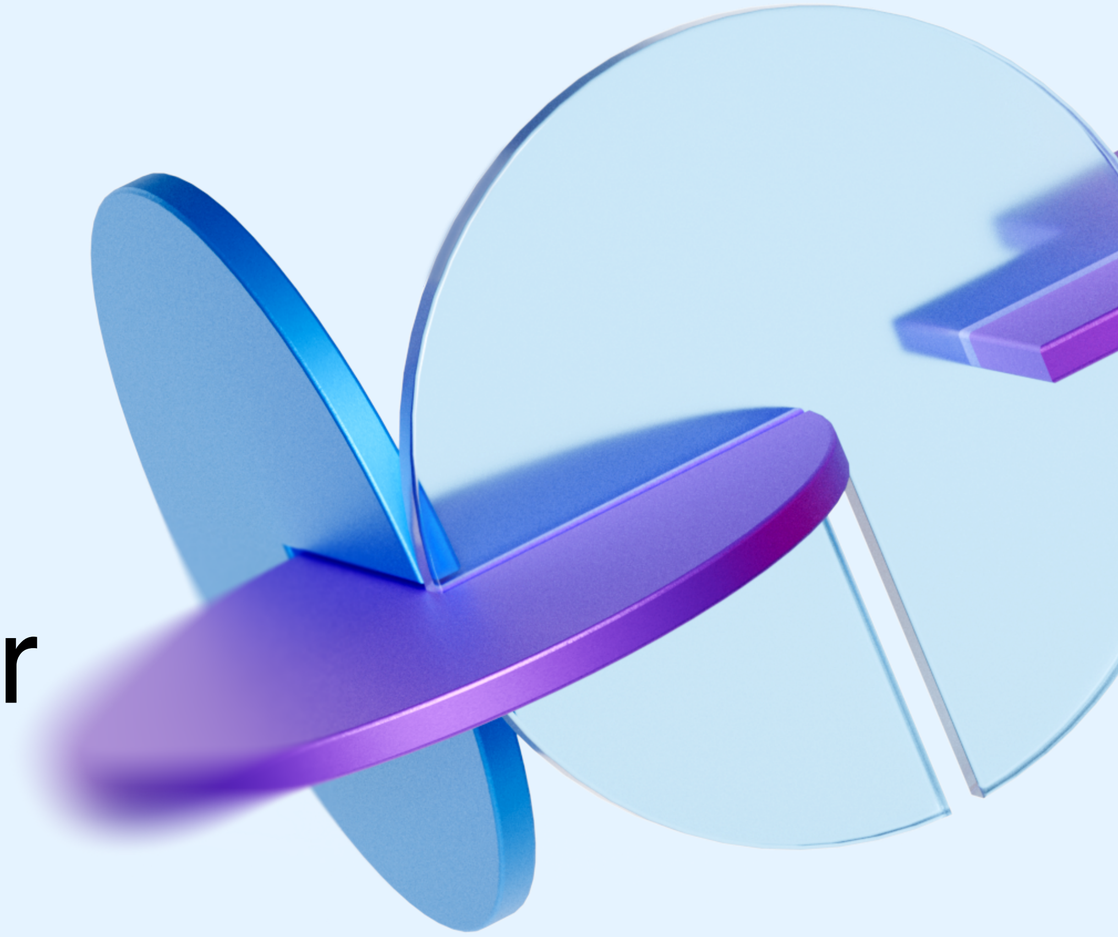
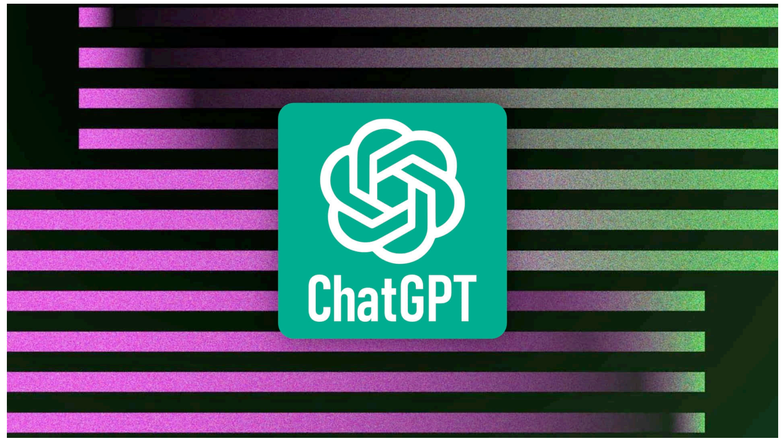
together.ai

Beat the long tail: Distribution-Aware Speculative Decoding for RL Training



Zelei Shao*, **Vikranth Srivatsa*,** Sanjana Srivastava, Qingyang Wu, Alpay Ariyak, Xiaoxia Wu, Ameen Patel, Jue Wang, Percy Liang, Tri Dao, Ce Zhang, Yiying Zhang, Ben Athiwaratkun, Chenfeng Xu, Junxiong Wang

RL responsible core breakthroughs



RL powers the next advancement in fine tuning

RL enables model specialization

OSWorld

Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments

COMPUTERRL: SCALING END-TO-END ONLINE REINFORCEMENT LEARNING FOR COMPUTER USE AGENTS

Hanyu Lai[1*†], Xiao Liu[1,2*], Yanxiao Zhao[3*†], Han Xu[2], Hanchen Zhang[1†], Bohao Jing[2†], Yanyu Ren[1†], Shuntian Yao[1†], Yuxiao Dong[1], Jie Tang[1]

[1] Tsinghua University [2] Zhipu AI [3] University of Chinese Academy of Sciences

Rank #1, Aug 17 2025, 47%

KernelBench: Can LLMs Write GPU Kernels?

Kevin-32B: Multi-Turn RL for Writing CUDA Kernels

SOTA, July 16 2025

terminal-bench: a benchmark for ai agents in terminal environments

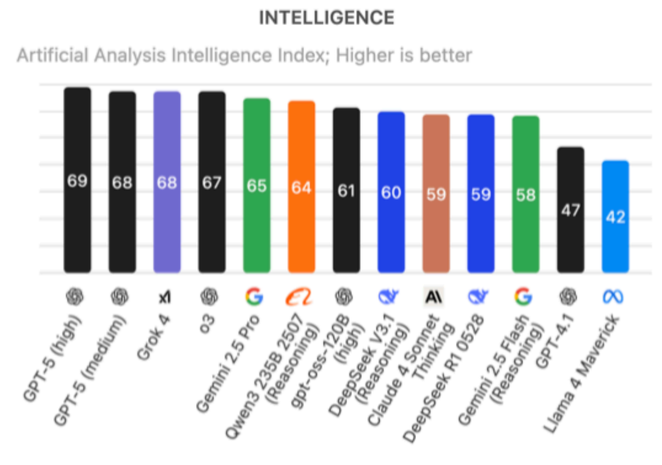
Claude Sonnet 4

Rank #3, Aug 24, 2025

What makes RL promising now?

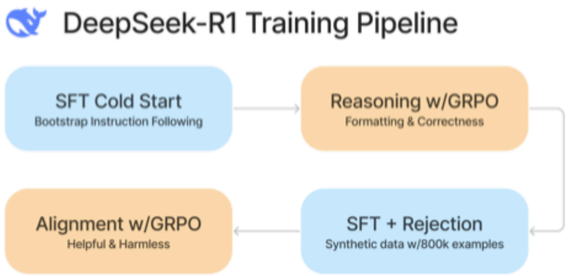
Rise of **Reasoning** Models

Together DeepSWE: "RL is more effective on reasoning models"



Deepseek release of **GRPO** based RL

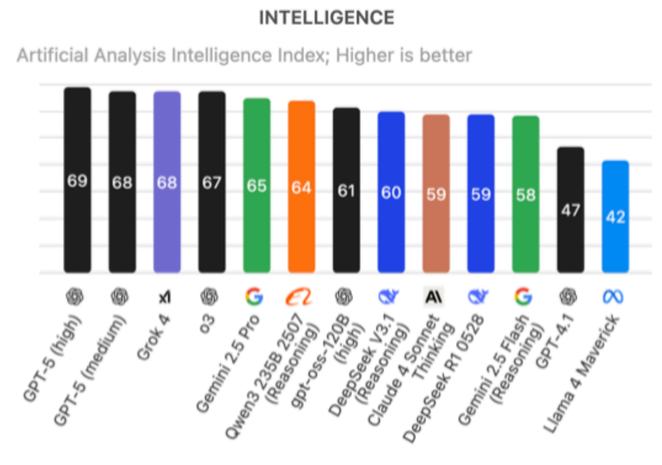
Makes RL Training Significantly easier



What makes RL promising now?

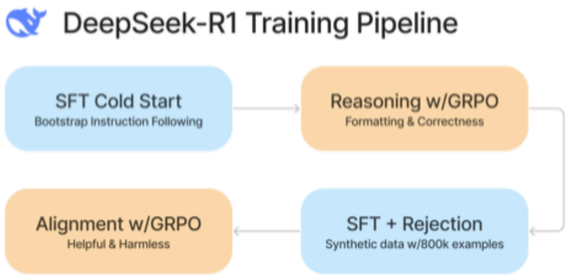
Rise of **Reasoning** Models

Together DeepSWE: "RL is more effective on reasoning models"



Deepseek release of **GRPO** based RL

Makes RL Training Significantly easier



Training RL is Expensive

gpt-oss-120b

2.1 **million** H100-hours

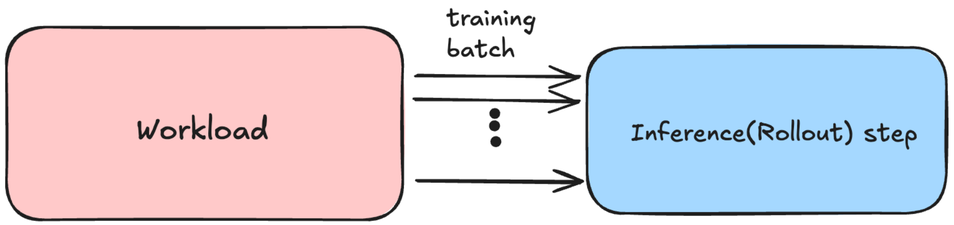
Deepseek v3

2.8 **million** H800 Hours

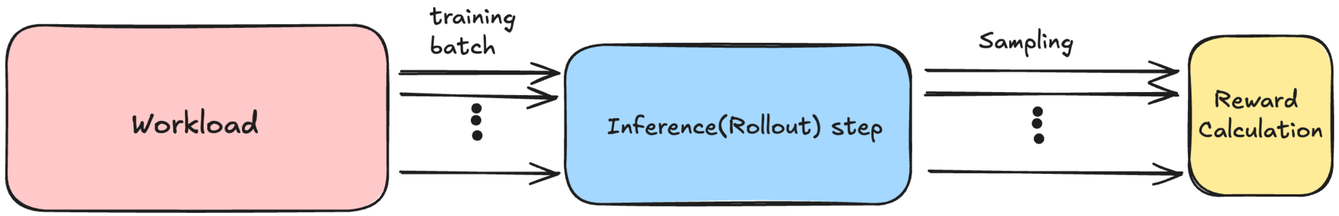
Scaling laws and specialization per dataset
Increases the training cost

DAS(Distribution Aware Spec) takes advantage of temporal repetition to speed up RL by up to 50%

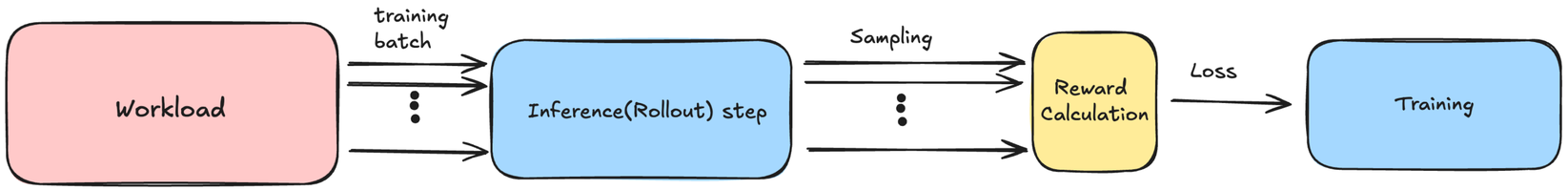
Basics of RL Inference



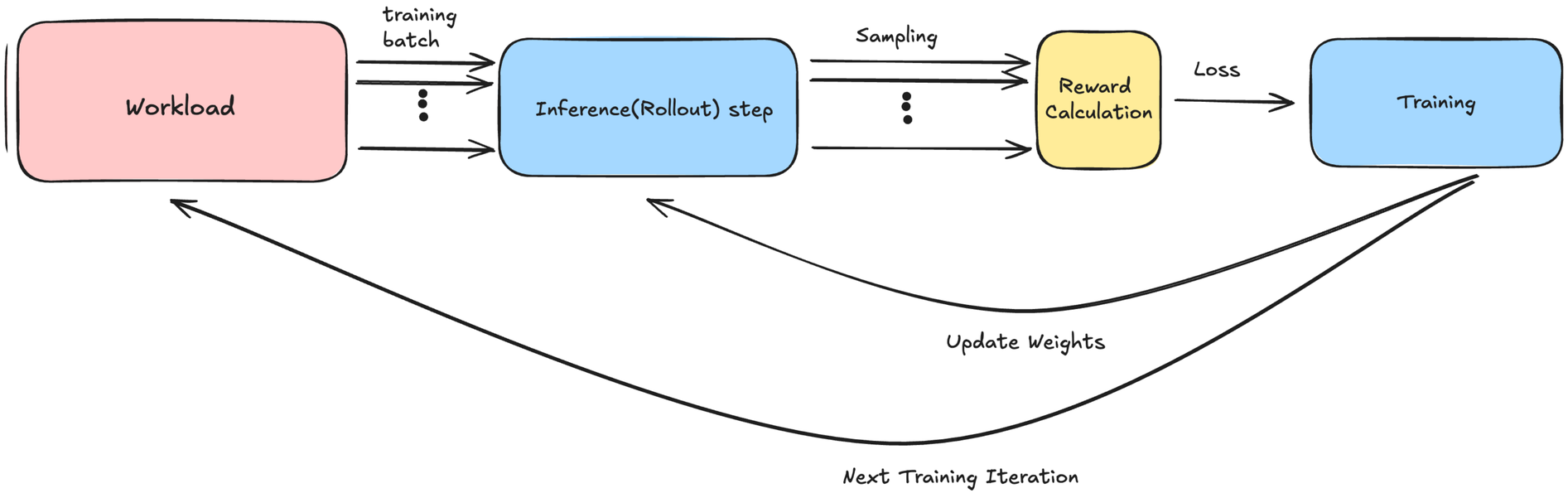
How does RL inference work today?



How does RL inference work today?

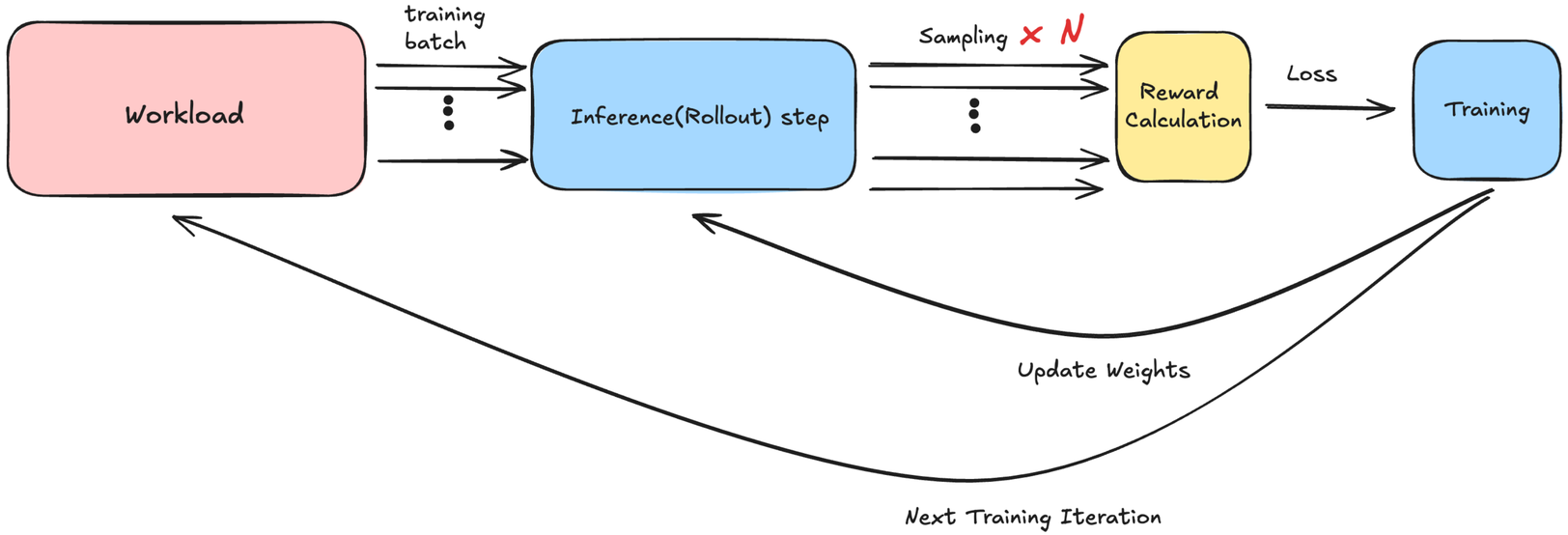


How does RL inference work today?



How does GRPO RL inference work today?

Deepseek's GRPO(Group Relative Policy Optimization)

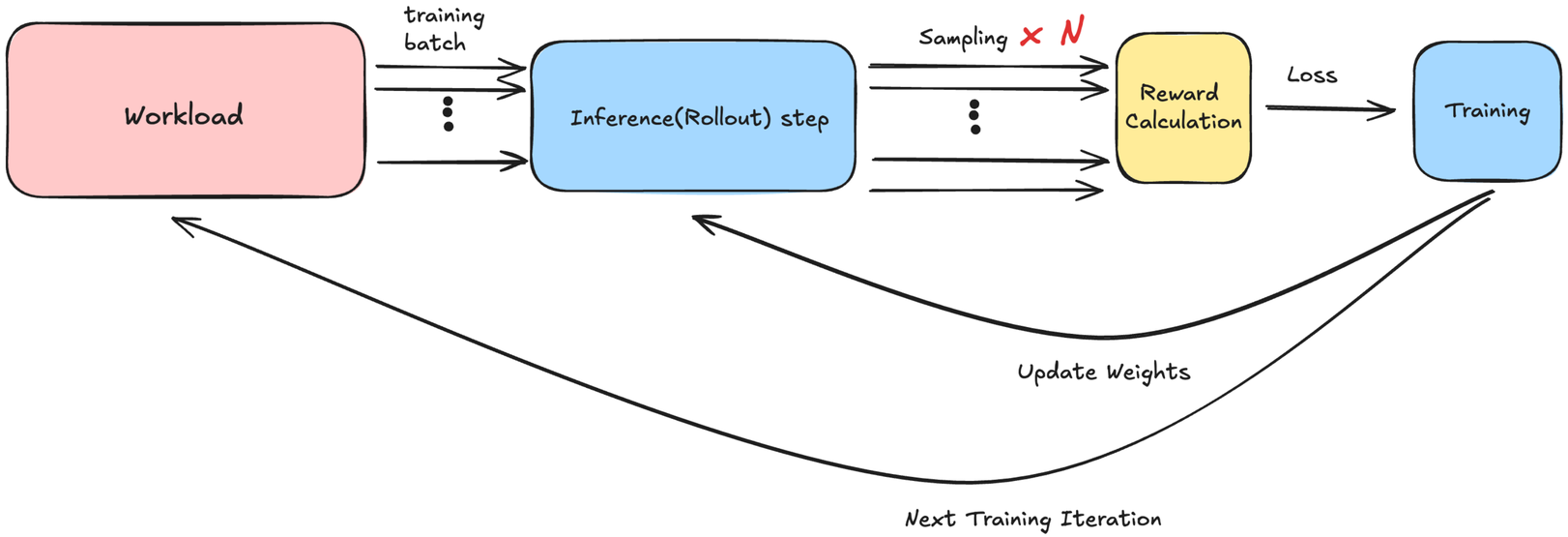


GRPO increases the cost of inference

decreases the cost of training

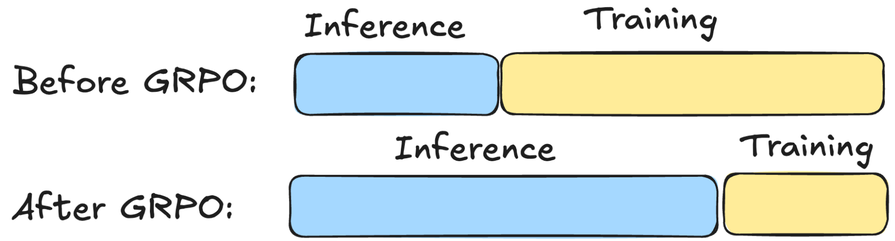
How does GRPO RL inference work today?

Deepseek's GRPO(Group Relative Policy Optimization)

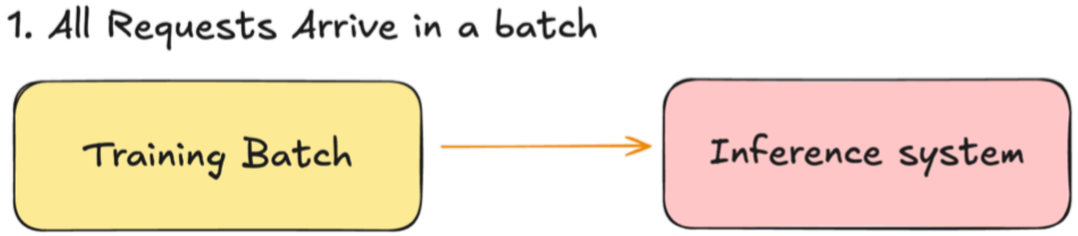


GRPO increases the cost of inference

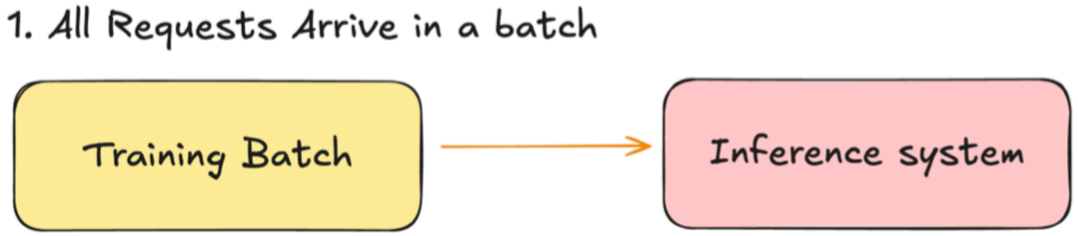
decreases the cost of training



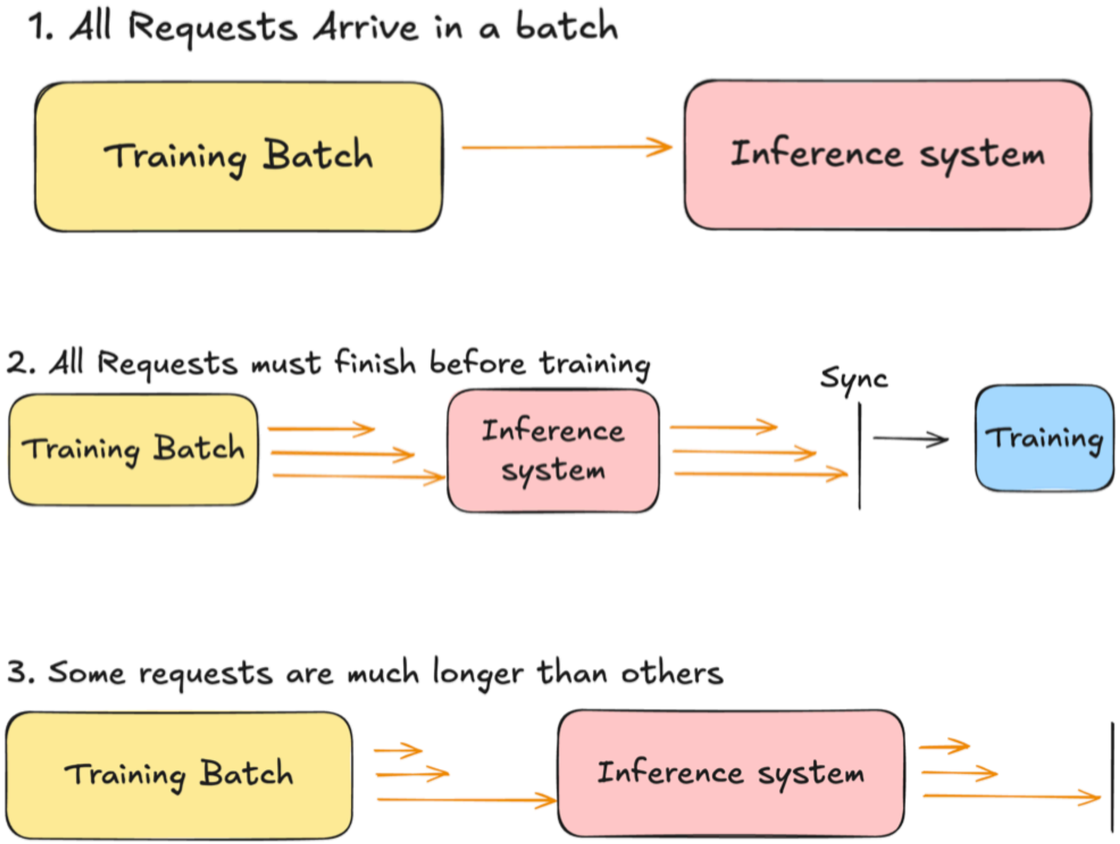
What are some properties to build RL Systems?



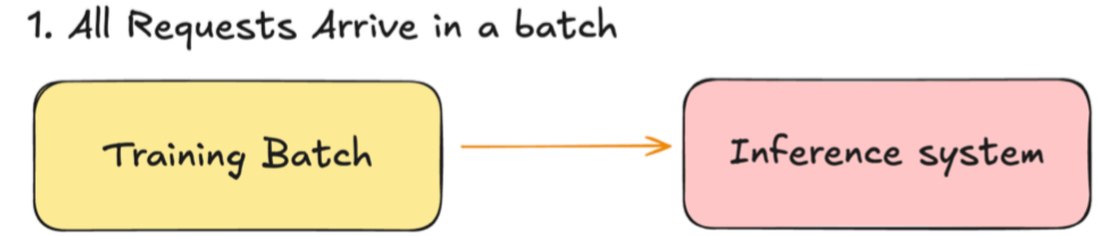
What are some properties to build RL Systems?

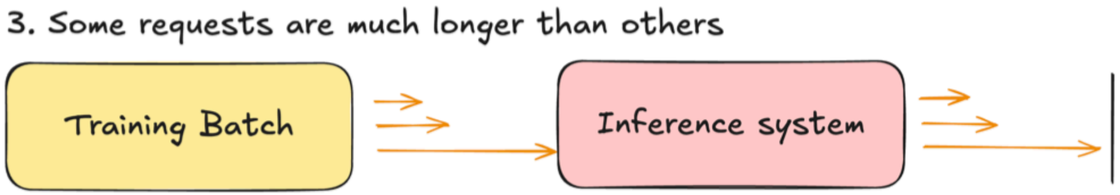


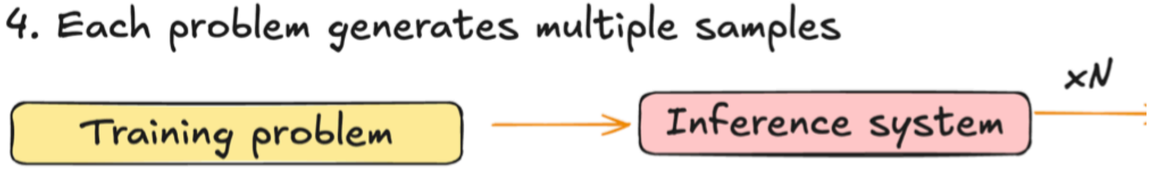
What are some properties to build RL Systems?



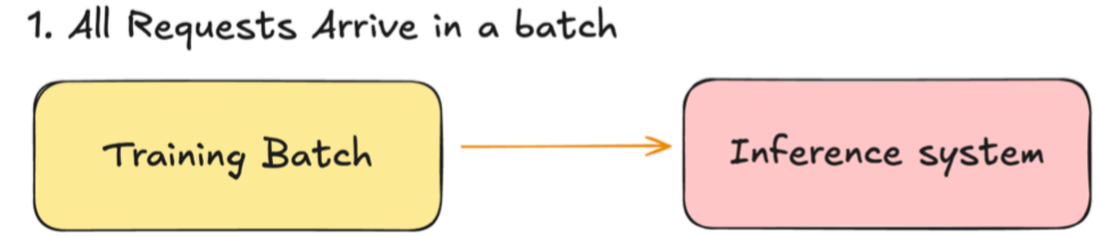
What are some properties to build RL Systems?

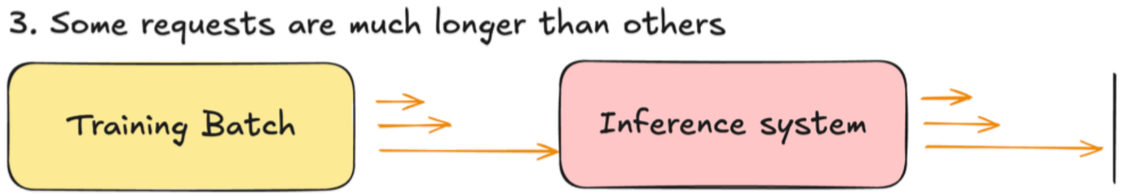


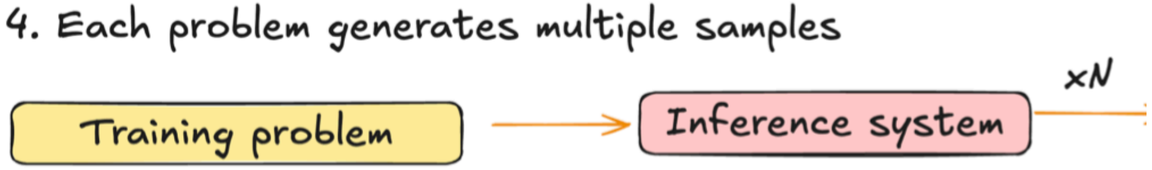


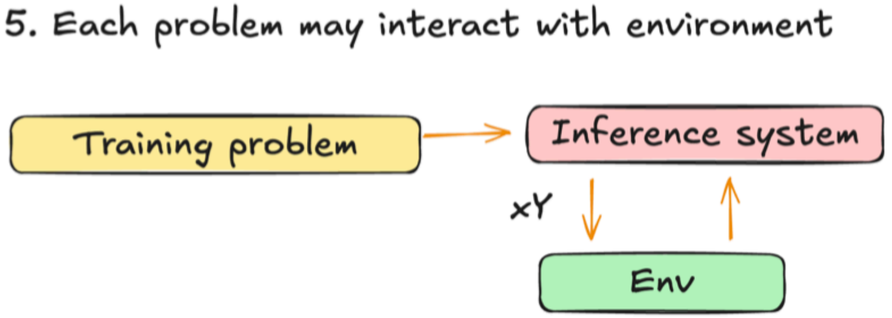


What are some properties to build RL Systems?

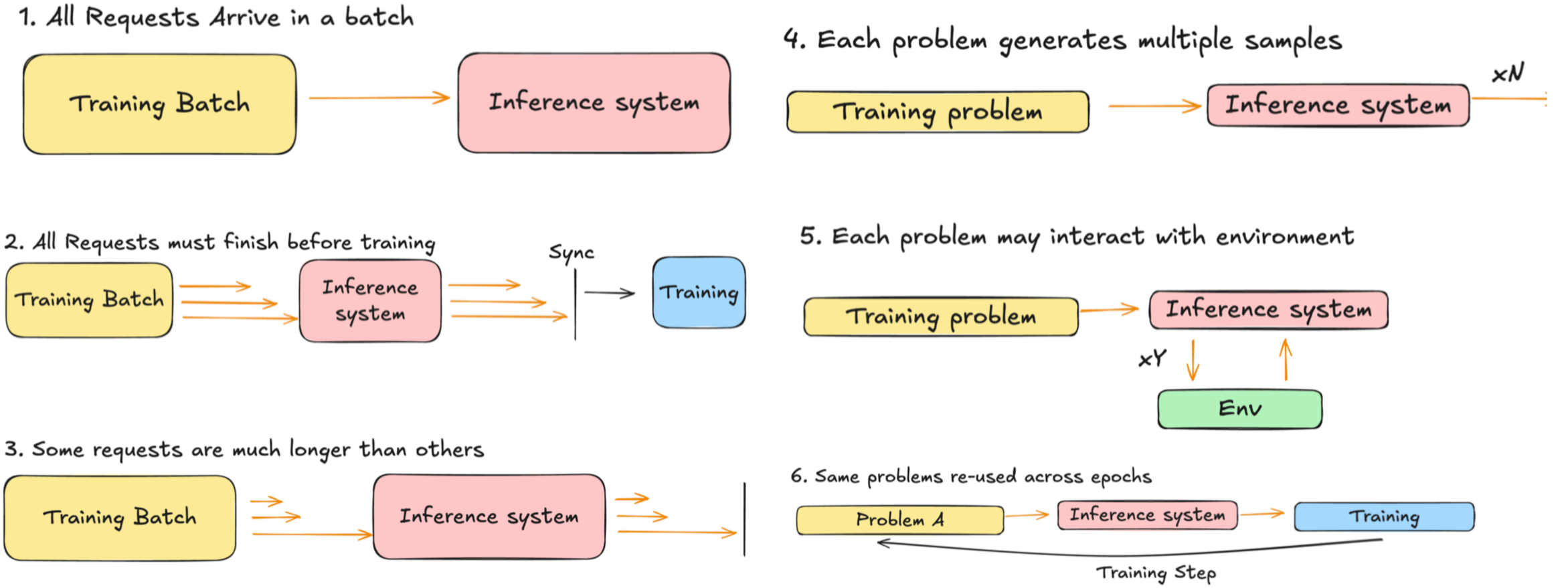




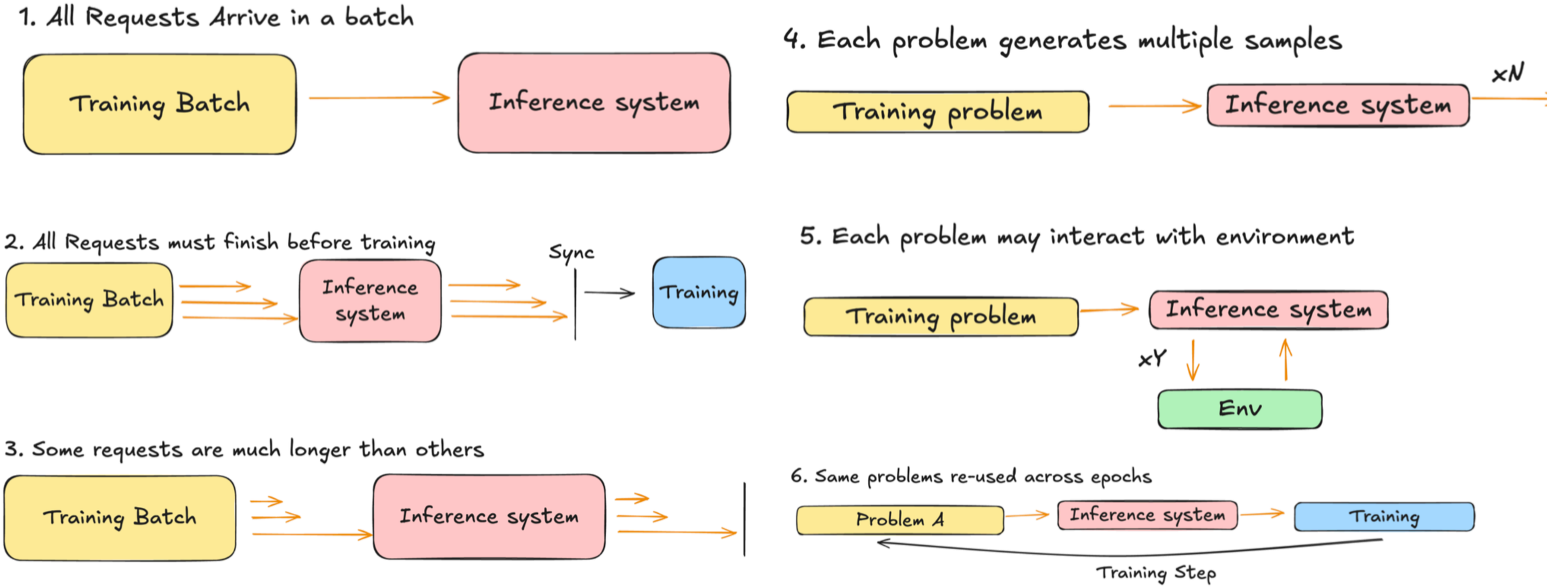




What are some properties to build RL Systems?



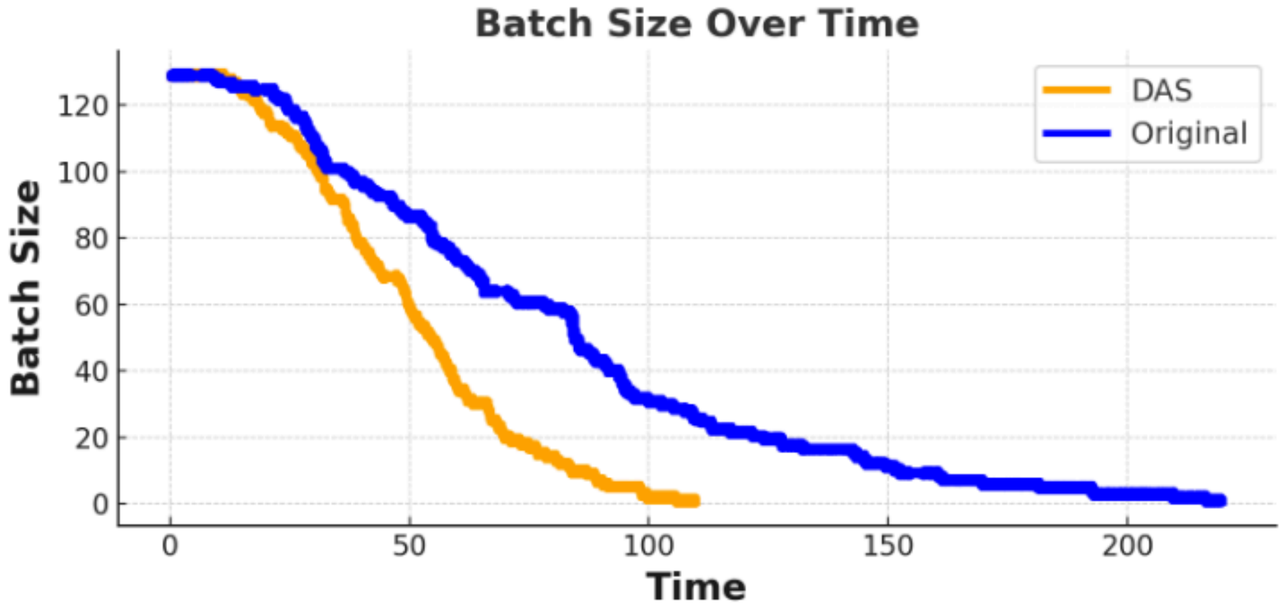
What are some properties to build RL Systems?



Study on Long Tail Rollouts

Straggler problem

Problem Output Length Variance

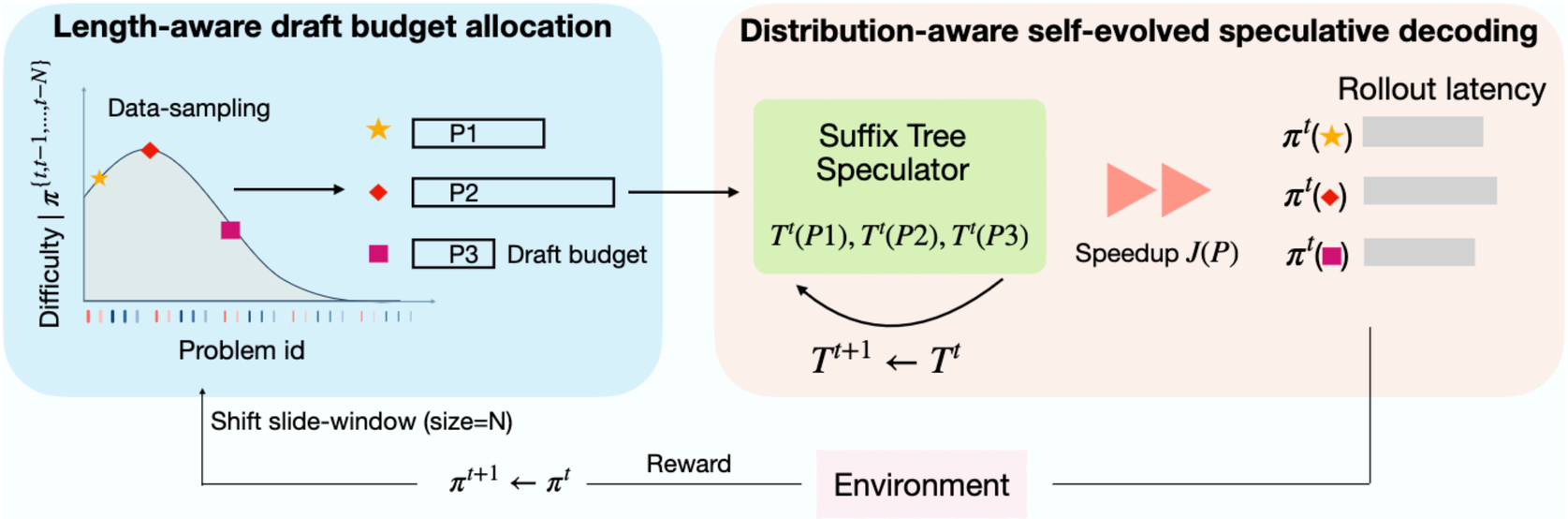


hard problems block the RL execution step

Key Insight

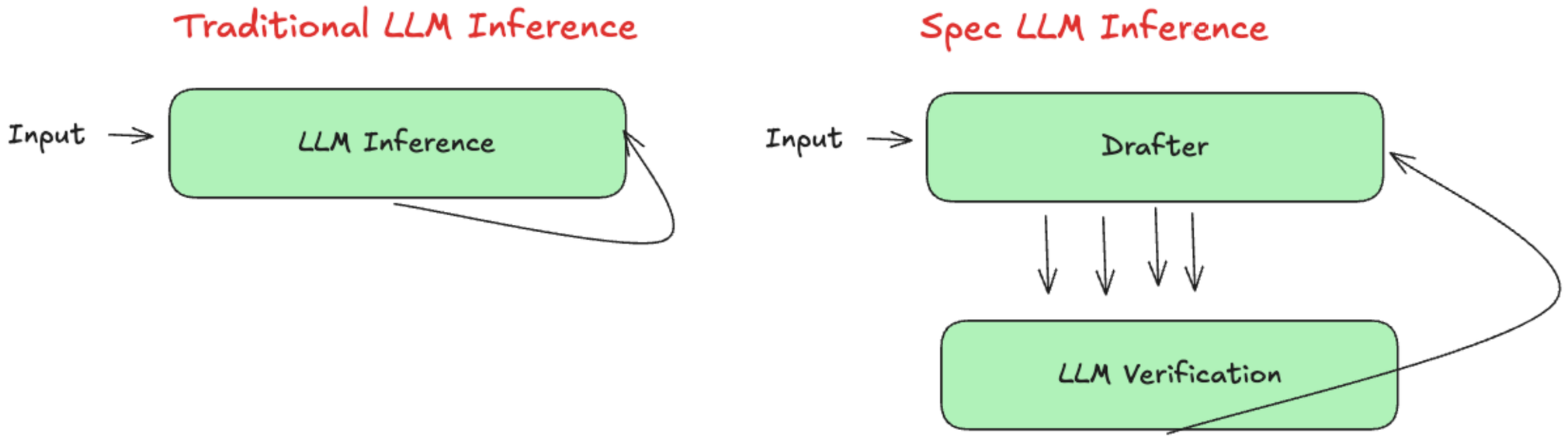
1. GRPO samples **similar problems** across each mini batch
2. Same problems **repeated through** through **training iterations**
3. RL Training **updates very slow** due to convergence
4. Problem variance creates **stragglers** in training

Overview of DAS

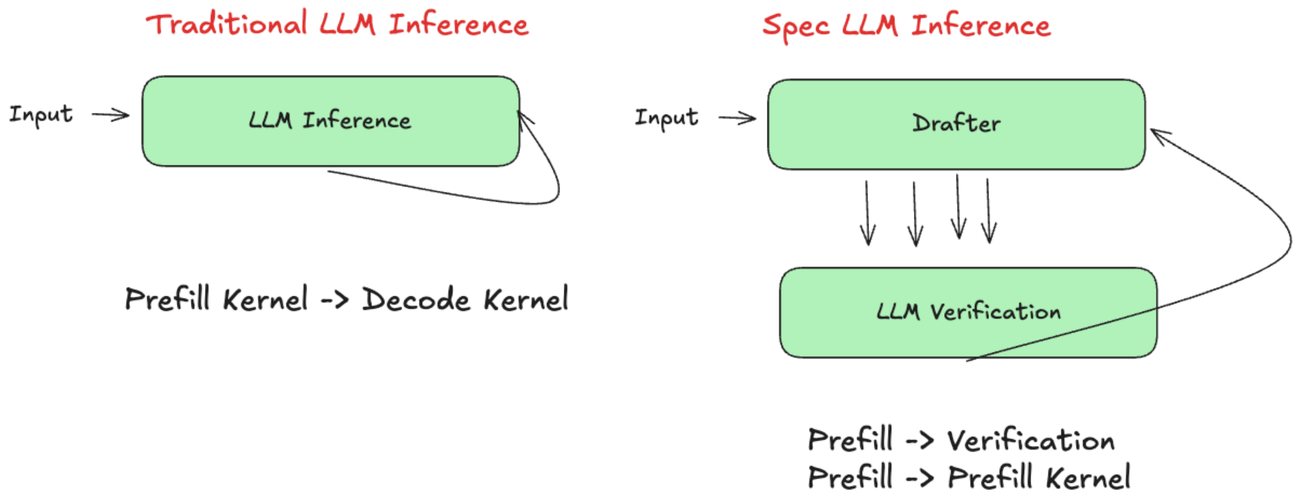


1. Distribution aware spec decoding
2. Length Aware Scheduling
3. Windowed Suffix Tree

Suffix Decoding Explained



Suffix Decoding Changes Bottleneck

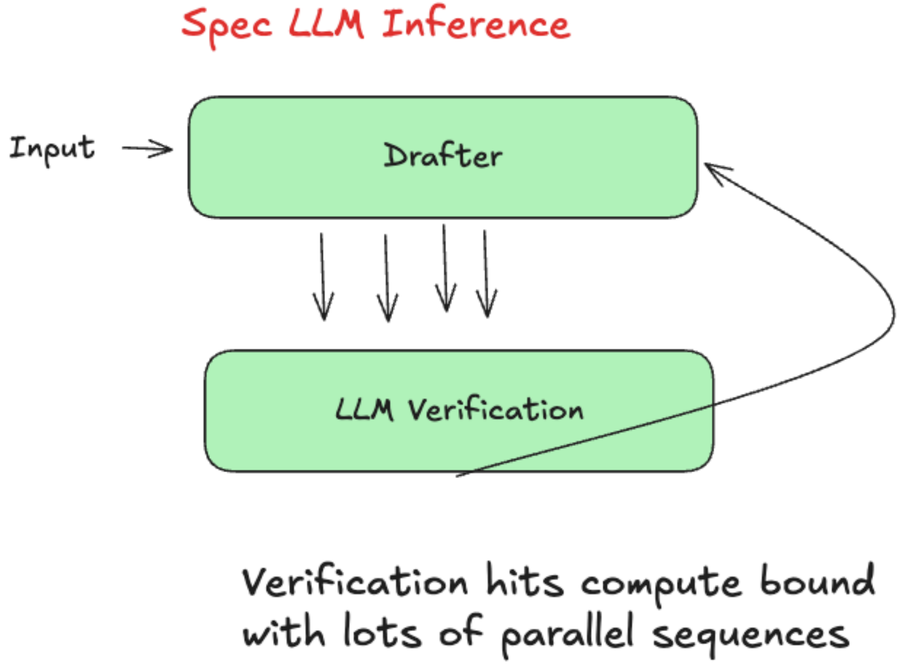


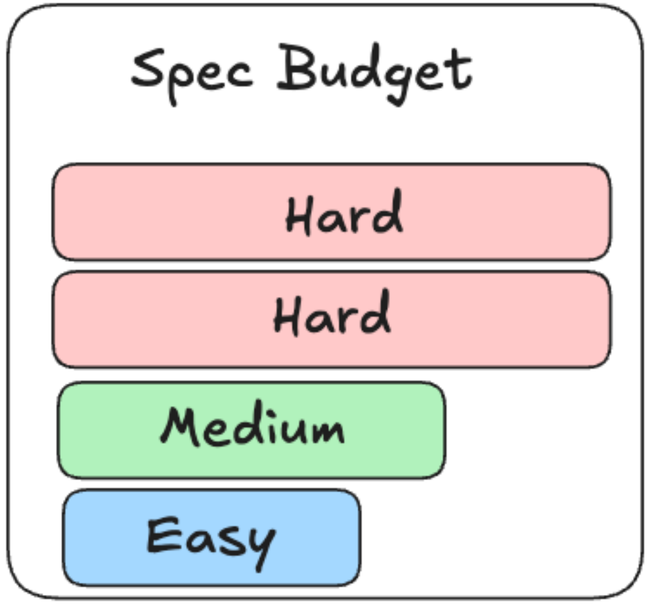
Normal Decoding -> **Memory bound**

Spec Decoding (Verification)-> can be **Compute bound**

Key Insight: Tradeoff between # verified sequences and proposed sequences

Distribution Aware Spec Decoding

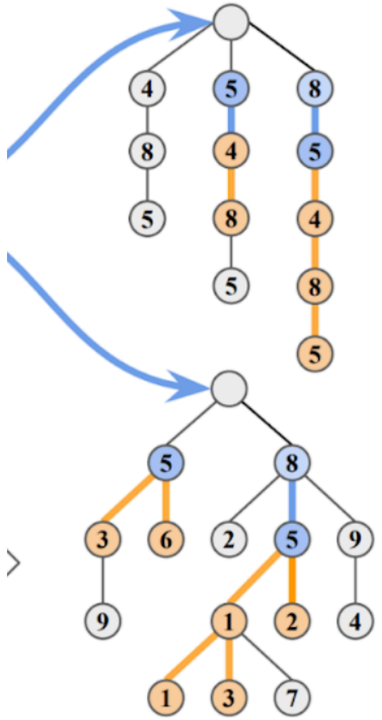




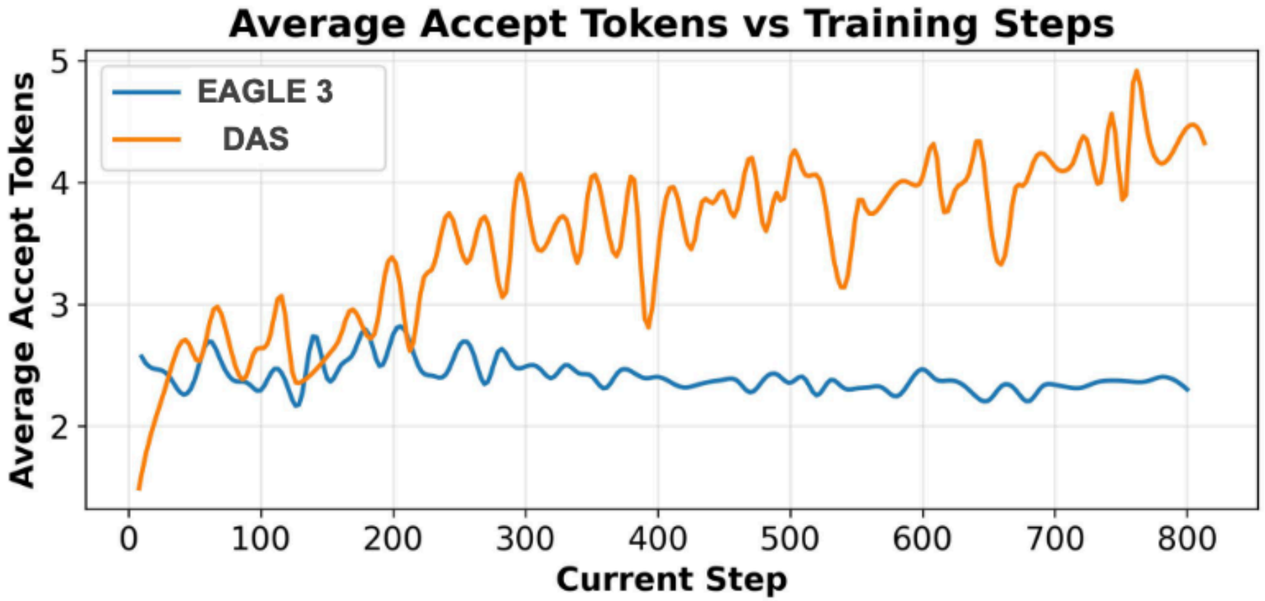
Key Insight: Propose problems proportional to problem difficulty

Distribution Aware Suffix Decoding

We propose using suffix instead of neural models



Suffix Trees



Spec decoding enables continuously updating as **training adapts**

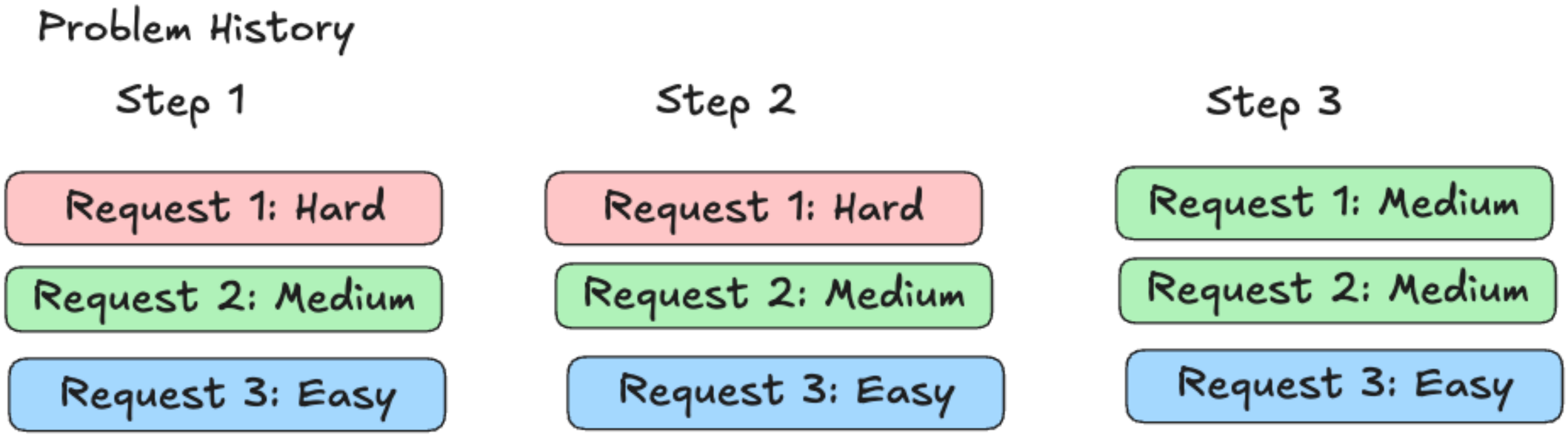
+ **Preserves Accuracy**

How do you schedule based on problem difficulty?

Understanding Problem Difficulty

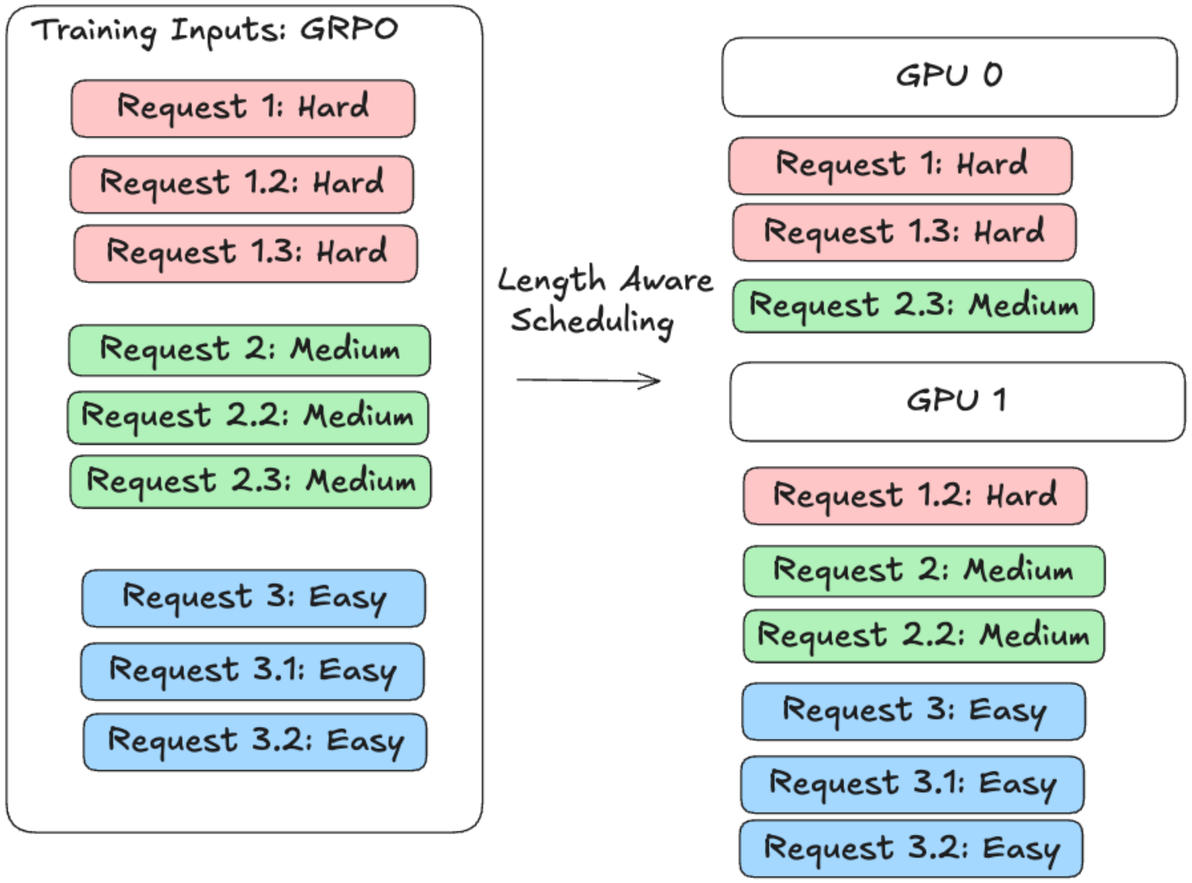
RL repeatedly trains the same inputs

Suffix spec decoding learns from the history



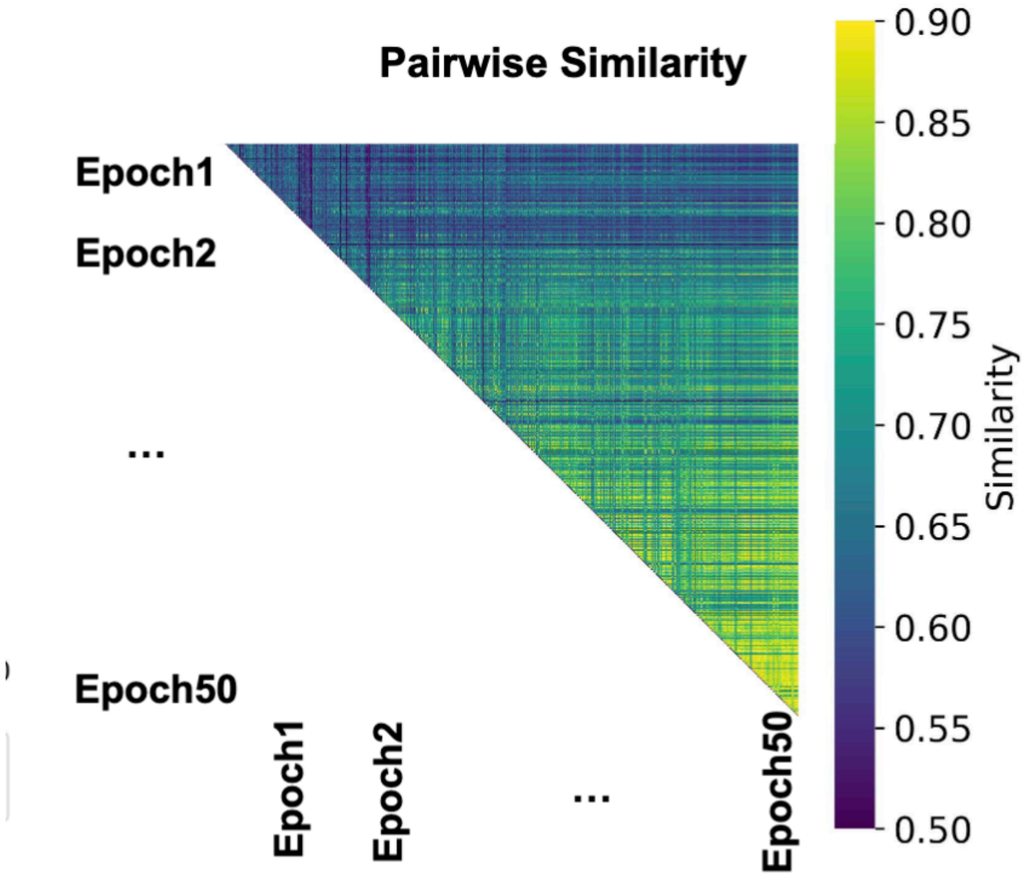
Problem Difficulty based Scheduling

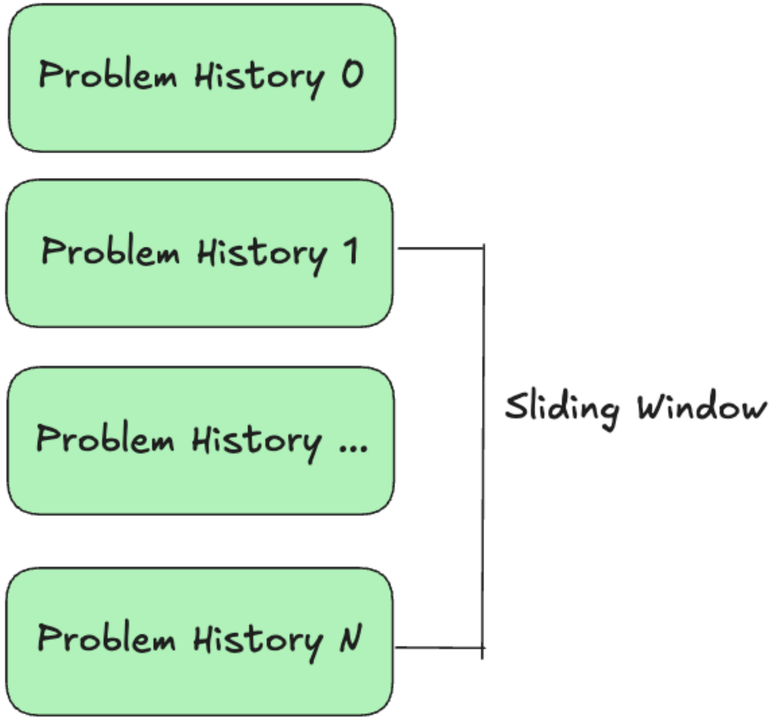
Schedule to minimize stragglers



Adapting to a changing model

Using a sliding window based on acceptance length





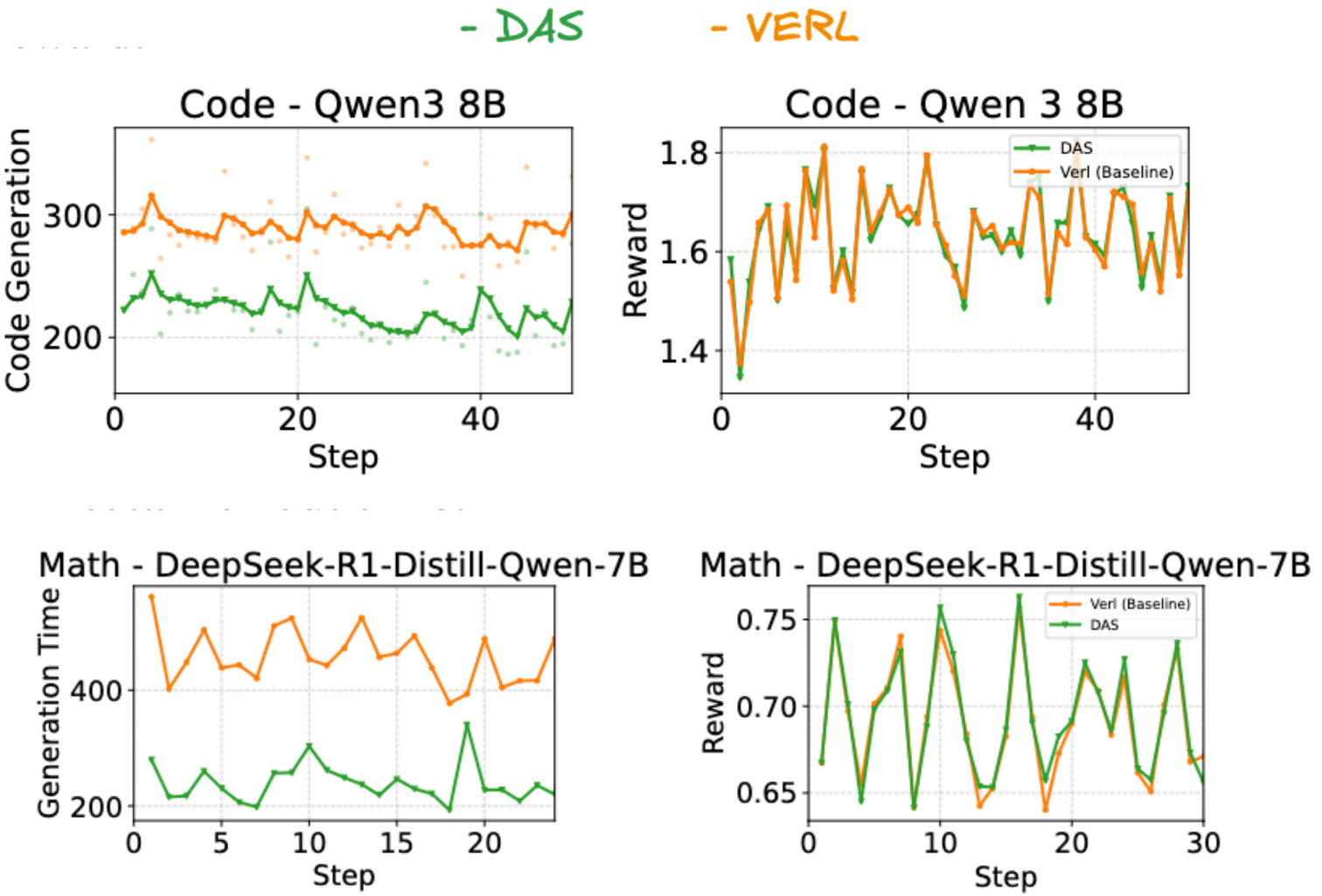
Results

2 Agentic Workloads:

- Math
- Code

GPUs: 2-6x 8 H100

Up to **50%** improvement



Takeaways

- RL training has a **long tail straggler problem** due to problem variance
- DAS takes advantage of suffix decoding and temporal request sharing
  - Utilizing a **Distribution aware** spec decoding
  - **Windowed aware** spec decoding
  - Length aware scheduling