Harvard John A. Paulson
School of Engineering
and Applied Sciences

Kempner
INSTITUTE
at Harvard University

Regression as Policy Optimization

Advantages In, Policies Out

Kianté Brantley, Assistant Professor, Harvard University

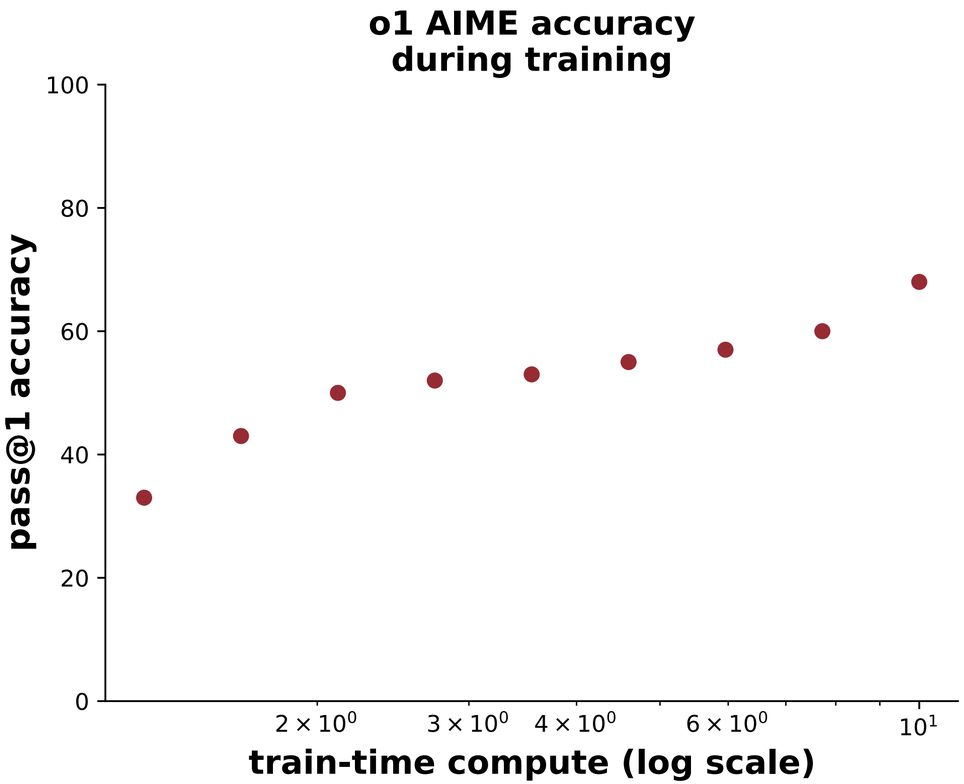
“A large language model is a deep learning model trained on massive text datasets to understand and generate human language.”

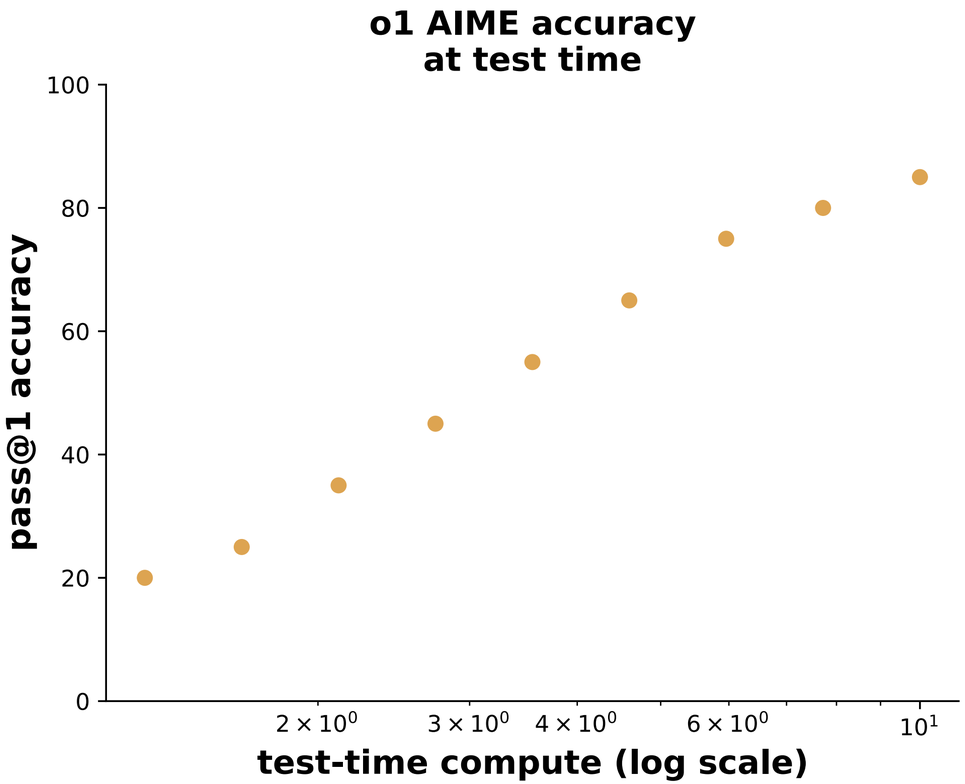
Large Language Model

How can we efficiently train LLMs to reason and act for complex tasks?

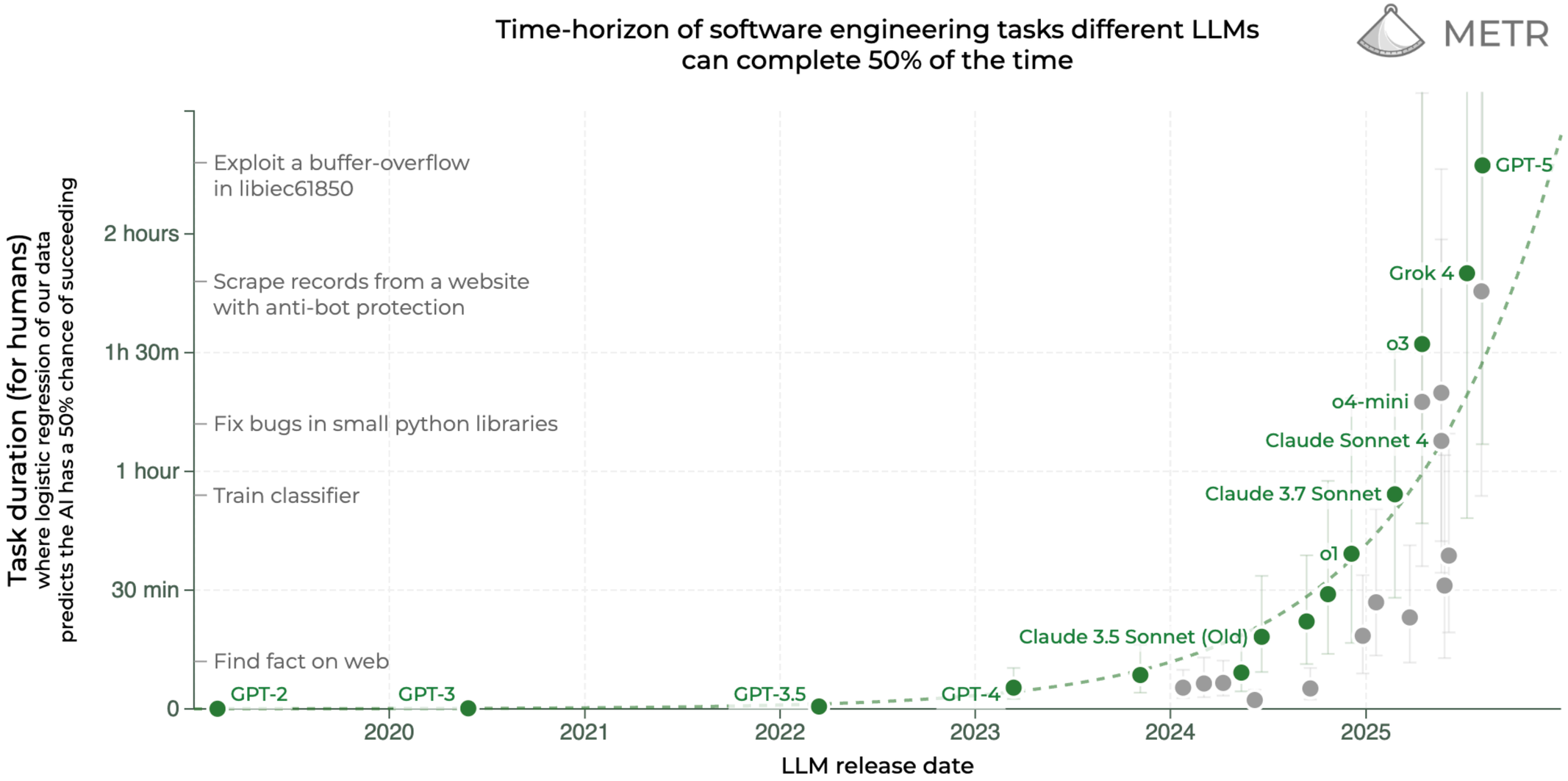
Learning to reason with LLMs

“We are introducing OpenAI o1, a new large language model trained with reinforcement learning to perform complex reasoning.” - Openai



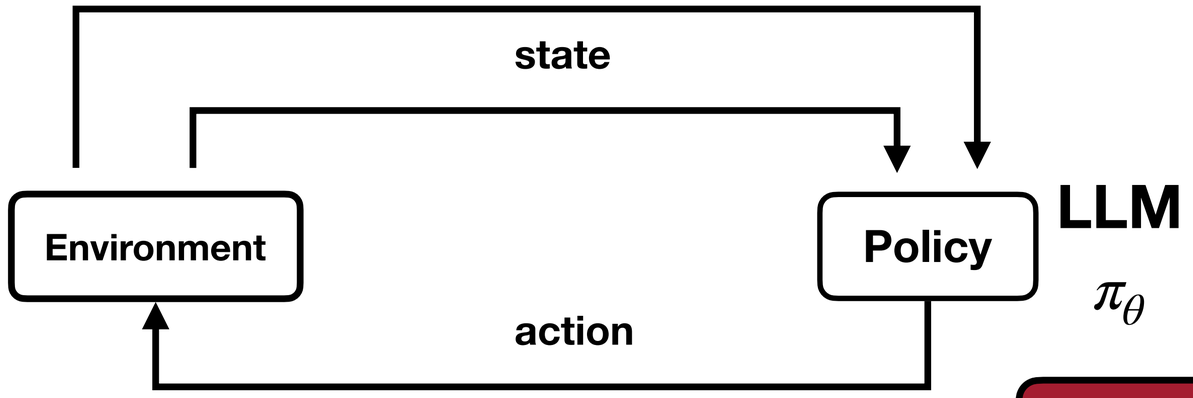


Learning to reason with LLMs



LLMs Post-Training MDP

reward (Verifiable tasks,e.g. math, or human preference)



How do we optimize π_θ ?

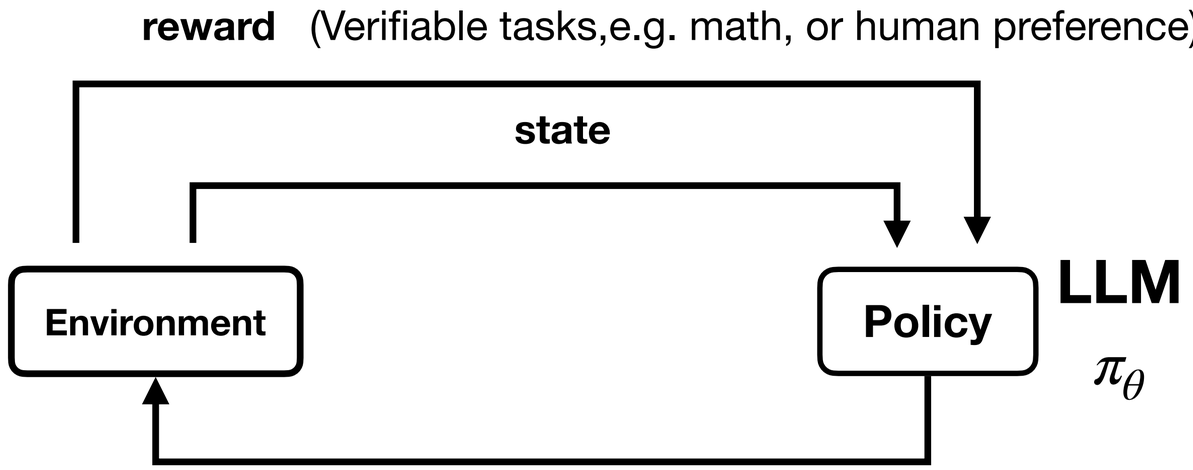
Prompt x:

What is 1+1?

Action y:

<think>
1 + 1 is a simple arithmetic addition.
Adding one to one equals two.
Therefore, the answer is 2.
</think>
$\boxed{2}$

LLMs Post-Training MDP



$$J(\pi) = \underbrace{\mathbb{E}_\pi \left[r(x, y) \right]}_{\text{Maximize reward}} - \frac{1}{\eta} \underbrace{\mathbb{D}_{\mathrm{KL}}(\pi \parallel \pi_{\mathrm{ref}})}_{\text{Minimal deviation}}$$

under

LLMs Post-Training

policy gradient

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\Big[\nabla_\theta \log \pi_\theta(y \mid x)\, r(x,y) - \beta \nabla_\theta \widehat{\mathrm{KL}}\Big]$$

LLM post-training focus has been on stabilizing policy-gradient methods.

$$J(\pi) = \underbrace{\mathbb{E}_\pi\big[r(x,y)\big]}_{\text{Maximize reward}} - \frac{1}{\eta}\underbrace{\mathbb{D}_{\mathrm{KL}}(\pi \,\|\, \pi_{\mathrm{ref}})}_{\text{Minimal deviation}}$$

under

LLMs Post-Training

policy gradient

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta}\Big[\nabla_\theta \log \pi_\theta(y \mid x)\, r(x,y) - \beta \nabla_\theta \widehat{\mathrm{KL}}\Big]$$

LLM post-training focus has been on stabilizing policy-gradient methods.

Are regression-based policy optimization more stable and compute efficient than policy gradients in LLM post-training?

$$J(\pi) = \underbrace{\mathbb{E}_\pi\big[r(x,y)\big]}_{\text{Maximize reward}} - \frac{1}{\eta}\underbrace{\mathbb{D}_{\mathrm{KL}}(\pi \,\|\, \pi_{\mathrm{ref}})}_{\text{Minimal deviation}}$$

under

Can regression-based policy optimization be more stable and more compute-efficient than policy gradients in LLM post-training?

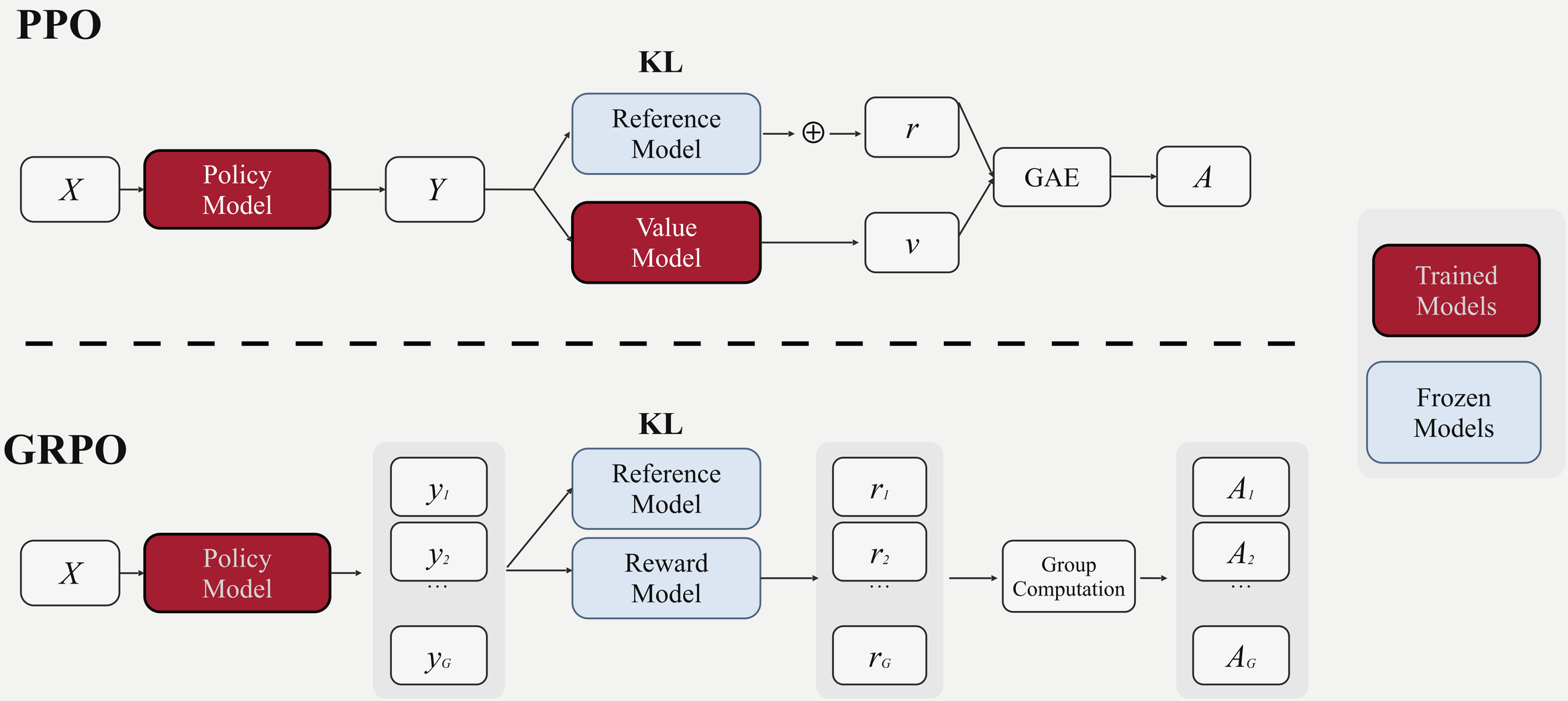
I. **A*PO:** Efficient training-time algorithm

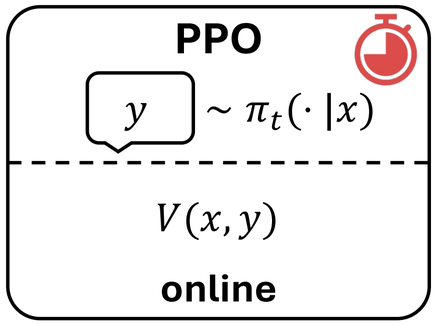
II. **OAPL:** Robust Off-Policy algorithm

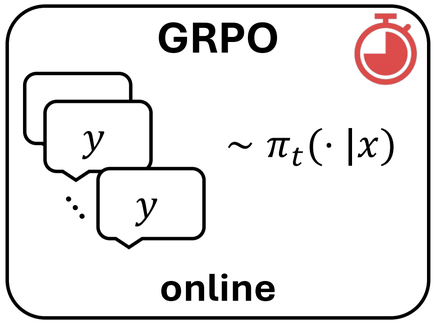
Accelerating RL for LLM Reasoning with Optimal Advantage Regression

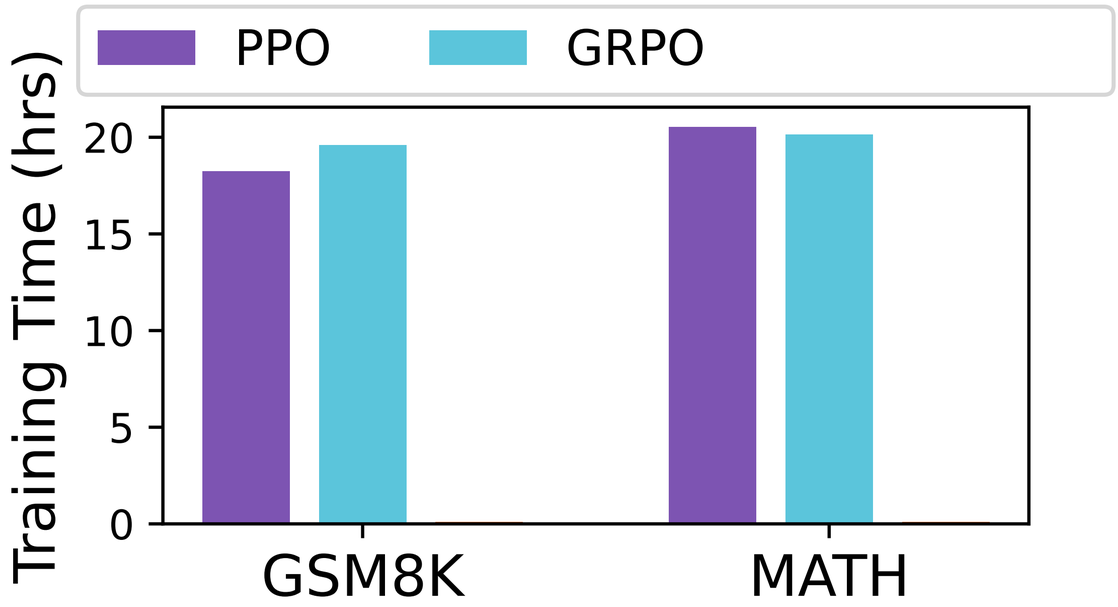
Kianté Brantley[1], **Mingyu Chen**[2], **Zhaolin Gao**[3],
Jason D. Lee[4], Wen Sun[3], **Wenhao Zhan**[4], Xuezhou Zhang[2]
(alphabetical order)

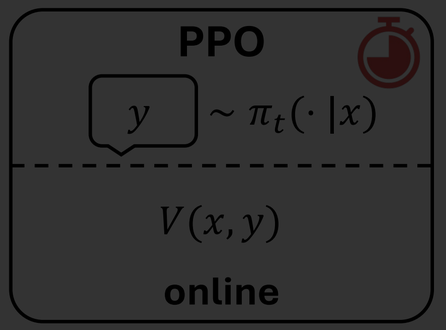
[1]Harvard University, [2]Boston University, [3]Cornell University, [4]Princeton University

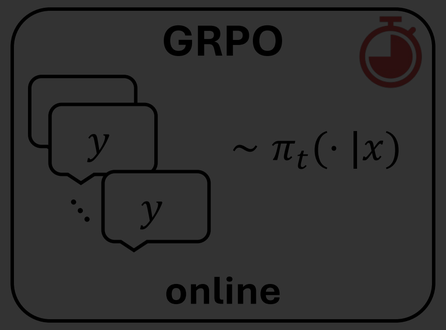

PPO
KL
X
Policy Model
Y
Reference Model
Value Model
r
v
GAE
A
Trained Models
Frozen Models
GRPO
KL
X
Policy Model
y_1
y_2
y_G
Reference Model
Reward Model
r_1
r_2
r_G
Group Computation
A_1
A_2
A_G


PPO
y
$\sim \pi_t(\cdot|x)$
$V(x, y)$
online


GRPO
y
y
$\sim \pi_t(\cdot|x)$
online

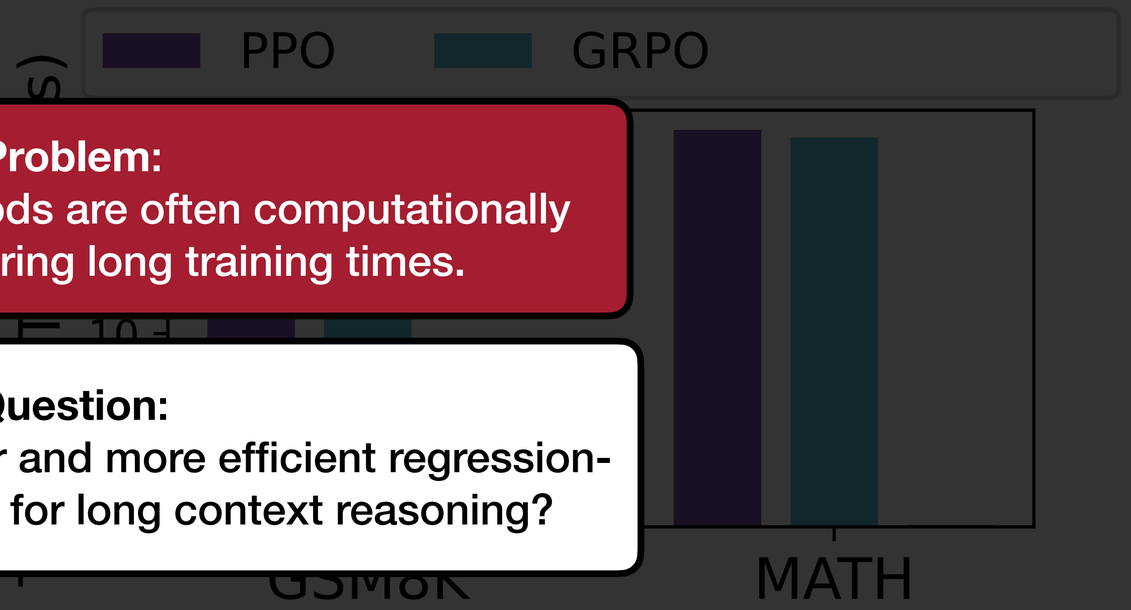

PPO
GRPO
Training Time (hrs)
0
5
10
15
20
GSM8K
MATH





Problem:
Policy gradient methods are often computationally expensive, requiring long training times.

Question:
Can we develop simpler and more efficient regression-based RL algorithms for long context reasoning?



RL Post-Training

$$\max_{\pi} \mathbb{E}_{\pi}\left[r(x,y)\right] - \frac{1}{\eta}\mathbb{D}_{\mathrm{KL}}(\pi \,\|\, \pi_{\mathrm{ref}})$$

Objective

$$\pi^*(y \mid x) = \frac{\pi_{\mathrm{ref}}(y \mid x)\, \exp(\eta r(x,y))}{Z(x)}, \qquad V^*(x) = \frac{1}{\eta}\log Z(x) = \frac{1}{\eta}\log \mathbb{E}_{\pi_{\mathrm{ref}}}[(y' \mid x)\, \exp(\eta r(x,y'))],$$

$$\pi^*(y \mid x) = \pi_{\mathrm{ref}}(y \mid x)\, \exp(\eta[r(x,y) - V^*(x)]),$$

Optimal Policy

RL Post-Training

$$\max_{\pi} \mathbb{E}_{\pi}\left[r(x,y)\right] - \frac{1}{\eta}\mathbb{D}_{\mathrm{KL}}(\pi \,\|\, \pi_{\mathrm{ref}})$$

Objective

$$\pi^*(y \mid x) = \frac{\pi_{\mathrm{ref}}(y \mid x)\, \exp(\eta r(x,y))}{Z(x)}, \qquad V^*(x) = \frac{1}{\eta}\log Z(x) = \frac{1}{\eta}\log \mathbb{E}_{\pi_{\mathrm{ref}}}[(y' \mid x)\, \exp(\eta r(x,y'))],$$

$$\pi^*(y \mid x) = \pi_{\mathrm{ref}}(y \mid x)\, \exp(\eta[r(x,y) - V^*(x)]),$$

Optimal Policy

In the bandit setting,
$r(x,y) = Q^{\pi}(x,y)$

$$\ell_t(\pi) := \mathbb{E}_{x,y\sim\pi_t(\cdot|x)}\left[\left(\eta \ln \frac{\pi(y\,|\,x)}{\pi_{\mathrm{ref}}(y\,|\,x)} - \left(r(x,y) - \hat{V}^*\right)\right)^2\right]$$

Regression Loss

A*PO Algorithm

1. Stage 1: Estimate $V^*(x)$ for all prompts

computed offline and kept fixed

Draw N i.i.d. samples $\{y_i\}_{i=1}^{N} \sim \pi_{\text{ref}}(\cdot \mid x)$

Compute $\hat{V}^{\star}(x) \leftarrow \beta \ln\left(\frac{1}{N}\sum_{i=1}^{N}\exp\left(\frac{r(x,y_i)}{\beta}\right)\right)$

biased estimate due to the nonlinearity of the ln function

2. Stage 2 (Online): On-policy Update with one rollout per prompt

Sample batch of training samples $D_t : \{x, y\}$

Regress to Estimated $V^*(x)$

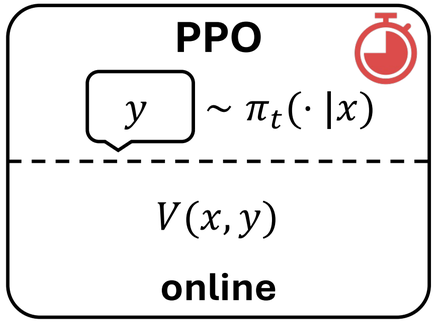
gradients only flow through π_θ online now

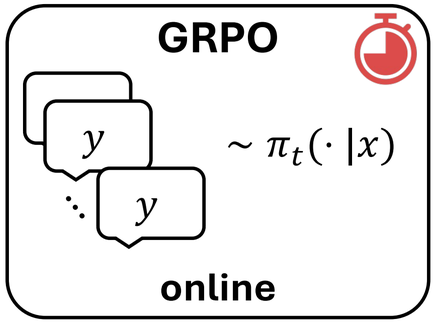
$$\ell(\pi_\theta) := \mathbb{E}_{x,y\sim D_t}\left[\left(\eta \ln \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)} - \left(r(x,y) - \hat{V}^*\right)\right)^2\right]$$

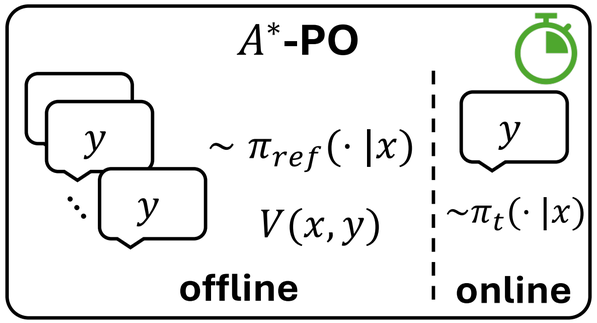
approximates the optimal advantage $A^* = r(x,y) - \hat{V}^*(x)$

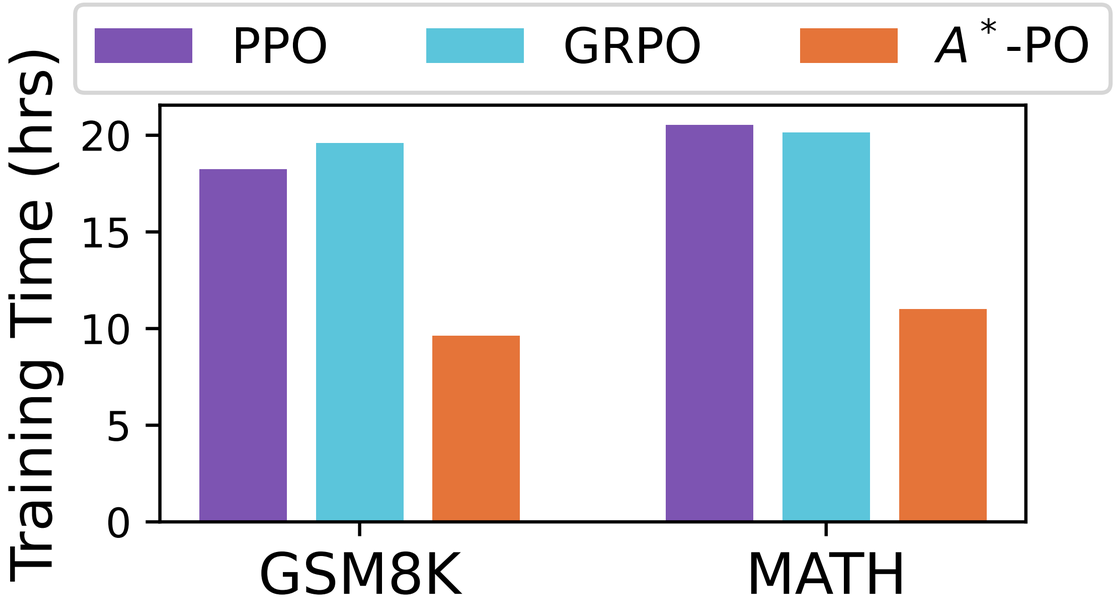
Predictor

Target


PPO
y
$\sim \pi_t(\cdot|x)$
$V(x,y)$
online


GRPO
y
y
$\sim \pi_t(\cdot|x)$
online


A^*-PO
y
y
$\sim \pi_{ref}(\cdot|x)$
$V(x,y)$
offline
y
$\sim \pi_t(\cdot|x)$
online


PPO
GRPO
A^*-PO
Training Time (hrs)
0
5
10
15
20
GSM8K
MATH

Experiments: Advanced Reasoning on MATH

MATH dataset [Hendrycks et al.,2021]:
Consists of challenging problems from high school math competitions.

Base Model Type:
Pre-trained Qwen2.5 models

Model Size	Method	Time (hrs)	Memory (%)	KL $(\pi \| \| \pi_{\text{ref}})$	MATH500
1.5B	*base*	/	/	/	45.8
	PPO	10.84	42.64	0.151	57.0
	GRPO	15.01	<u>42.19</u>	<u>0.091</u>	**58.0**
	REBEL	<u>10.33</u>	63.34	0.098	<u>57.8</u>
	$A^\star$-PO	**6.92**	**41.59**	**0.069**	<u>57.8</u>

Experiments: Advanced Reasoning on MATH

MATH dataset [Hendrycks et al.,2021]:
Consists of challenging problems from high school math competitions.

Base Model Type:
Pre-trained Qwen2.5 models

Model Size	Method	Time (hrs)	Memory (%)	KL $(\pi \| \| \pi_{\text{ref}})$	MATH500
3B	*base*	/	/	/	50.8
	PPO	13.59	54.95	0.111	65.8
	GRPO	18.26	<u>49.75</u>	0.102	66.0
	REBEL	<u>12.88</u>	74.32	<u>0.099</u>	**67.0**
	$A^\star$-PO	**8.78**	**49.28**	**0.082**	<u>66.2</u>

Experiments: Advanced Reasoning on MATH

MATH dataset [Hendrycks et al.,2021]:
Consists of challenging problems from high school math competitions.

Base Model Type:
Pre-trained Qwen2.5 models

Model Size	Method	Time (hrs)	Memory (%)	KL $(\pi \| \| \pi_{\text{ref}})$	MATH500
7B	*base*	/	/	/	62.8
	PPO	20.53	92.81	0.133	74.4
	GRPO	20.15	<u>77.82</u>	0.172	73.2
	REBEL	<u>14.67</u>	98.77	<u>0.124</u>	<u>74.6</u>
	$A^\star$-PO	**11.01**	**76.57**	**0.078**	**76.2**

Experiments: Advanced Reasoning on MATH

MATH dataset [Hendrycks et al.,2021]:
Consists of challenging problems from high school math competitions.

Base Model Type:
Pre-trained Qwen2.5 models

Model Size	Method	Time (hrs)	Memory (%)	KL $(\pi \| \| \pi_{\text{ref}})$	MATH500
1.5B	*base*	/	/	/	45.8
	PPO	10.84	42.64	0.151	57.0
	GRPO	15.01	<u>42.19</u>	<u>0.091</u>	**58.0**
	REBEL	<u>10.33</u>	63.34	0.098	<u>57.8</u>
	$A^\star$-PO	**6.92**	**41.59**	**0.069**	<u>57.8</u>
3B	*base*	/	/	/	50.8
	PPO	13.59	54.95	0.111	65.8
	GRPO	18.26	<u>49.75</u>	0.102	66.0
	REBEL	<u>12.88</u>	74.32	<u>0.099</u>	**67.0**
	$A^\star$-PO	**8.78**	**49.28**	**0.082**	<u>66.2</u>
7B	*base*	/	/	/	62.8
	PPO	20.53	92.81	0.133	74.4
	GRPO	20.15	<u>77.82</u>	0.172	73.2
	REBEL	<u>14.67</u>	98.77	<u>0.124</u>	<u>74.6</u>
	$A^\star$-PO	**11.01**	**76.57**	**0.078**	**76.2**

Experiments: Long-Context Reasoning

OpenR1-Cleaned Dataset [From VGS]:

Randomly selected subset of 5,000 problems
Max Generation Length 16,384

Evaluate:

AIME (American Invitational Mathematics Examination) an annual US high school math competition

Base Model Type:

DeepSeek-R1 distilled Qwen-1.5B model

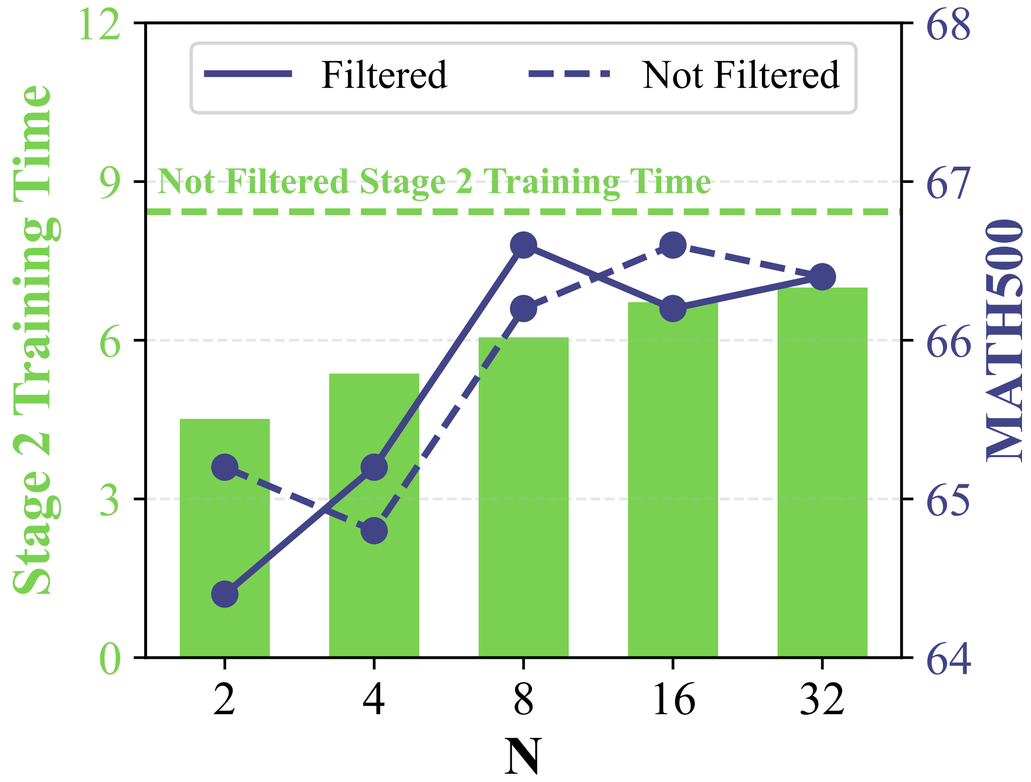
Method	AIME 24		AIME 25	
	Avg@32	Pass@32	Avg@32	Pass@32
base	21.25	60.00	21.15	43.33
PPO	**30.94**	**80.00**	25.12	**46.67**
GRPO	30.00	73.33	25.00	43.33
$A^\star$-PO	29.17	70.00	**26.67**	**46.67**

Ablation: Filtering Out Hard Problems.

Can training efficiency improve to filter out examples whose pass@N is zero?

Filter Pass@N can be done at stage 1

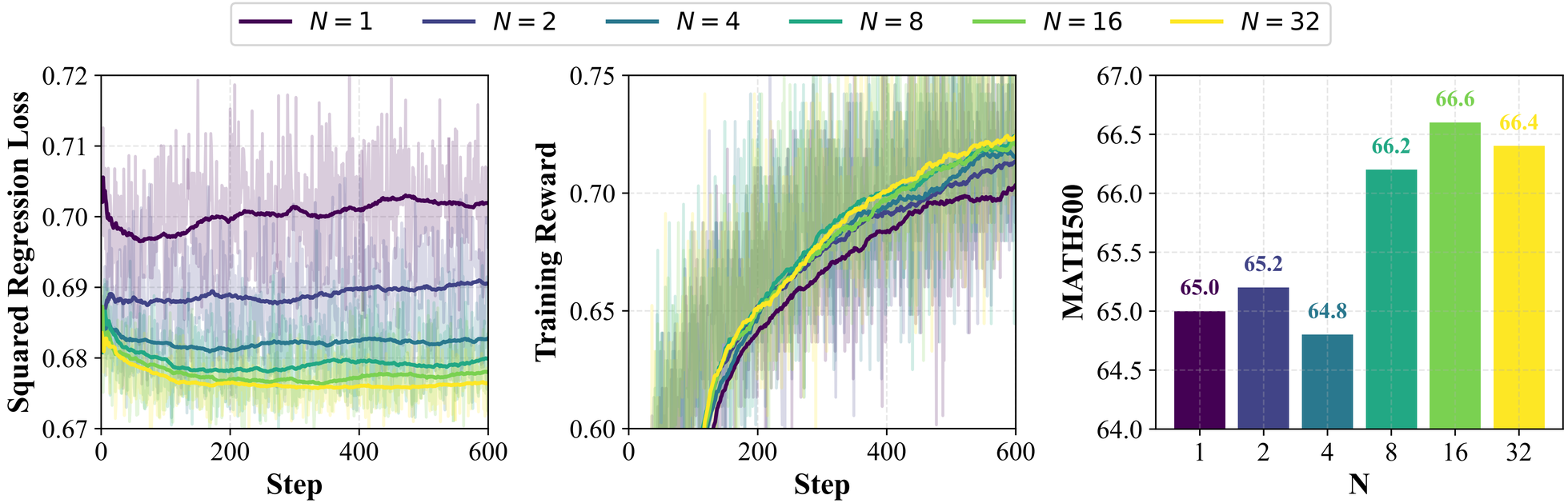
Problems that the reference policy is unable to solve are unlikely to be solved through RL post-training.



Ablation: Number of Generations

How many generations are used during stage 1 to estimate V^* on the MATH dataset?

We vary the number of generations N from 1 to 32.



Takeaways

Idea:

- Precompute $\hat{V}^{\star}$offline, then regress to $r - \hat{V}^{\star}$online.

Intuition:

- Replaces critics / multi-rollout advantage estimation with a much simpler update

Results:

- Competitive reasoning performance at much lower cost—faster, more memory-efficient, and scalable

Can regression-based policy optimization be more stable and more compute-efficient than policy gradients in LLM post-training?

I. **A*PO:** Efficient training-time algorithm

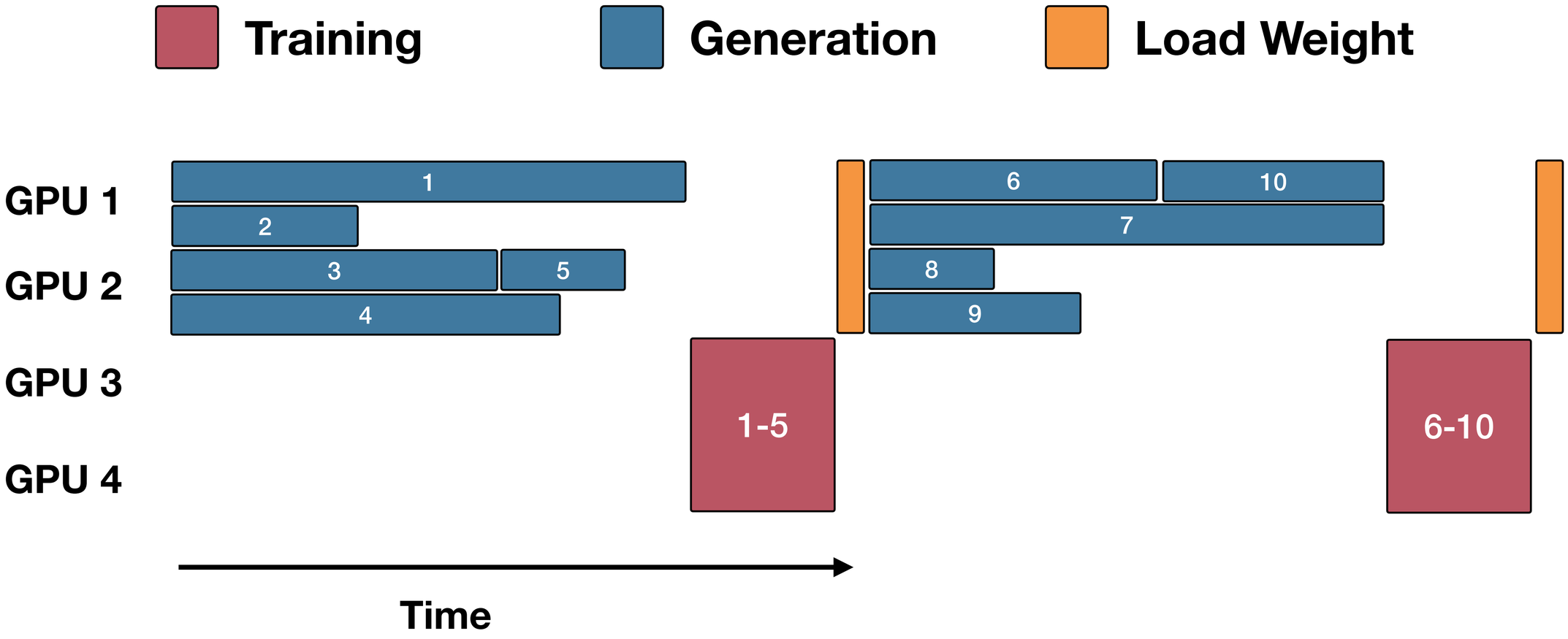
II. **OAPL:** Robust Off-Policy algorithm

LLMs Can Learn to Reason Via Off-Policy RL

Daniel Ritter[13], **Owen Oertell**[12], **Bradley Guo**[1],
Jonathan D. Chang[2], Kianté Brantley[3], Wen Sun[2]

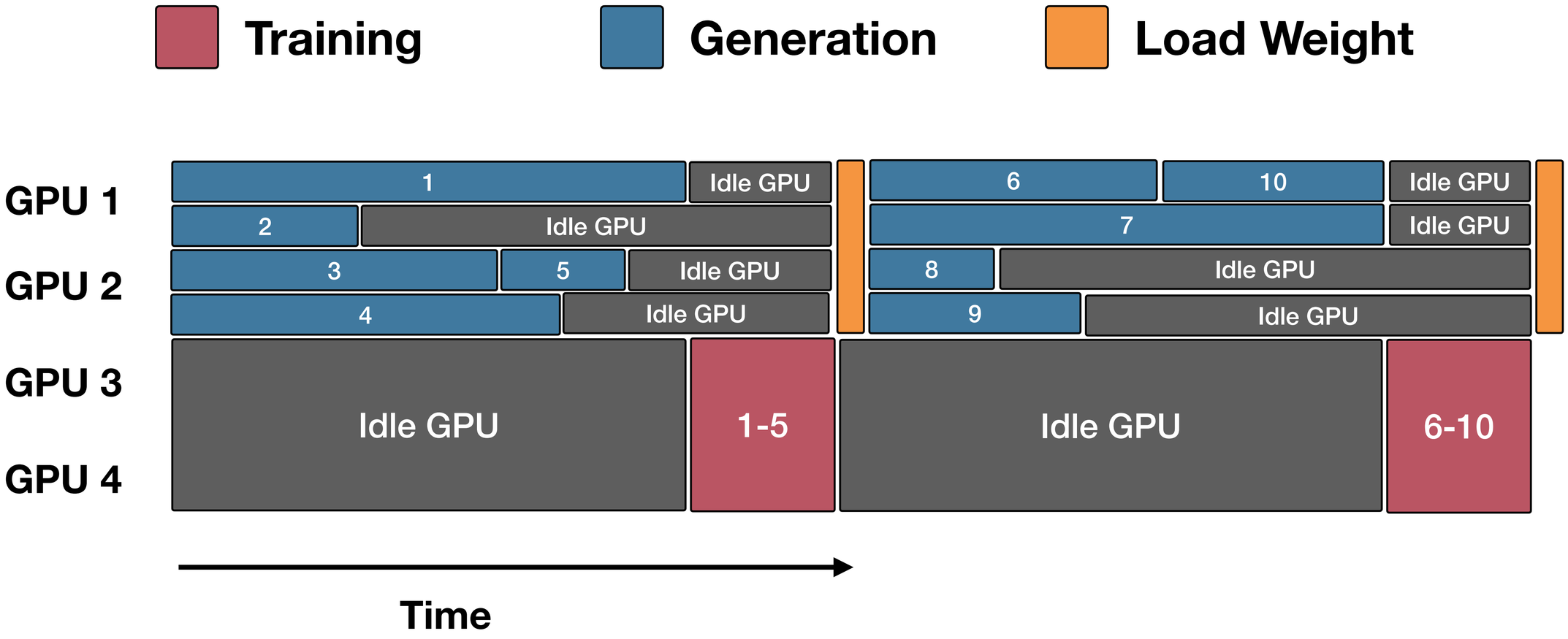
[1]Cornell University, [2]Databricks, [3]Harvard University

Synchronous RL Post-Training



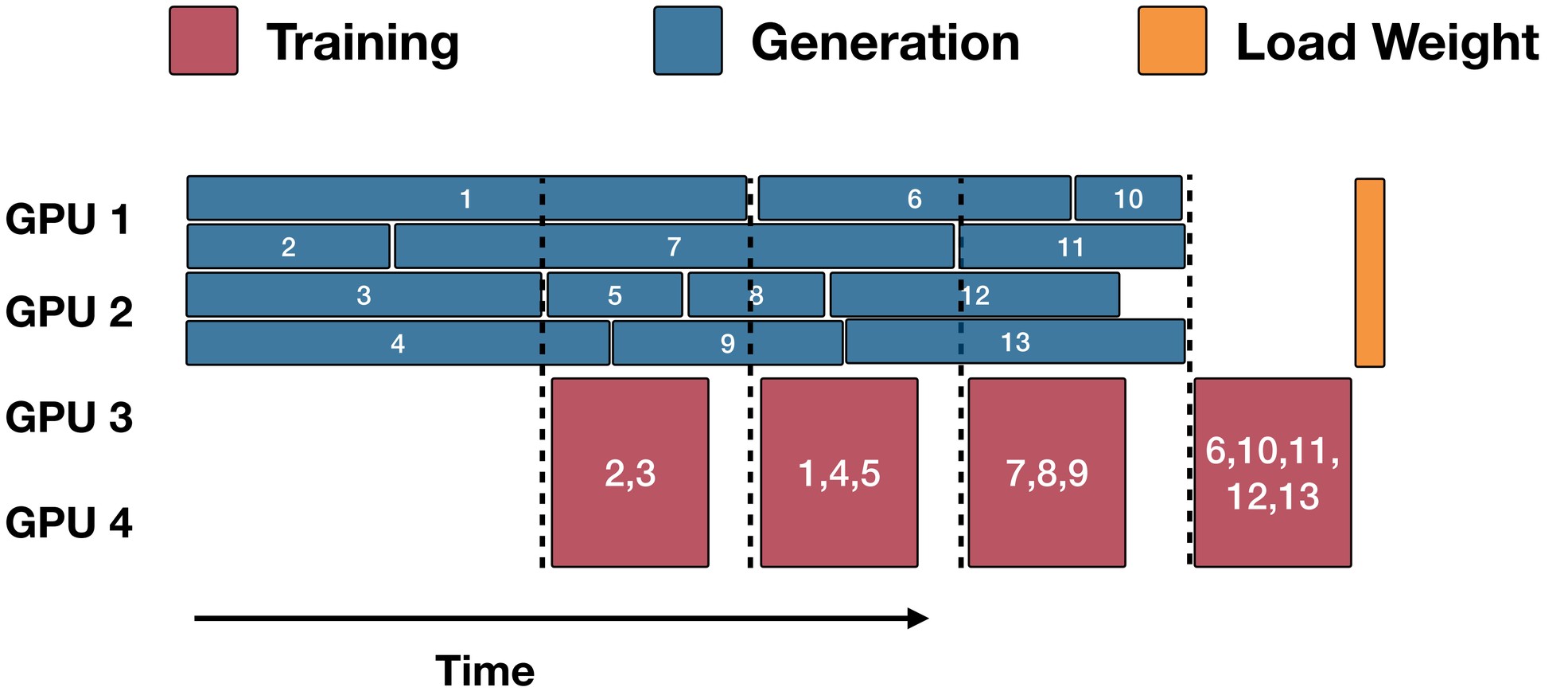
AReaL: A Large-Scale Asynchronous Reinforcement Learning System for Language Reasoning Fu et al. 2026

Synchronous RL Post-Training



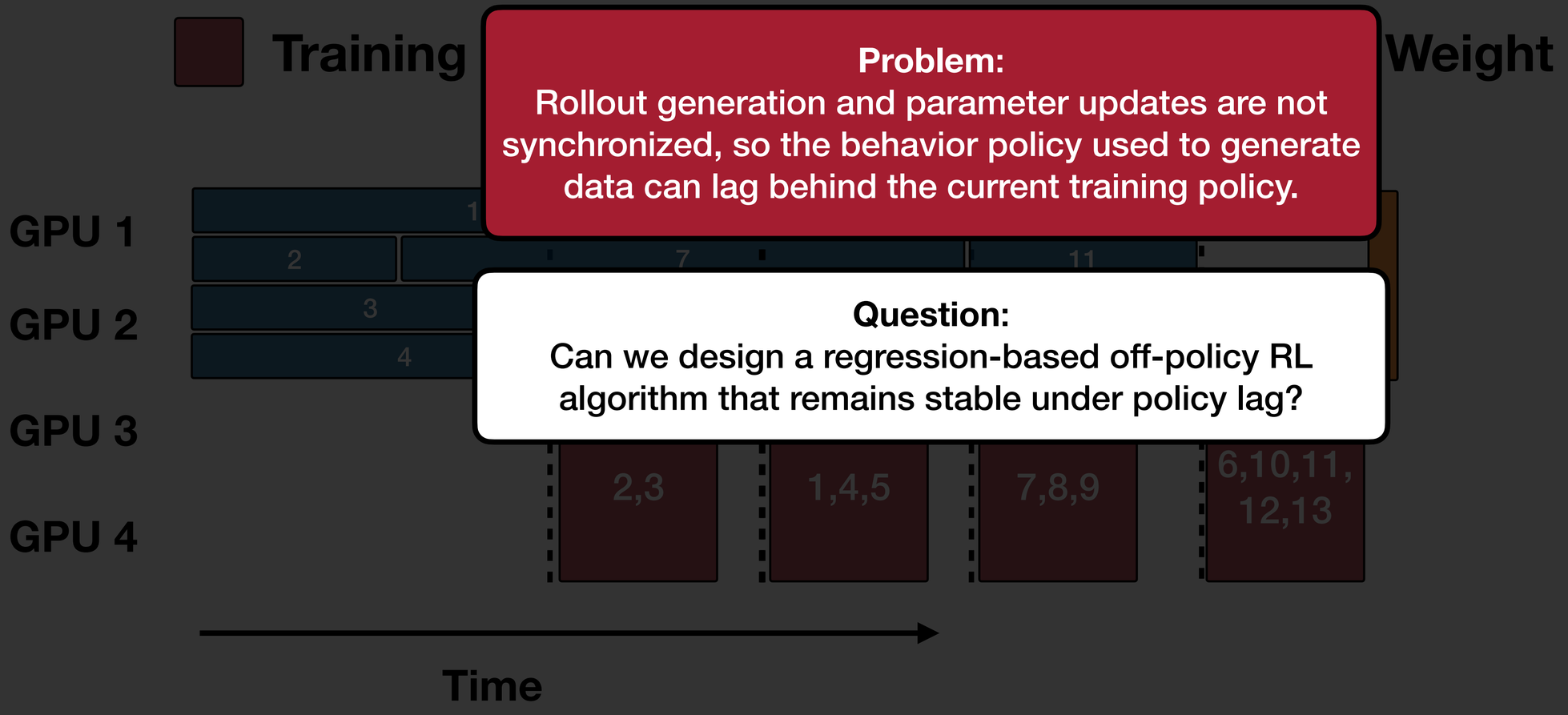
AReaL: A Large-Scale Asynchronous Reinforcement Learning System for Language Reasoning Fu et al. 2026

Asynchronous RL Post-Training



AReaL: A Large-Scale Asynchronous Reinforcement Learning System for Language Reasoning Fu et al. 2026

Asynchronous RL Post-Training



Problem:
Rollout generation and parameter updates are not synchronized, so the behavior policy used to generate data can lag behind the current training policy.

Question:
Can we design a regression-based off-policy RL algorithm that remains stable under policy lag?

AReaL: A Large-Scale Asynchronous Reinforcement Learning System for Language Reasoning Fu et al. 2026

RL Post-Training

$$\max_{\pi} \mathbb{E}_{\pi}\left[r(x,y)\right] - \frac{1}{\eta}\,\mathbb{D}_{\mathrm{KL}}(\pi \,\|\, \pi_{\mathrm{vllm}})$$

Objective

$$\pi^*(y \mid x) = \frac{\pi_{\mathrm{vllm}}(y \mid x)\,\exp(\eta r(x,y))}{Z(x)}, \qquad V^*(x) = \frac{1}{\eta}\log Z(x) = \frac{1}{\eta}\log \mathbb{E}_{\pi_{\mathrm{vllm}}}[(y' \mid x)\,\exp(\eta r(x,y'))],$$

$$\pi^*(y \mid x) = \pi_{\mathrm{vllm}}(y \mid x)\,\exp(\eta[r(x,y) - V^*(x)]),$$

Optimal Policy

$$\ell_t(\pi) := \mathbb{E}_{x,y\sim\pi_t(\cdot|x)}\left[\left(\eta \ln \frac{\pi(y\,|\,x)}{\pi_{\mathrm{vllm}}(y\,|\,x)} - \left(r(x,y) - \hat{V}^*\right)\right)^2\right]$$

Regression Loss

OAPL

Optimal Advantage-Based Policy Optimization with Lagged Inference Policy

Synchronize $\pi_{\text{vllm}} \leftarrow \pi$

For $t = 1 \rightarrow T$

Sample a batch $\{x, \{y_i\}_{i=1}^{G}\}$ from π_{vllm} and store it in D

Optimize π using D

If t mod $L = 0$ **then**

Synchronize $\pi_{\text{vllm}} \leftarrow \pi$

e.g., Larger L means more potential policy lag/staleness between π and π_{vllm}

Baseline idea:

correct stale rollouts with an extra importance ratio.

Importance sampling between π_θ and π_{prox}

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y_t\sim\pi_{\text{vllm}}}\left[\sum_{t=1}^{H} \frac{\pi_{\text{prox}}}{\pi_{\text{vllm}}} \min\left(u_t^{\text{prox}}(\theta)\,\hat{A}_t,\ \text{clip}\left(u_t^{\text{prox}}(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t\right)\right]$$

Importance sampling between π_{vllm} and π_{prox}

Experiments: Advanced Reasoning on MATH

Datasets:

Training Deepscaler Dataset

- Approximately 40,000 unique problem-answer pairs
- AIME problems from 1984-2023
- AMC problems prior to 2023
- Omni-MATH datasets
- Slow Thinking with LLMs datasets

Base Model Type:

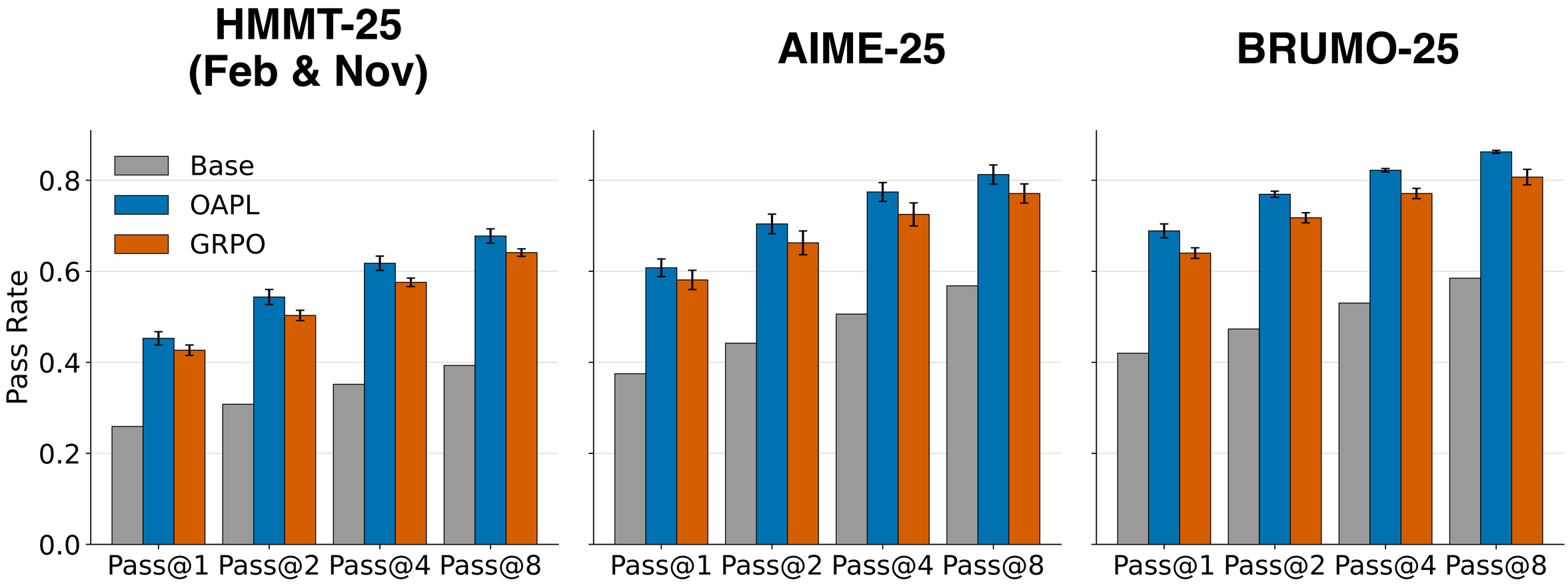
Qwen3-4B-Thinking-2507

Experimental Setup:

OAPL: Set $L = 50$, meaning s**ynchronize** $\pi_{\text{vllm}} \leftarrow \pi$ every 50 iterations

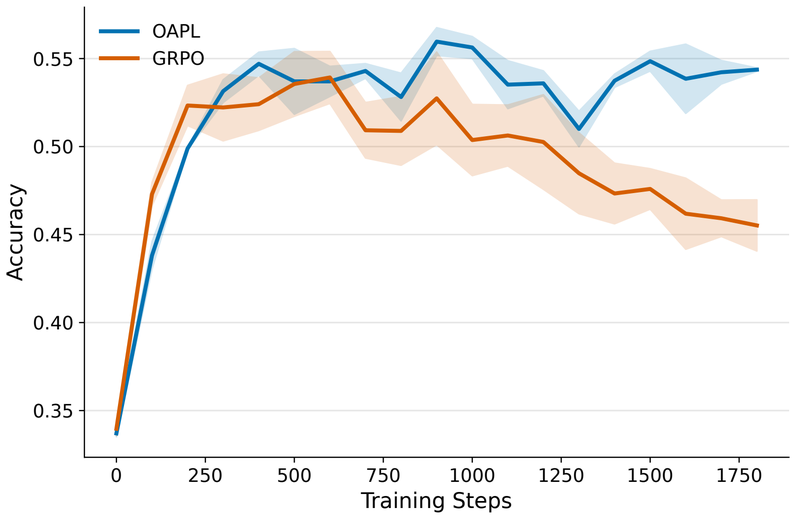
GRPO: Off-by-one asynchronous training

Experiments: Advanced Reasoning on MATH

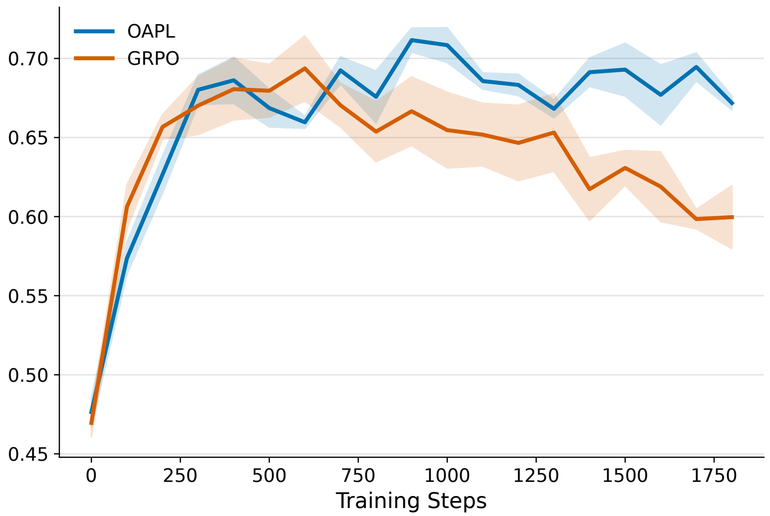


Experiments: Advanced Reasoning on MATH

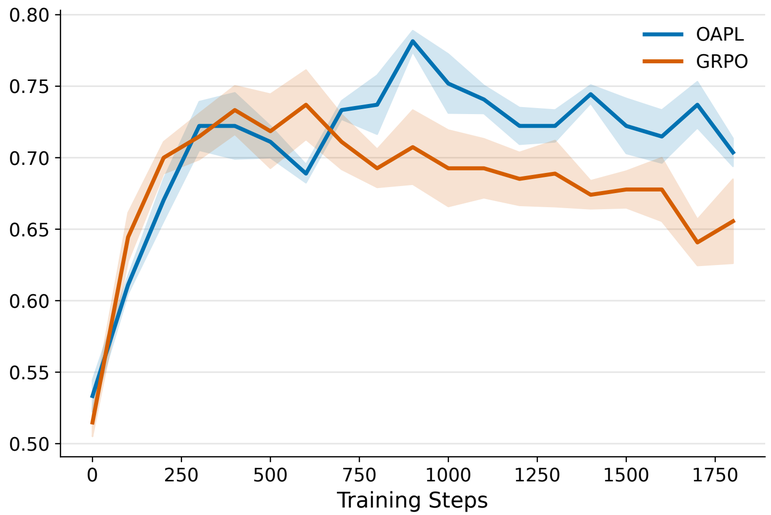
Average Accruacy across (HMMT-25, AIME-25, BRUMO-25)



pass@1



pass@5

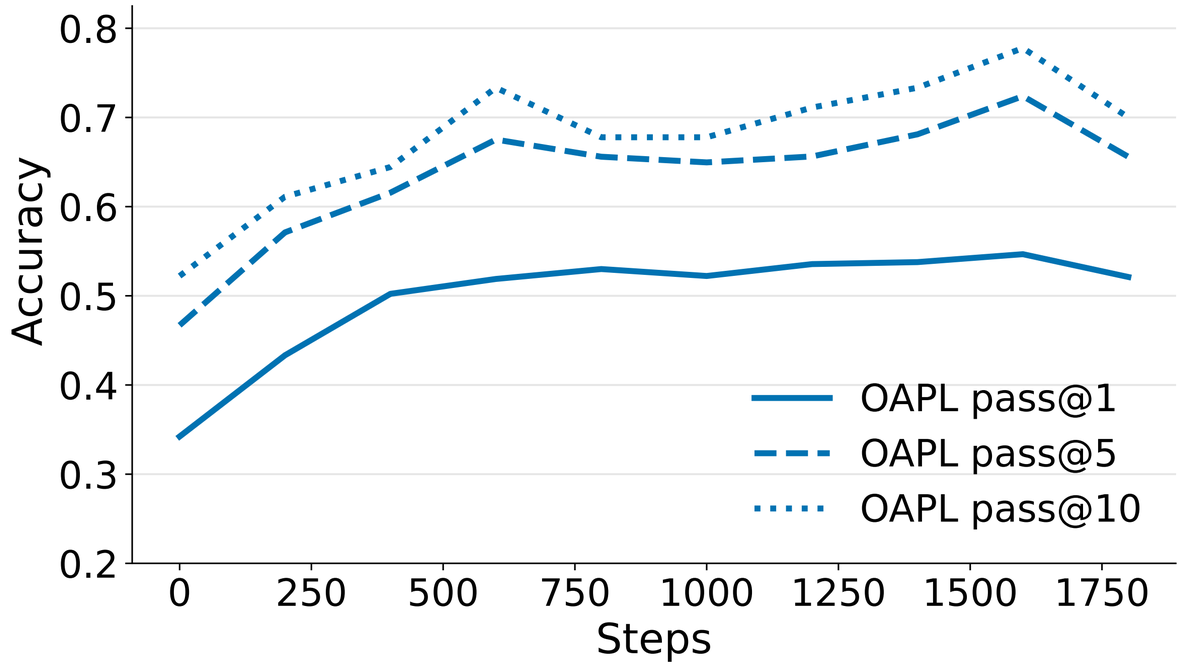


pass@10

Ablation: Training stability with large policy lag

Can OAPL still learn stably when the inference engine policy lags significantly behind the trainer?

We further evaluate OAPL, i.e., we only synchronize π every 100 iterations.



Experiments: DeepCoder

Datasets:

Training DeepCoder Dataset

- Generate an offline dataset of 8 responses for every prompt

Evaluation:

- 279 LiveCodeBench problems

Base Model Type:

DeepSeek-R1-Distill-Qwen-14B

Experimental Setup:

OAPL:

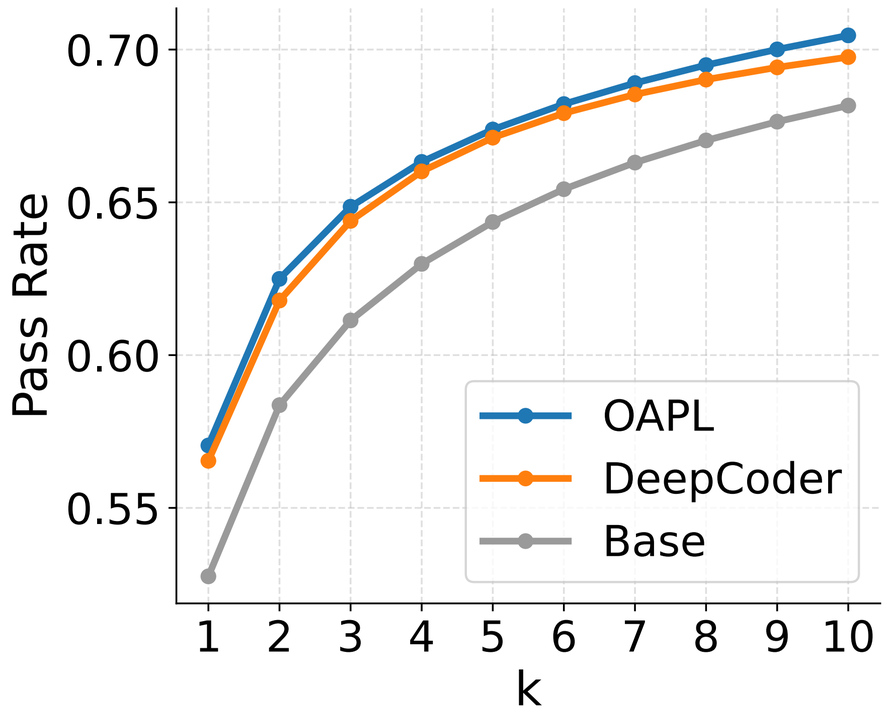
- Train the base model on this dataset with OAPL for 1 epoch without synchronizing ($L = 1$)
- **Then** generate a new offline dataset from a subset and continue training for an additional four epochs

GRPO:

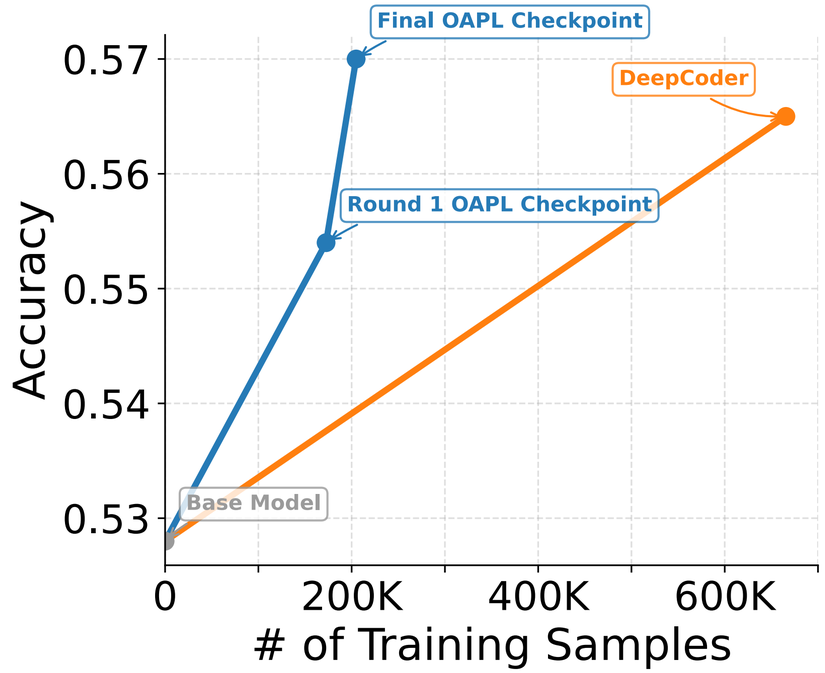
- Use the publicly available checkpoints from DeepCoder

Experiments: DeepCoder

Pass@k Scaling



Sample Efficiency



Takeaways

Idea:

- Use the lagged inference policy as the KL anchor and train with a regression loss

Intuition:

- Avoids importance weighting and stays stable even when the data is stale

Results:

- Better robustness under policy lag, stronger Pass@k scaling, and much better sample efficiency

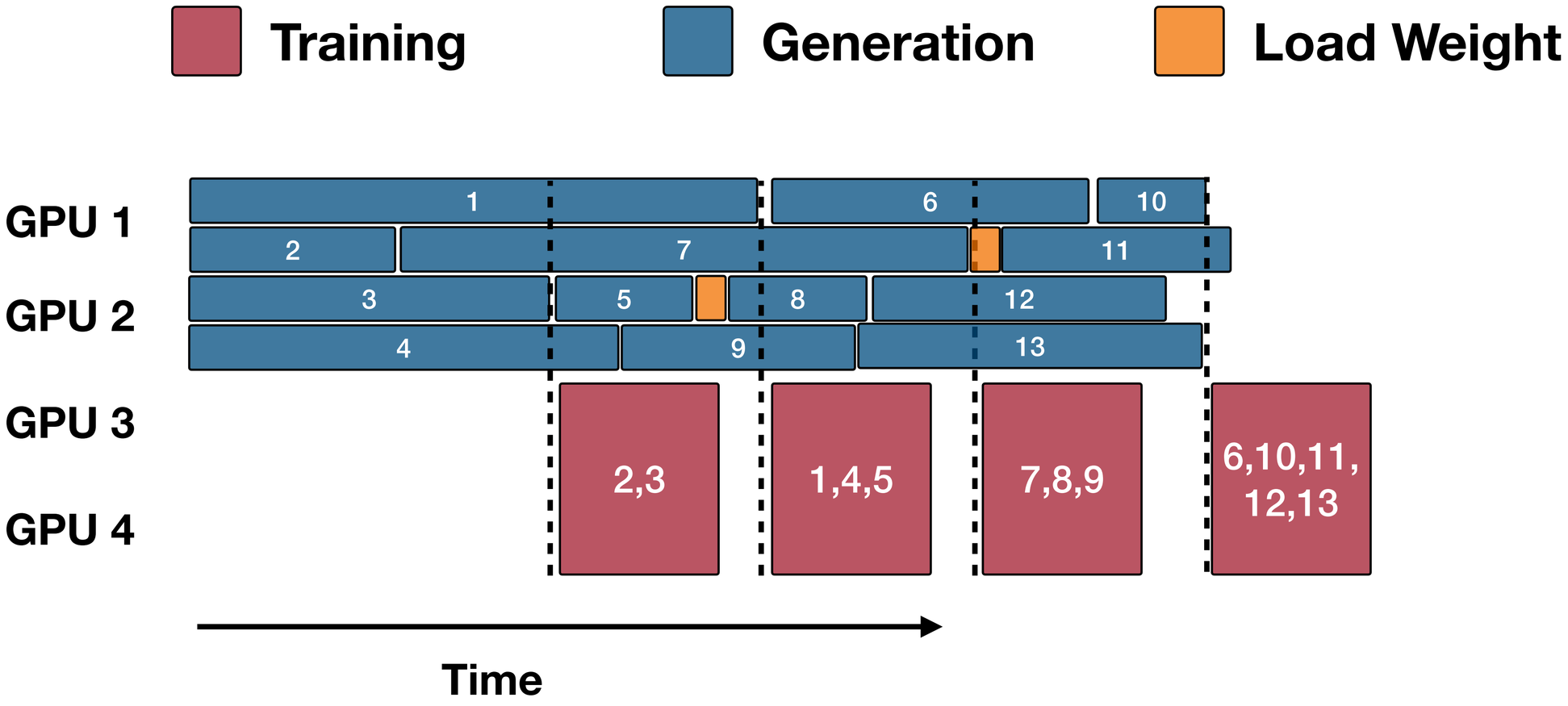
Can regression-based policy optimization be more stable and more compute-efficient than policy gradients in LLM post-training?

I. **A*PO:** Efficient training-time algorithm

II. **OAPL:** Robust Off-Policy algorithm

Yes, regression-based policy optimization can be more efficient (A*PO) and more robust to off-policy lag (OAPL).

Future Thoughts



AReaL: A Large-Scale Asynchronous Reinforcement Learning System for Language Reasoning Fu et al. 2026

Acknowledgement