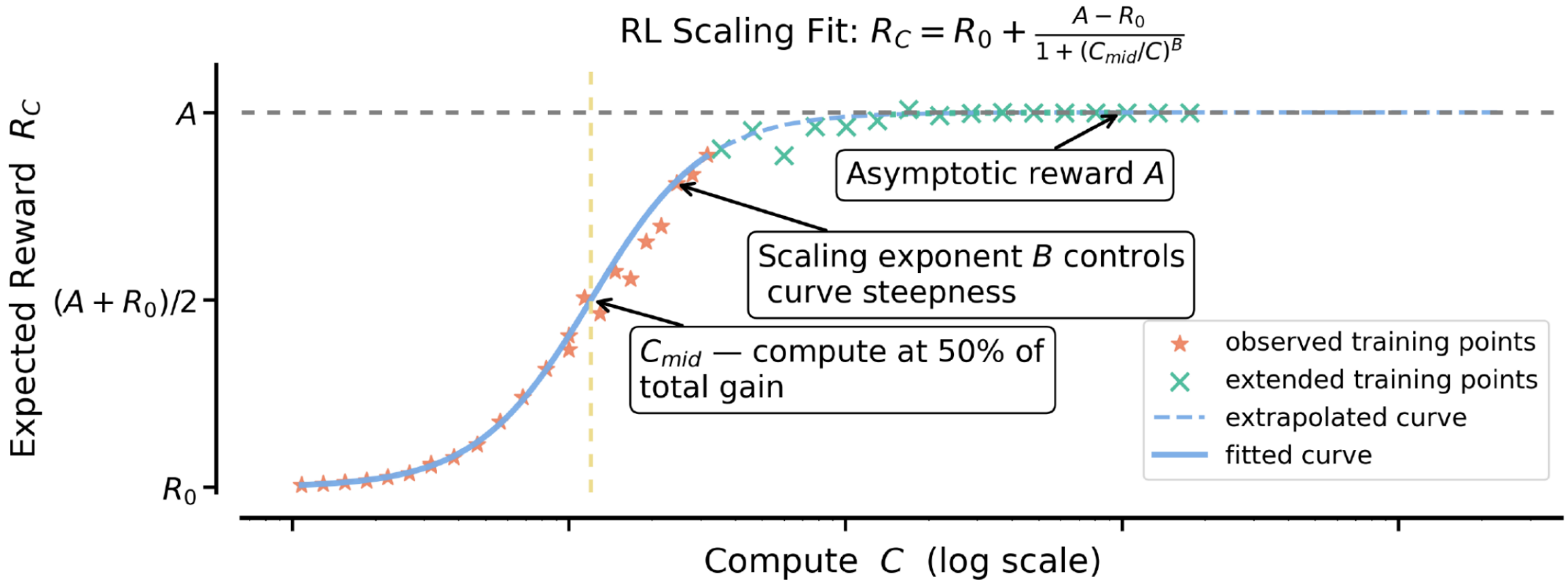


Stable RL for Large MoEs (1T+ params) requires several infra pieces!

- Dealing with train-inference discrepancy
- Asynchronous RL
- “Easy-to-tune” and stable algorithm

Grateful for OSS libraries such as Megatron, vLLM, and SGLang that we heavily benefit from as well as contribute to

Background: How RL scales with Compute



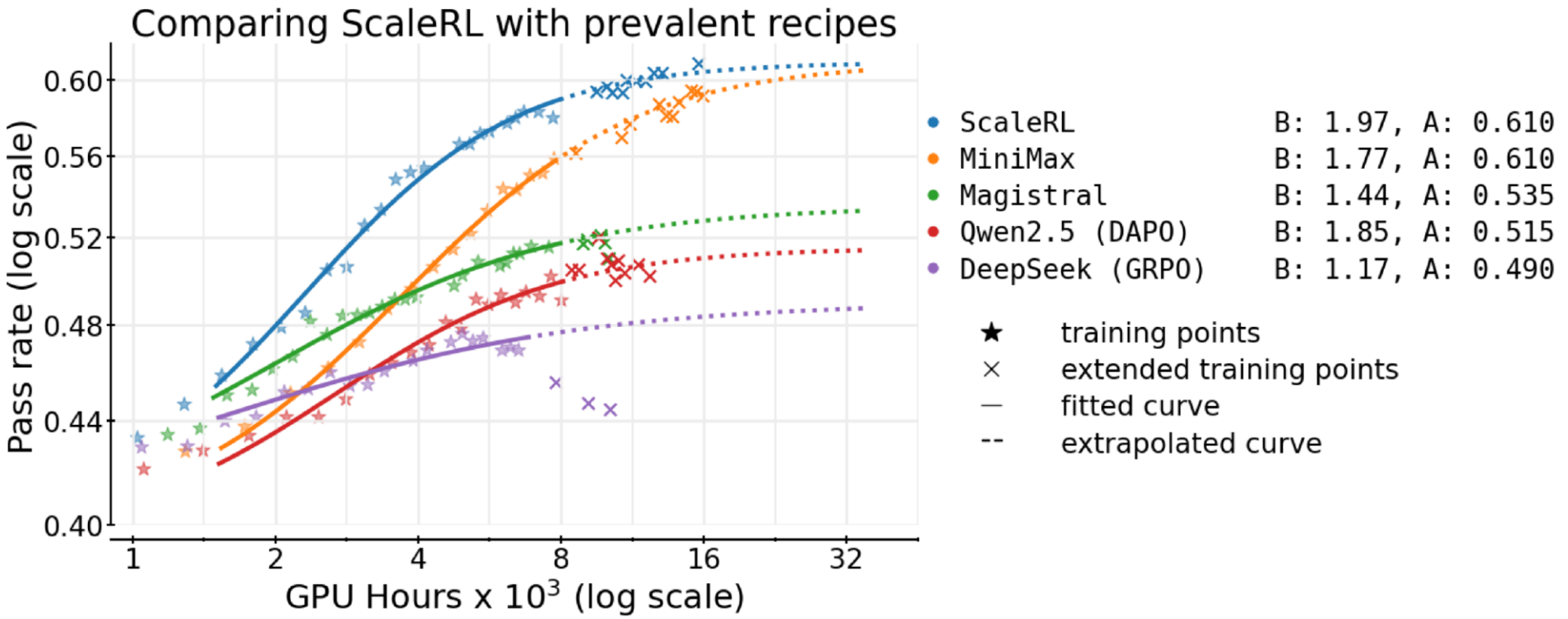
A **Asymptotic Ceiling**
How high?

B **Compute Efficiency**
How fast?

C_{mid} **Inflection**
When saturate?

[The Art of Scaling Reinforcement Learning Compute for LLMs]

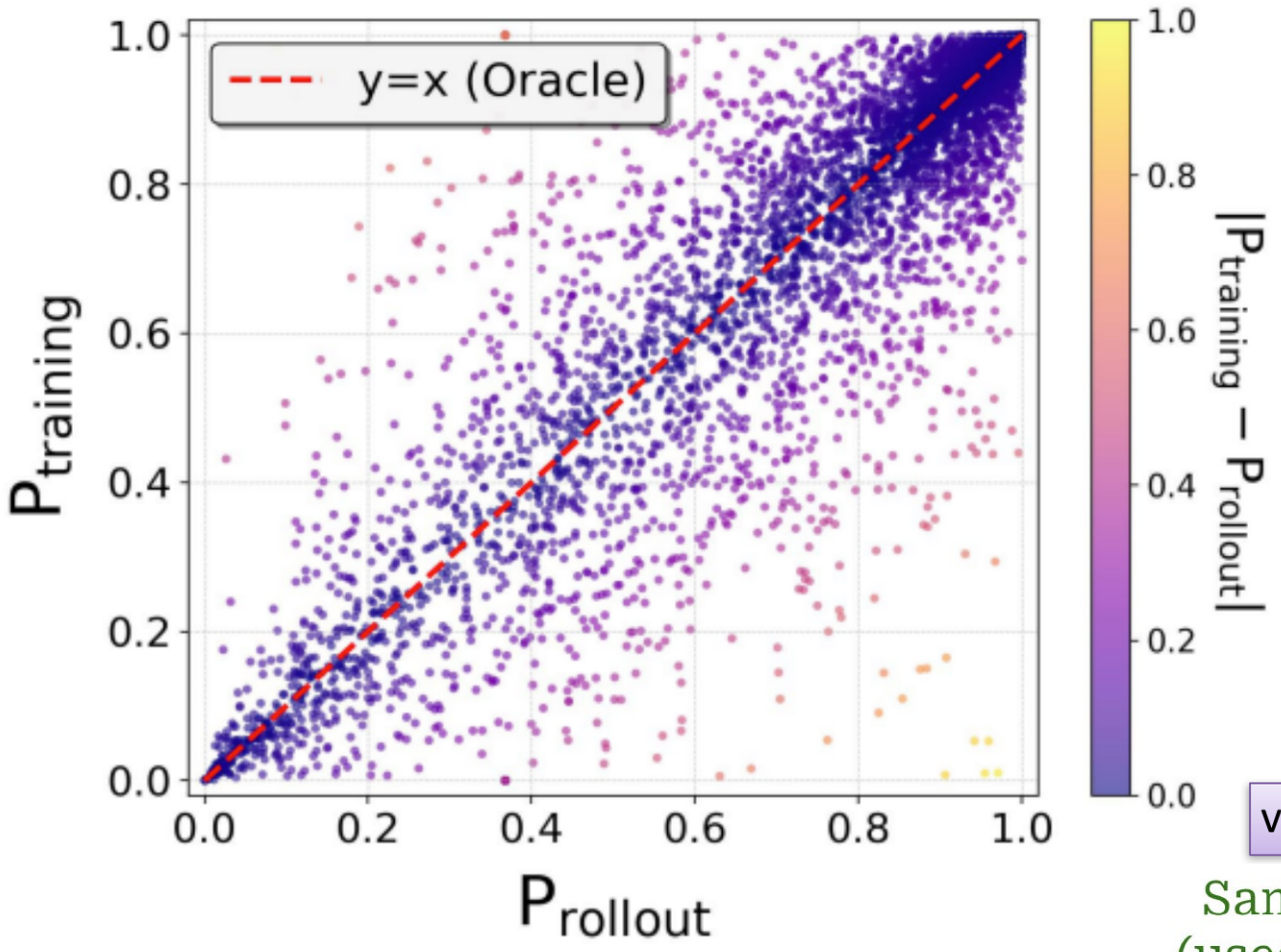
Different RL recipes, different ceilings



RL design choices do impact the ceiling you can reach and how fast you reach them!

The Art of Scaling Reinforcement Learning Compute for LLMs

Train-inference mismatch in on-policy RL



Training & inference are optimized for different purposes, resulting in mismatch due to different kernels, precision etc!

vLLM, SGLANG

Sampler Engine (used for rollouts)

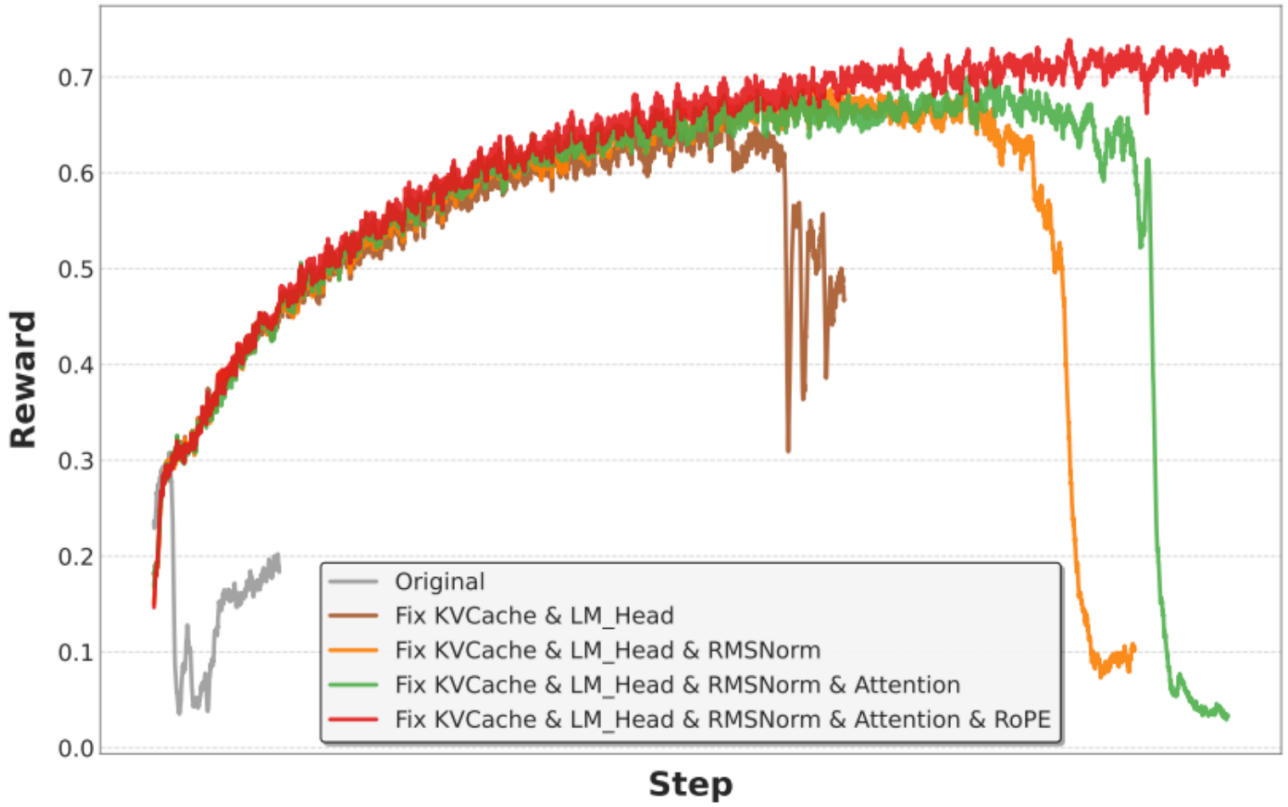
e.g., Megatron

Learner Engine (used for policy updates)

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{a\sim\overbrace{\pi_{\theta_{\text{sampler}}}}, s\sim\mathcal{D}} \left[\nabla_\theta \log \overbrace{\pi_{\theta_{\text{learner}}}}(a|s) \cdot R(a)\right]$$

https://arxiv.org/pdf/2510.19338

Train-inference mismatch is worse for MoEs



Source: https://arxiv.org/pdf/2510.19338

RL wants to work for small* LLMs, we MAKE it work for large MoEs (e.g., Kimi-K2)

*Small is subjective but typical LLMs (<100B params) in academic research

Mitigating Train-Inference Discrepancy for MoEs: Router Replay

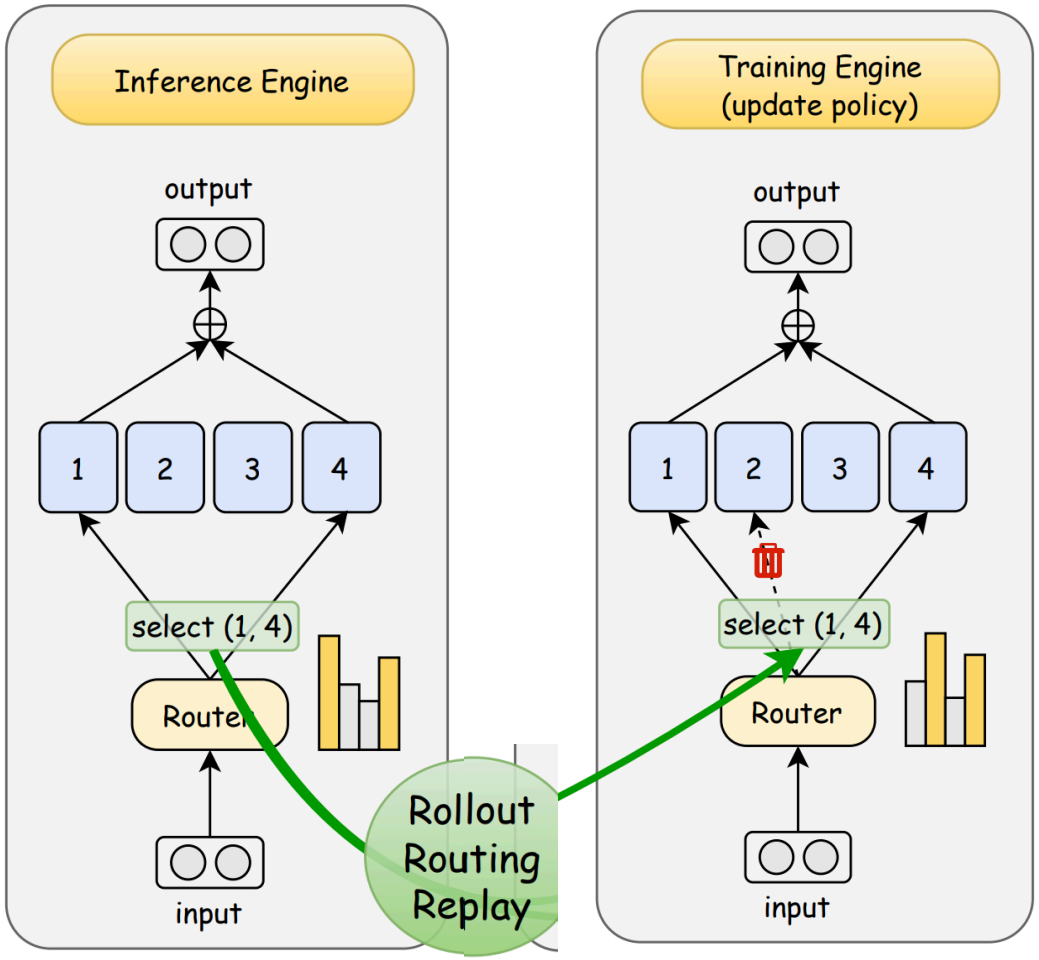
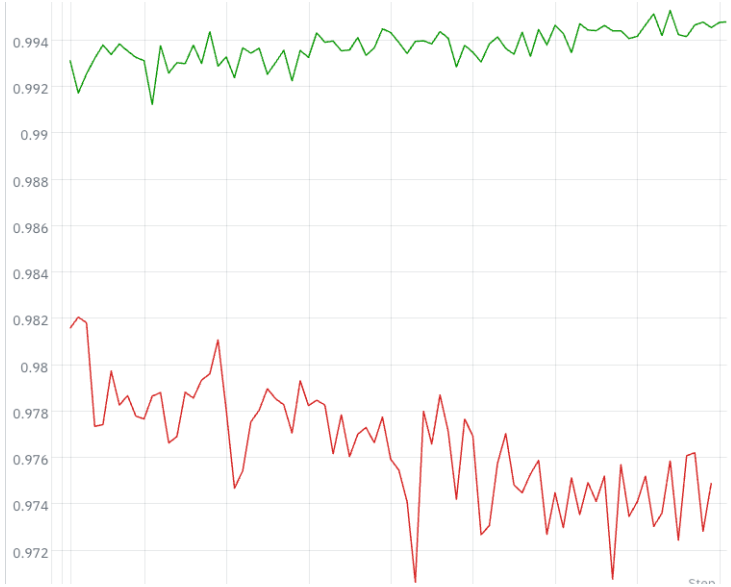


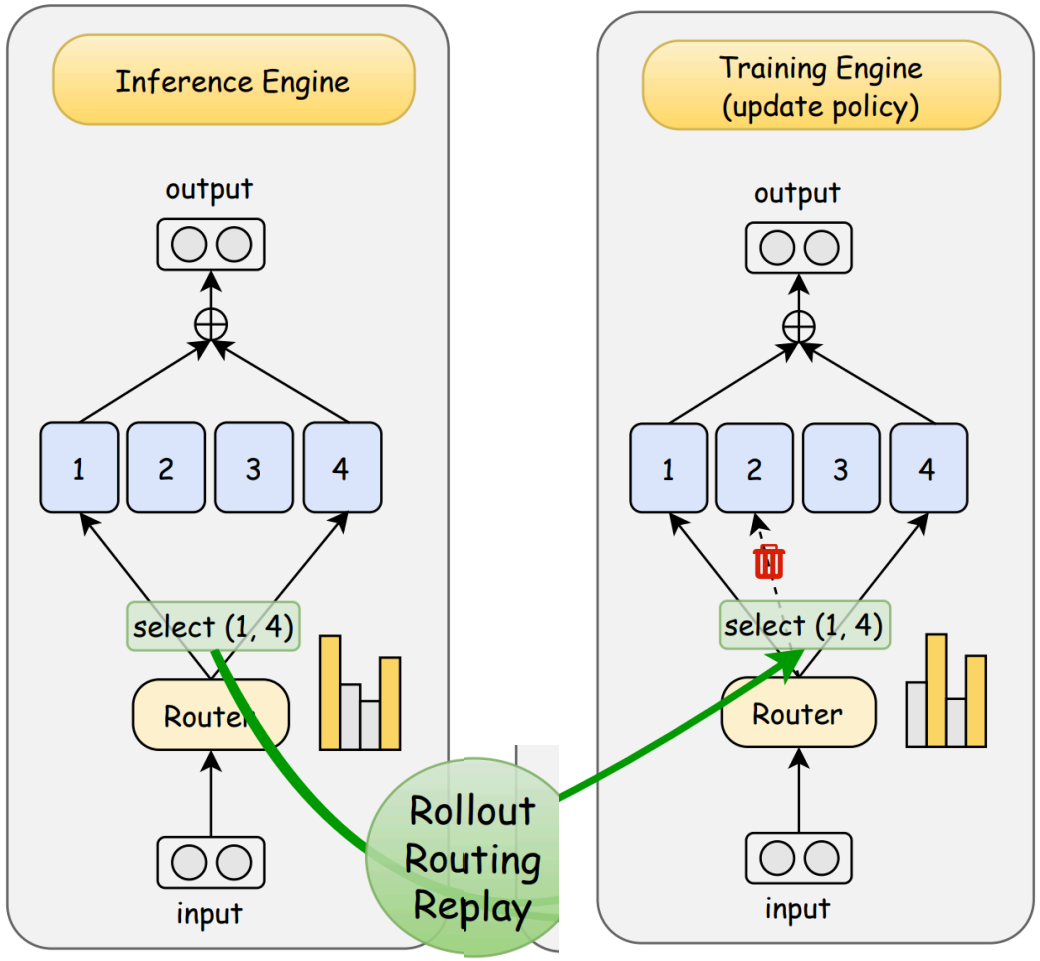
Figure modified from [this paper]

ESS for Kimi-K2 (Higher is better)

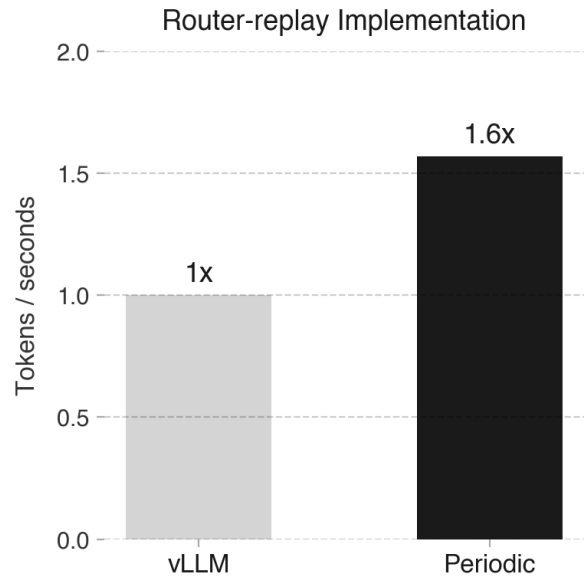


Router-replay leads to more stable RL training dynamics

No free lunch: Router Replay for MoE-RL



- Infra complexity
- Overhead in trainer



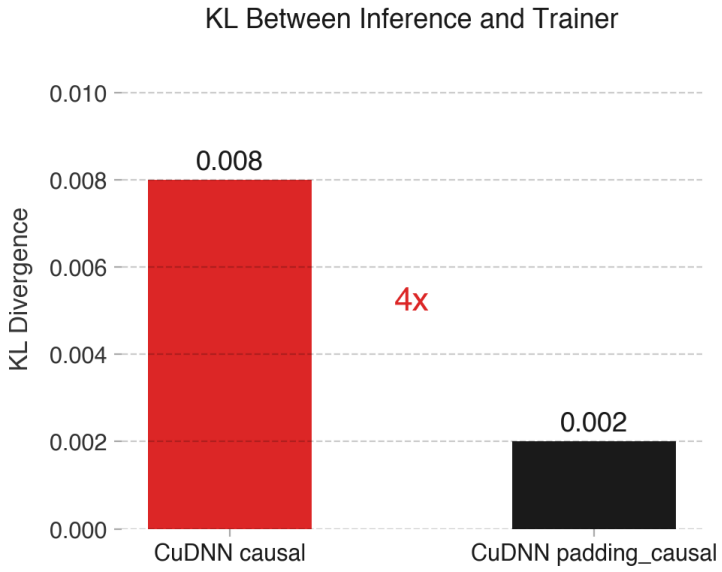
In-house implementation @ Periodic with much smaller overhead than vLLM.

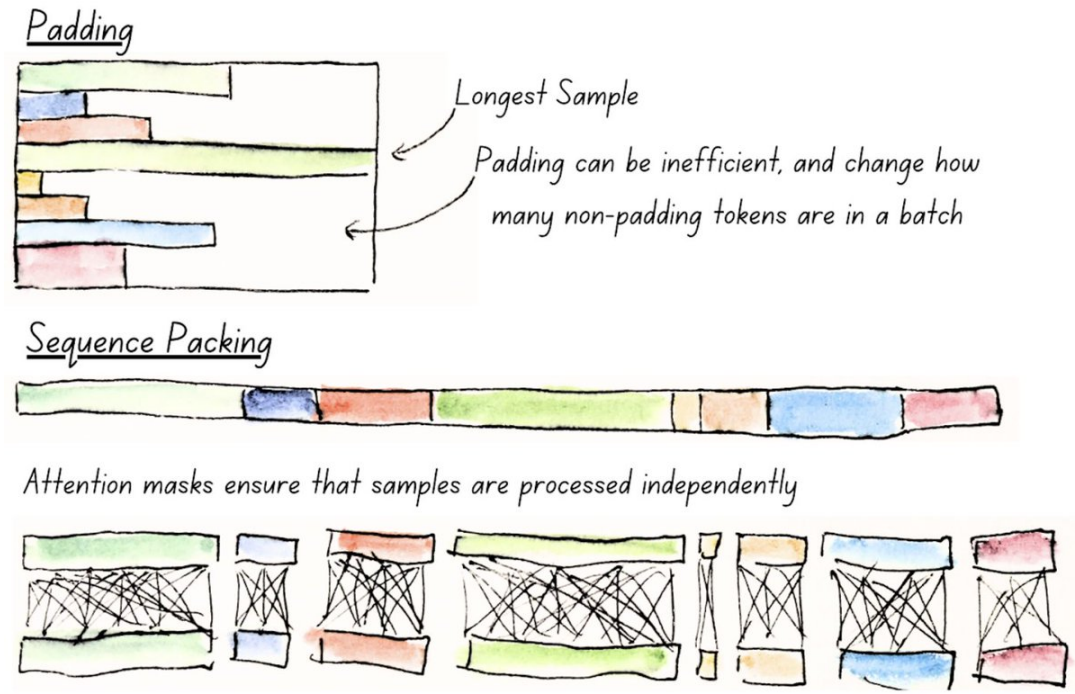
Figure modified from this paper

Frontier RL can run into kernel bugs!

causal and **padding_causal** (required for sequence packing) kernels produce different attention outputs

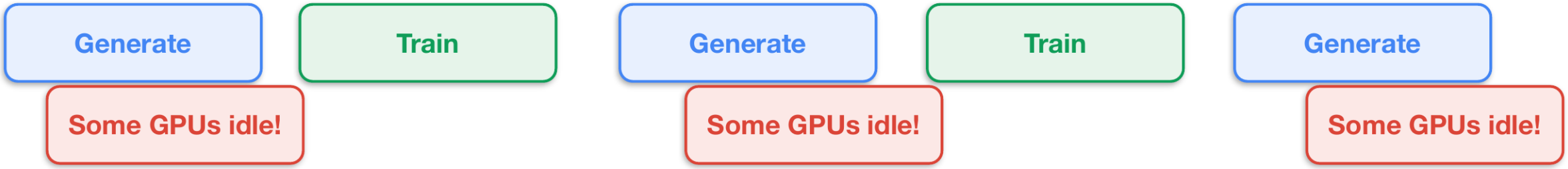
- Train / inference mismatch is **4x worse** in KL for Kimi-K2
- Nvidia is currently working on a CuDNN upstream fix!

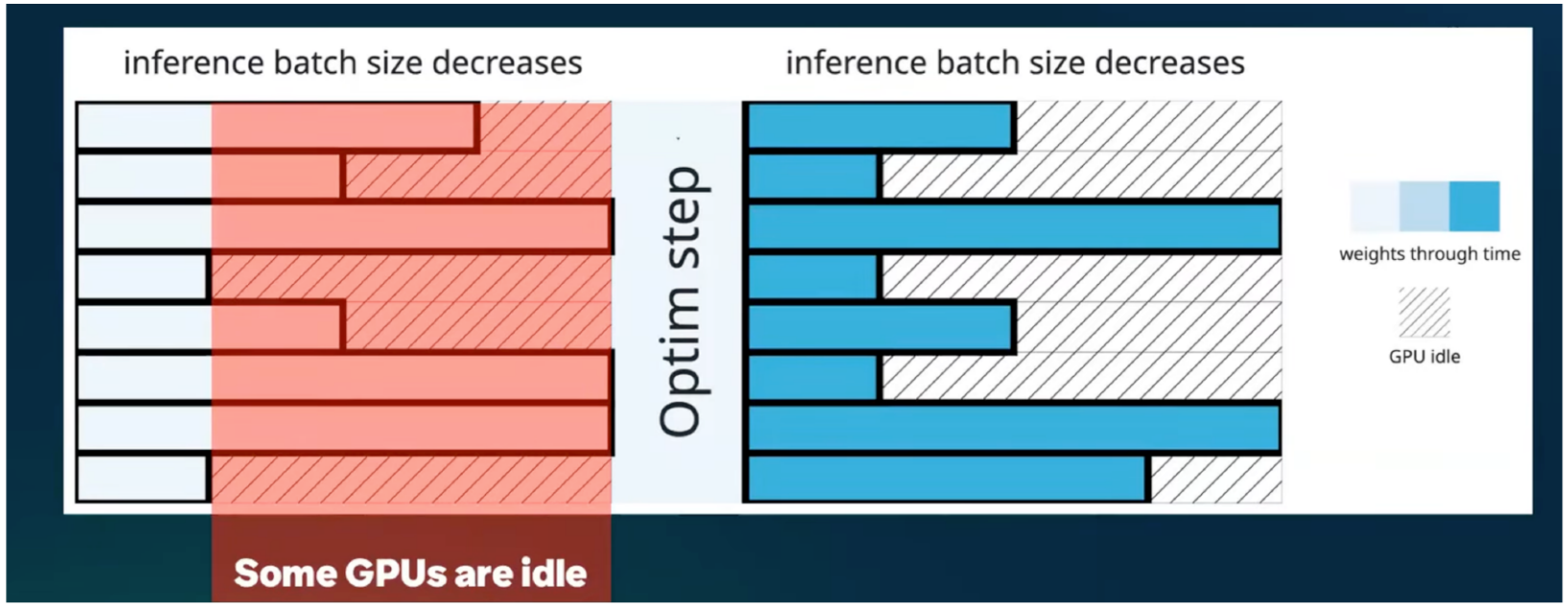




Frontier RL needs to be Asynchronous

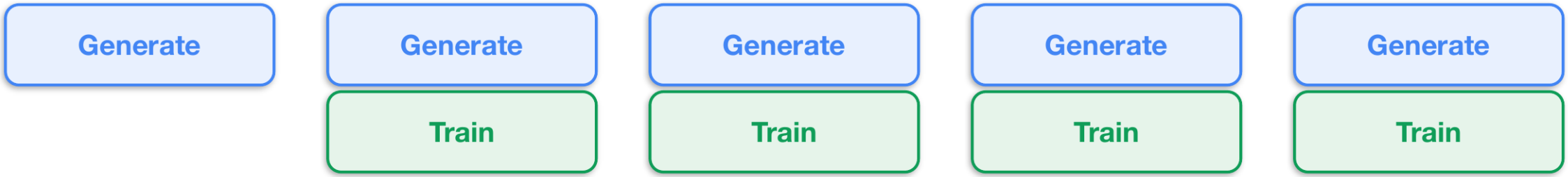
Synchronous On-Policy RL: GPUs sit idle significant % of the time



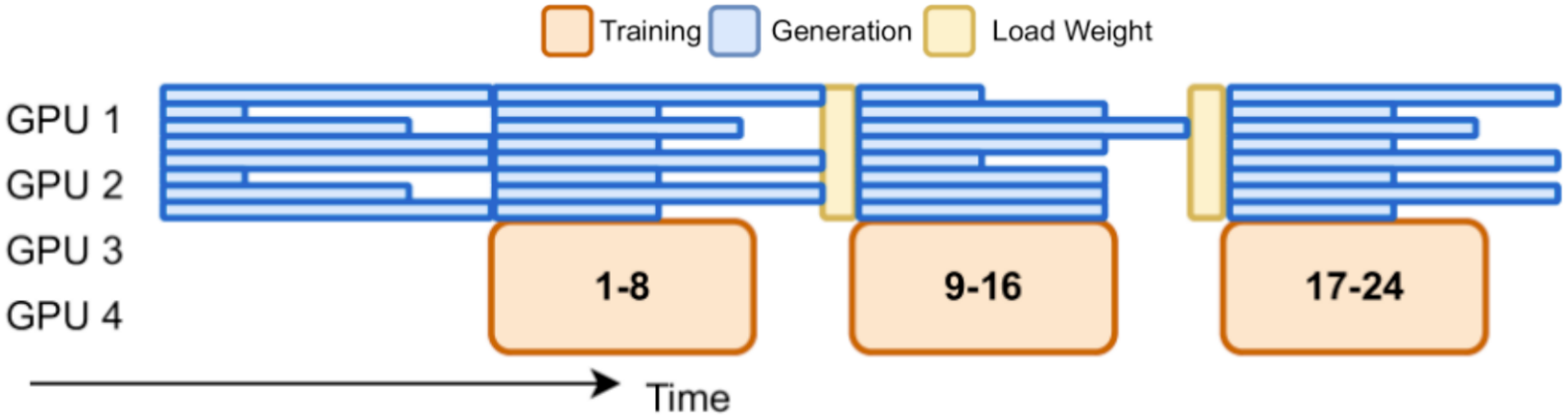


Frontier RL needs to be Asynchronous

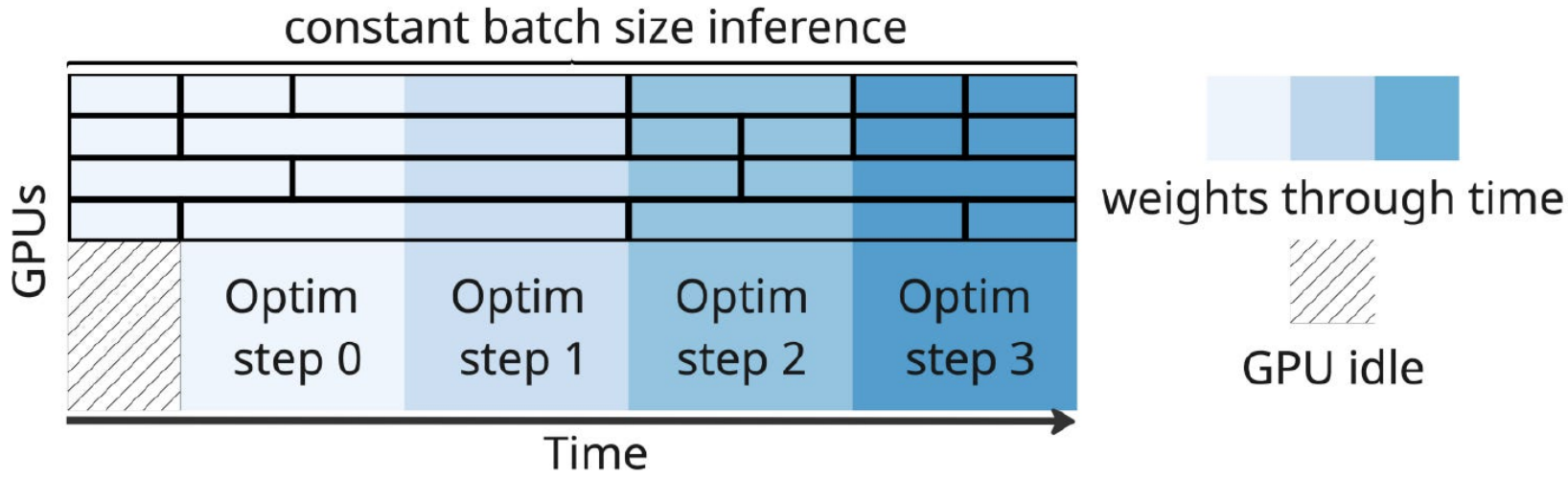
Async RL: generate & train simultaneously



ASYNCHRONOUS RLHF: FASTER AND MORE EFFICIENT OFF-POLICY RL FOR LANGUAGE MODELS



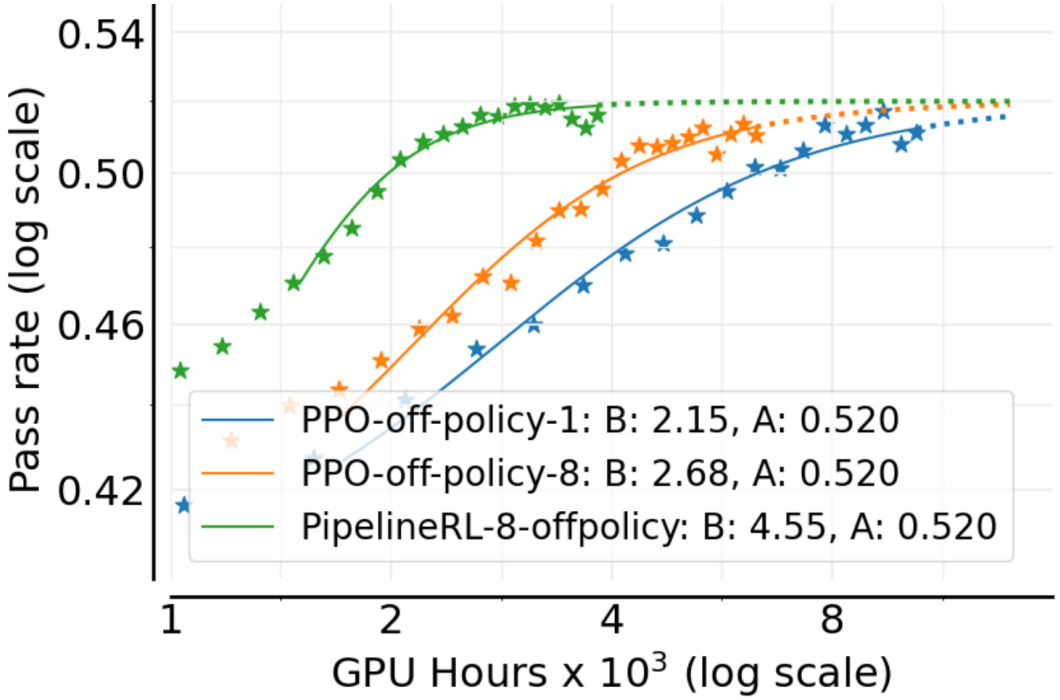
Pipeline RL for efficient Async RL



b) Pipeline RL with inflight weight updates.

PipelineRL runs generation and training concurrently, always using the freshest model weights for generations thanks to the in-flight weight updates.

Pipeline RL is our choice of Async RL

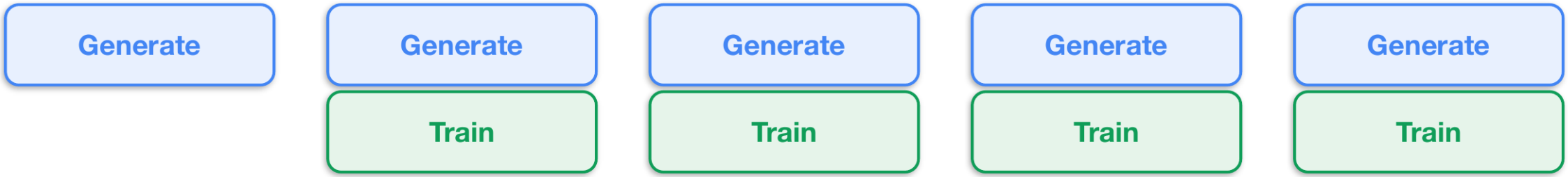


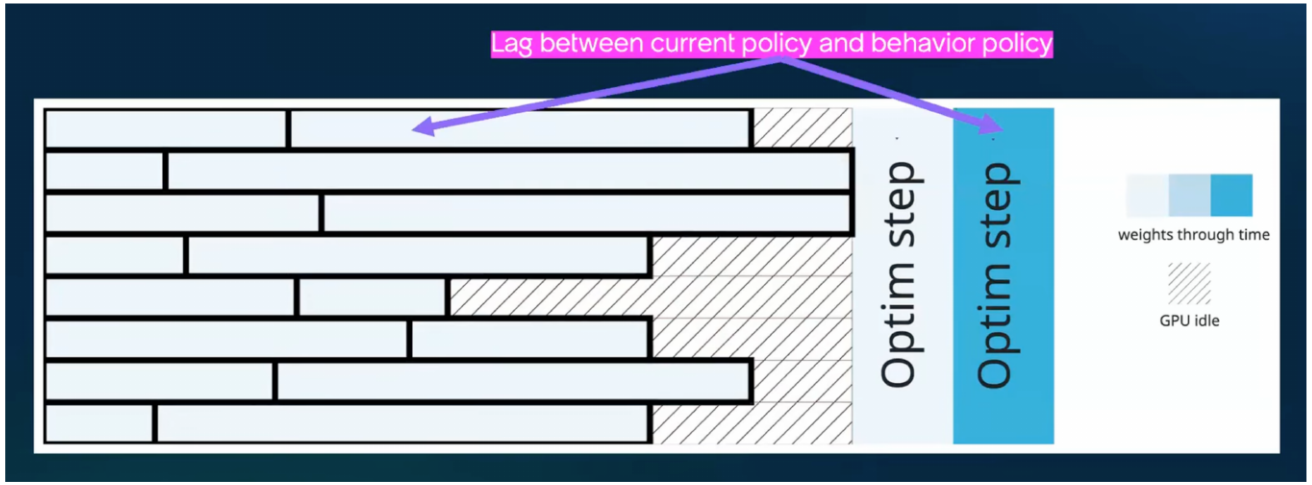
In-house Pipeline RL:
Enabling pause and resume during weight updates,
where inference requests are not quiesced or drained.

The Art of Scaling Reinforcement Learning Compute for LLMs, ICLR 2026 (Oral)

Asynchronous RL use “stale” rollouts

Async RL: generate & train simultaneously





Off-Policy Training:
The data generating policy is stale (old) compared to training policy.

Handling off-policy in RL: Importance Sampling

RL Objective

$$\overbrace{J(\pi_\theta)} = \mathbb{E}_{a\sim\pi_{\text{learner}}(\theta)}[R(a)] = \mathbb{E}_{a\sim\pi_{\text{sampler}}(\theta)}\left[\underbrace{\frac{\pi_{\text{learner}}(a,\theta)}{\pi_{\text{sampler}}(a,\theta)}}_{\text{importance ratio}} \cdot R(a)\right]$$

Policy Ratio
(different from reward r_t)

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$$

Current Policy

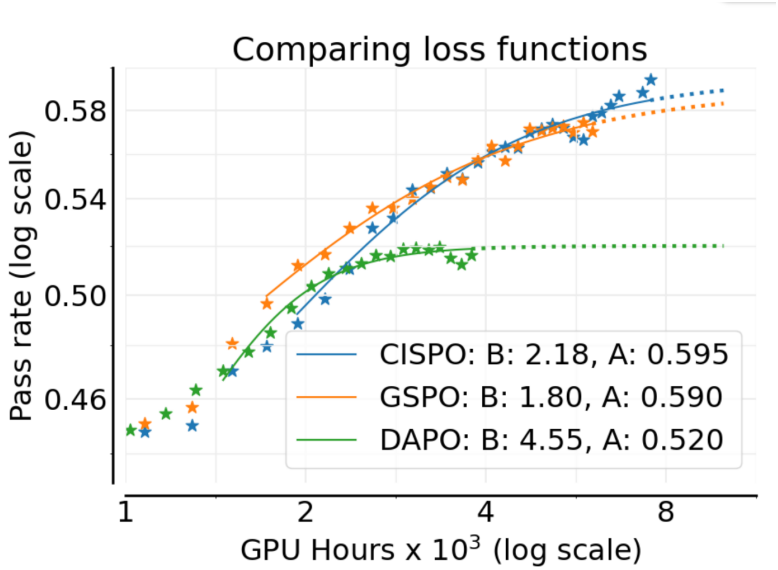
Old Policy (before any optimization steps / epochs)

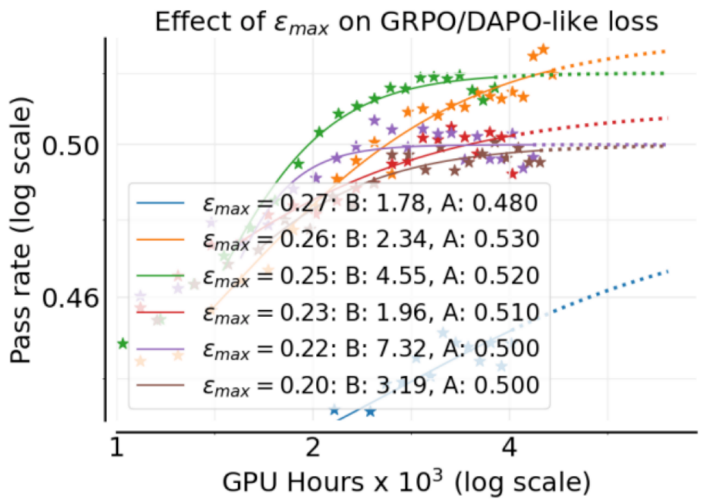
Sampling policy

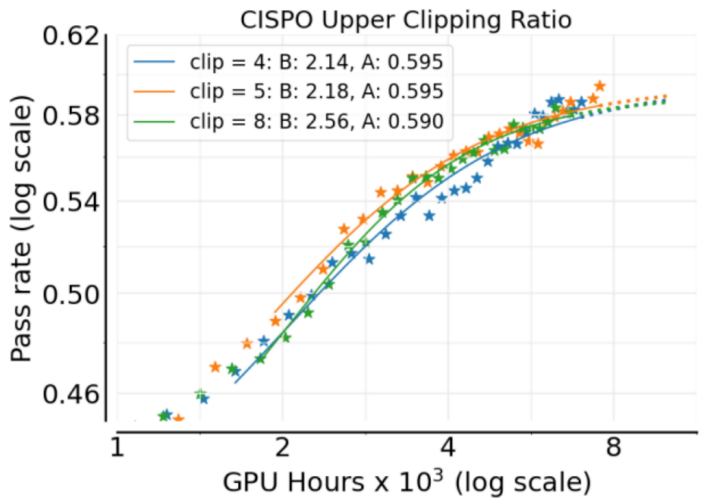
Policy Gradient

$$\boxed{\nabla_\theta J(\pi_\theta)} = \mathbb{E}_{a\sim\pi_{\text{sampler}}(\theta)}\left[\frac{\pi_{\text{learner}}(a,\theta)}{\pi_{\text{sampler}}(a,\theta)} \cdot R(a) \cdot \nabla_\theta \log \pi_{\text{learner}}(a,\theta)\right]$$

Not RL algorithms are created equal







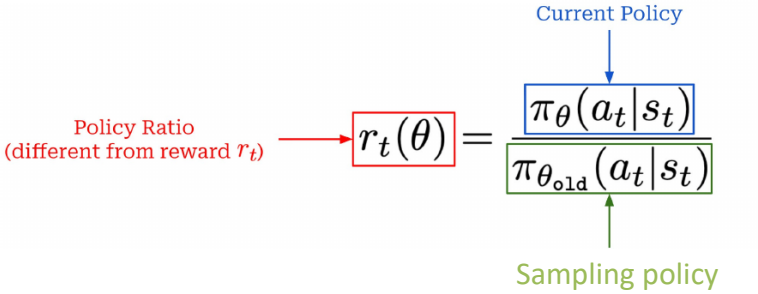
"Hard-to-tune" algorithms are not worth it
- Any new hyperparamter adds "one" more knob to tune
- Pick the simplest choice that works (e.g., drop KL regularization if it works!)

[The Art of Scaling Reinforcement Learning Compute for LLMs]

Masked-IS REINFORCE for MoE RL

$$\max_{\theta} \mathbb{E}_{y\sim\mu}\left[\sum_{t=1}^{|y|} M_t \cdot r_t(\theta) \cdot \hat{A}_t \nabla_\theta \log \pi_{\text{learner}}(a_t \mid \theta)\right]$$

Mask tokens with large train-inference mismatch, usually <0.02% tokens or lower!



[See IcePop for example.]

Masked-IS REINFORCE for MoE RL

$$\max_{\theta} \mathbb{E}_{y\sim\mu}\left[\sum_{t=1}^{|y|} M_t \cdot r_t(\theta) \cdot \hat{A}_t \nabla_\theta \log \pi_{\text{learner}}(a_t \mid \theta)\right]$$

Mask tokens with large train-inference discrepancy, usually <0.02% tokens or lower!

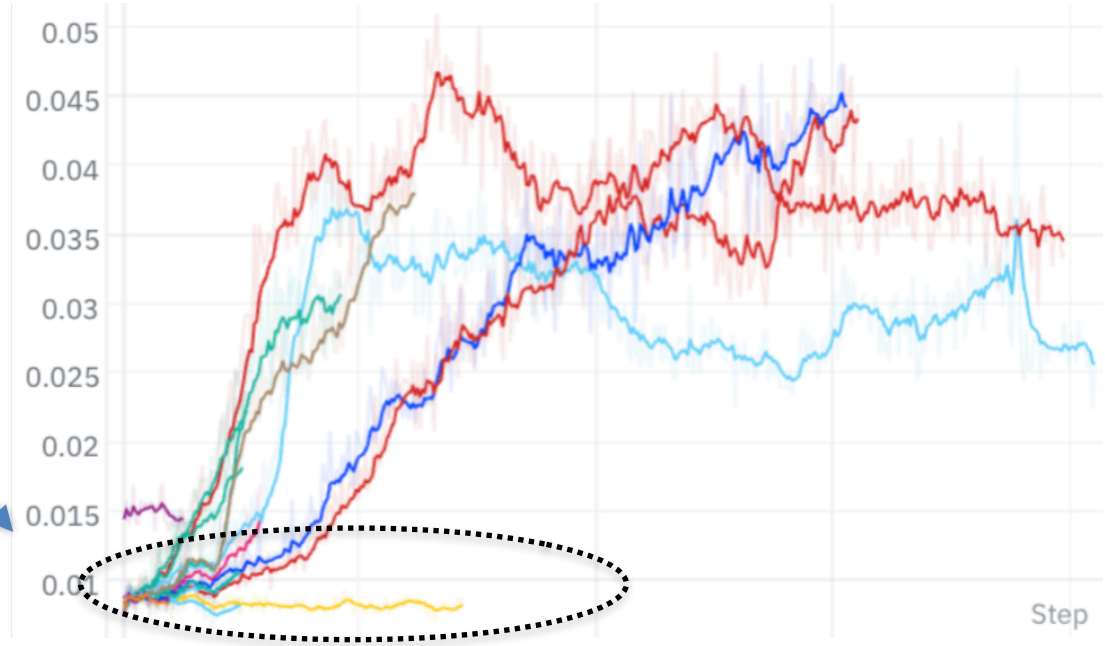
Policy Ratio (different from reward r_t)

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$$

Current Policy

Sampling policy

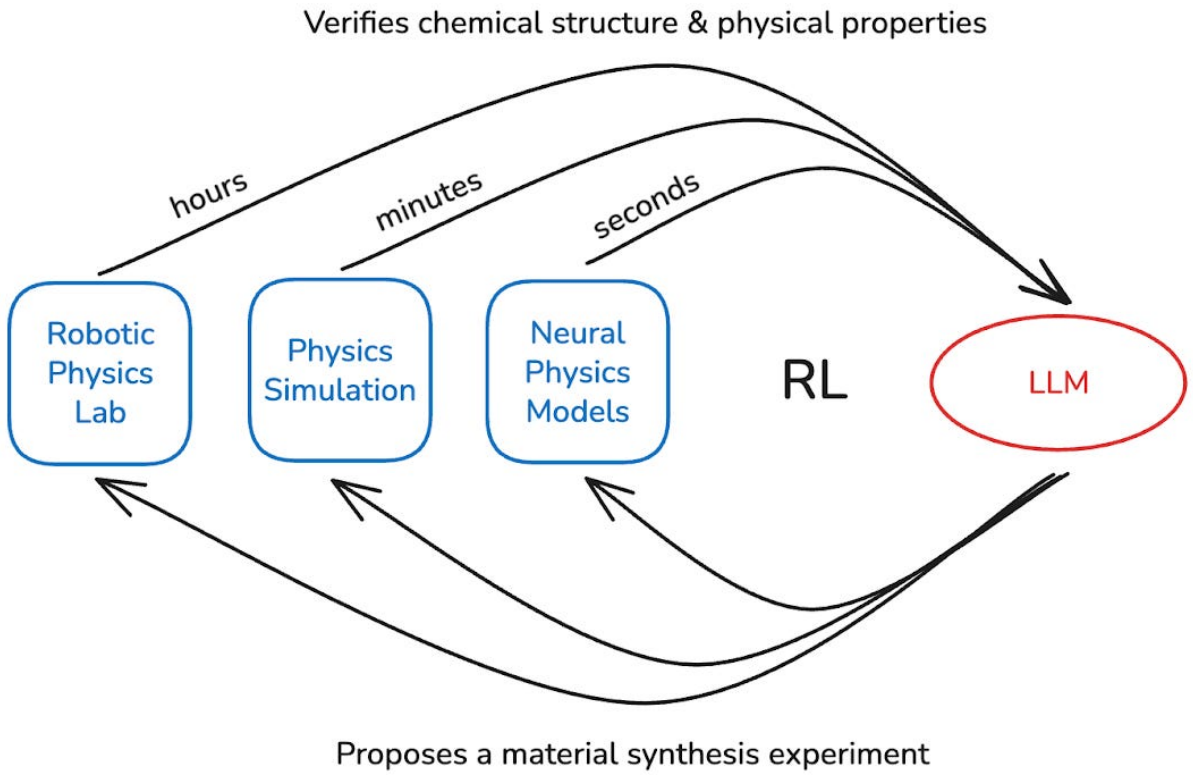
Kimi-K2: KL divergence between inference and trainer



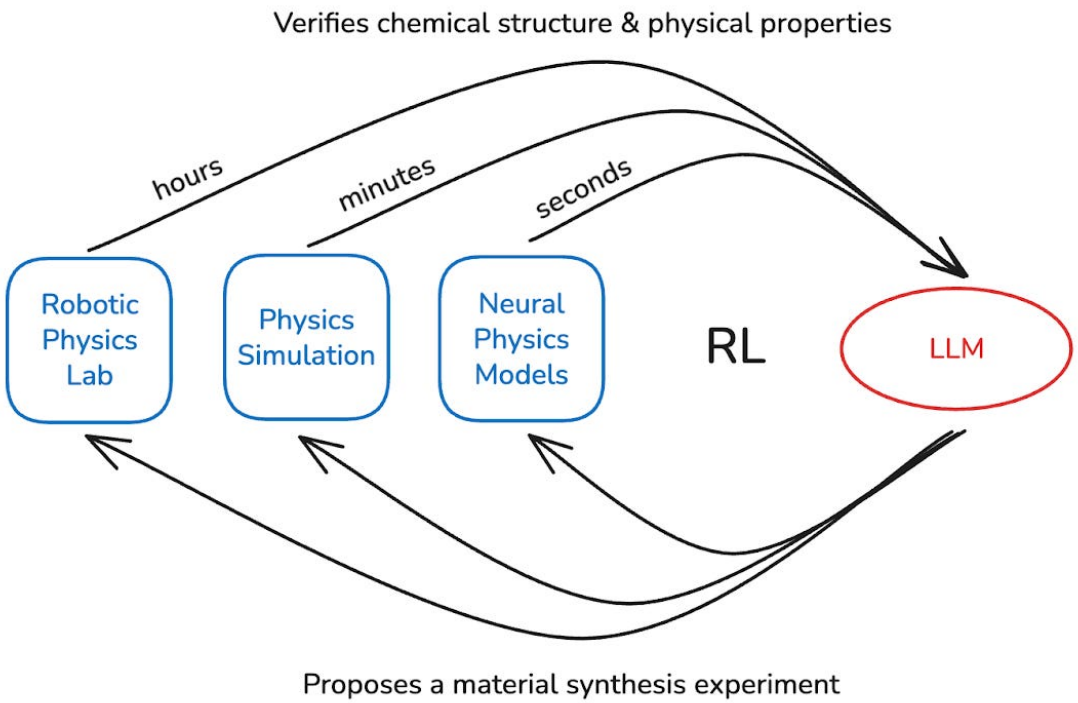
[See IcePop for example.]

What's next? RL for scientific discovery

By focusing directly on advancing science, different sets of ML infra and research questions rise to prominence



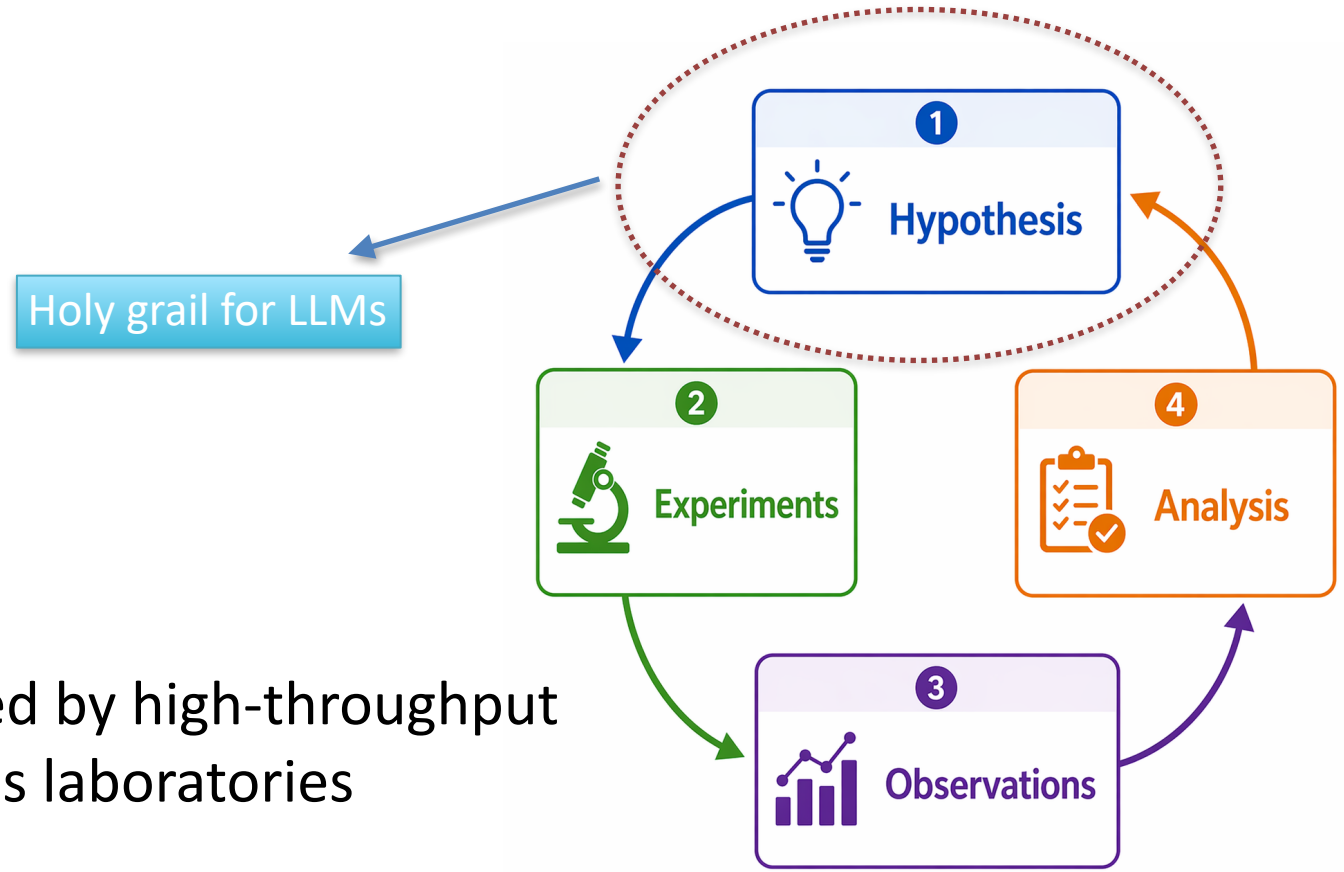
What's next? RL for scientific discovery



As an example, simulations and physical laboratories as reward functions are high-latency! e.g., compare an RLHF RM vs. results from an experiment

Producing async RL stacks with high resilience to very stale, off-policy samples key for achieving sufficient throughput

What's next? RL for scientific discovery



Bottlenecked by "intelligence"

Bottlenecked by high-throughput autonomous laboratories

Thank you!
Questions?