Advanced Topics in Reinforcement Learning

Lecture 22: RL for LLMs

Josiah Hanna
University of Wisconsin — Madison

Announcements

- Work on final projects.
- Read “Empirical Design in Reinforcement Learning”

Learning Outcomes

After today, you will be able to:

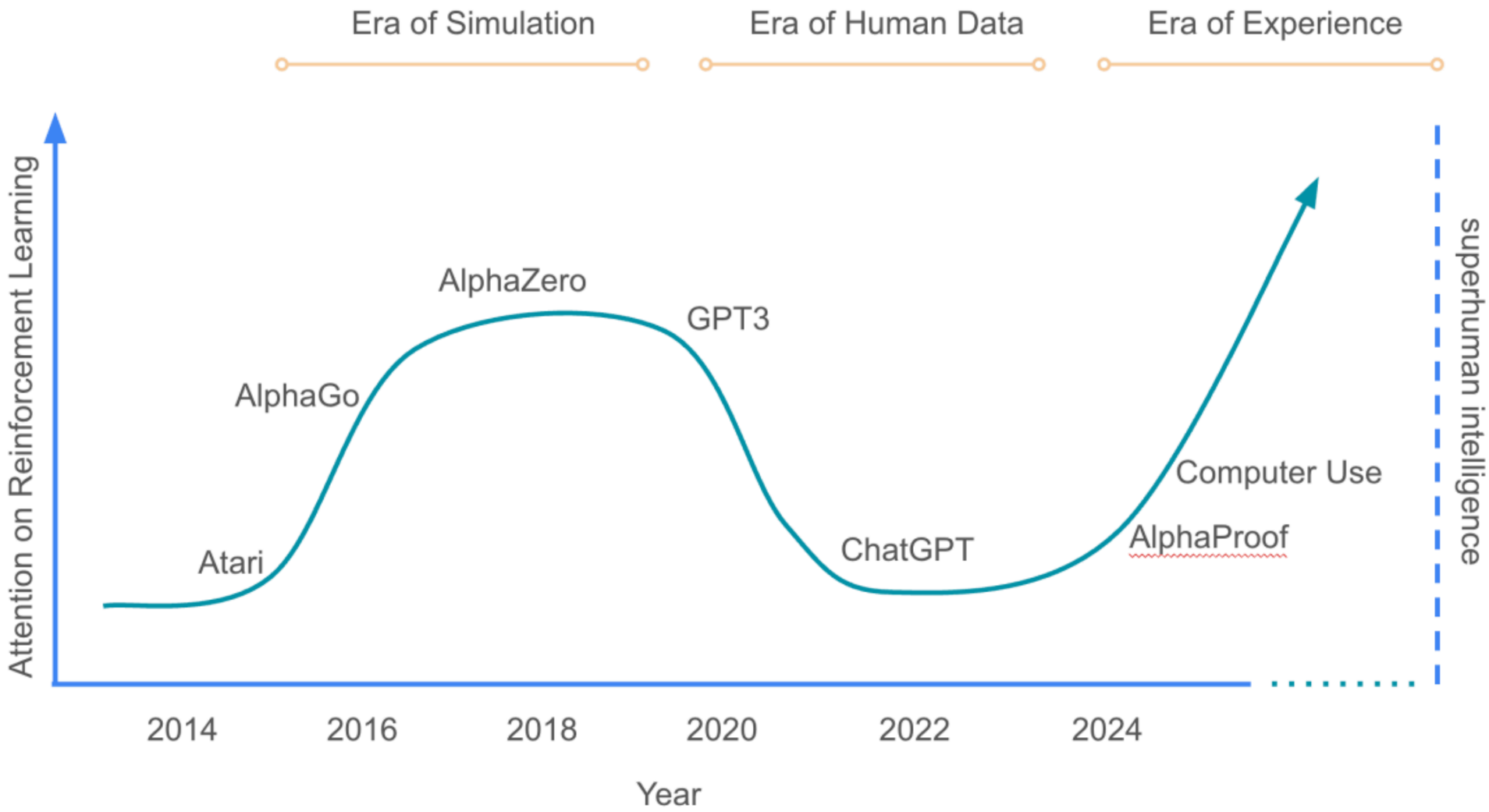
1. Formalize fine-tuning of LLMs as an RL problem.

2. Identify key features of RL algorithms for LLMs.

Motivation

- LLMs are trained with self-supervised pre-training.
  - Models distribution of text but not sufficient for building agents.
- Supervised fine-tuning?
  - Often helpful but requires demonstration data, ratings, or rankings.

Motivation



Welcome to the Era of Experience. Silver and Sutton (2025)

RL for LLMs

- LLMs are autoregressive models of language or formally: $\pi : \mathscr{H} \rightarrow \mathscr{W}$.
- Can view as policies that map histories of text (i.e., context) to a probability distribution over next tokens to output.
- Often we discuss the probability of an entire response: $p_\theta(y \mid x)$ where x is a prompt and y is the full response that consists of a sequence of output tokens.

$$p_\theta(y \mid x) = \prod_{t=0}^{l} \pi_\theta(y_t \mid x, y_{0:t-1})$$

RL for LLMs

- How do we define states?
- How do we define actions?
- What are transitions?
- What are rewards?
- Discount or no?

RL for LLMs

- Goal: given a reward function, $r(x, y)$, apply RL to maximize the probability of y in response to x.
  - Supporting goal: keep $p_\theta(y \mid x)$ close to the initial model.
- $J(\theta) = \mathbb{E}[r(x, y)]$
- $\nabla J(\theta) = \mathbb{E}[r(x, y) \nabla \log p_\theta(y \mid x)]$

$$= \mathbb{E}[r(x, y) \sum_{t=0}^{l} \nabla \log \pi_\theta(y_t \mid x, y_{0:t-1})]$$

$$p_\theta(y \mid x) = \prod_{t=0}^{l} \pi_\theta(y_t \mid x, y_{0:t-1})$$

RL for LLMs

Policy gradient is the gradient of a surrogate loss function:

$$L(\theta) = \mathbb{E}_{y \sim p_{\texttt{ref}}}[r(x, y) \log p_\theta(y \mid x)]$$

$$\nabla J(\theta) = \nabla L(\theta)$$

- If $p_{\texttt{ref}}$ is the current model's response distribution then we are doing on-policy policy gradient.
- If $p_{\texttt{ref}}$ is any other model, we are doing off-policy policy gradient.
- If $p_{\texttt{ref}}$ is responses generated by a human expert and $r(x, y) = 1$ then we are doing imitation learning (supervised finetuning).

Vanilla Policy Gradient for LLMs

$$J_{\mathtt{RL}}(\theta) = \mathbb{E}[q \sim P(Q), o \sim \pi_\theta(o \,|\, q))] \frac{1}{|\, o \,|} \sum_{t=0}^{|o|} A_t \log \pi_\theta(o_t \,|\, q, o_{<t})$$

- $A_t \approx q_\pi(o_t \,|\, q, o_{<t}) - v_\pi(q, o_{<t})$ estimates the *advantage* of outputting o_t vs sampling the next token from π.

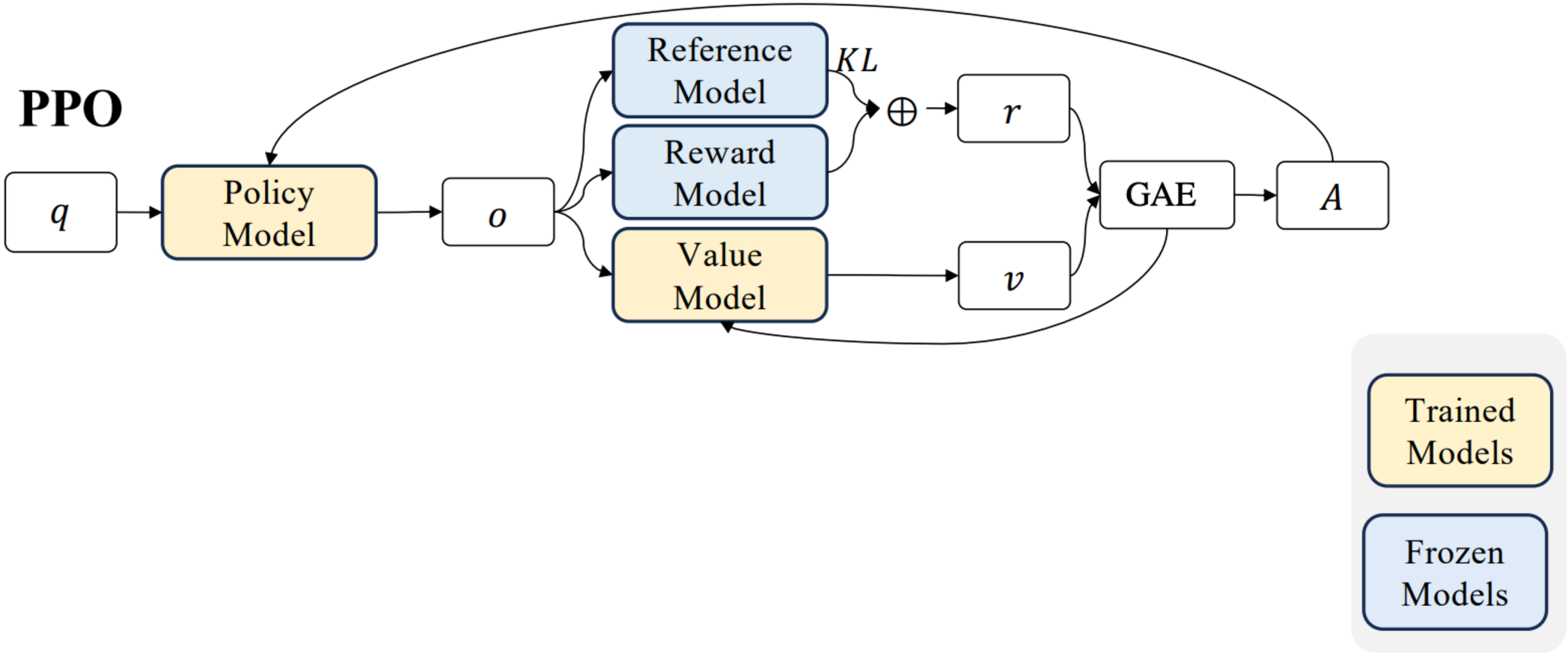
PPO for LLMs

$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{old}}(o_t|q,o_{<t})} A_t, \text{clip} \left(\frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{old}}(o_t|q,o_{<t})}, 1-\varepsilon, 1+\varepsilon \right) A_t \right]$$

- PPO (typically) improves upon REINFORCE with clipping mechanism.
- Define reward that penalizes deviation from reference model.

$$r_t = r_\varphi(q, o_{\leq t}) - \beta \log \frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{ref}(o_t|q,o_{<t})},$$

PPO for LLMs


PPO
q
Policy Model
o
Reference Model
Reward Model
Value Model
KL
$\oplus$
r
v
GAE
A
Trained Models
Frozen Models

DPO for LLMs

- Do we need RL in RLHF?

$$E_{x\sim D, y\sim\pi_\theta(y|x)}[r(x,y)] - \beta D_{\mathtt{KL}}[\pi_\theta(\cdot\,|\,x)\,||\,\pi_{\mathtt{ref}}(\cdot\,|\,x)]$$

- This constrained RL problem can be solved analytically.

$$\pi^*(y\,|\,x) = \frac{1}{Z(x)}\pi_{\mathtt{ref}}(y\,|\,x)e^{\frac{1}{\beta}r(x,y)}$$

- Enables solving for r(x,y) in terms of $\pi_{\mathtt{ref}}$ and π^*; plugging into reward learning loss function yields:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta;\pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w\mid x)}{\pi_{\text{ref}}(y_w\mid x)} - \beta\log\frac{\pi_\theta(y_l\mid x)}{\pi_{\text{ref}}(y_l\mid x)}\right)\right]$$

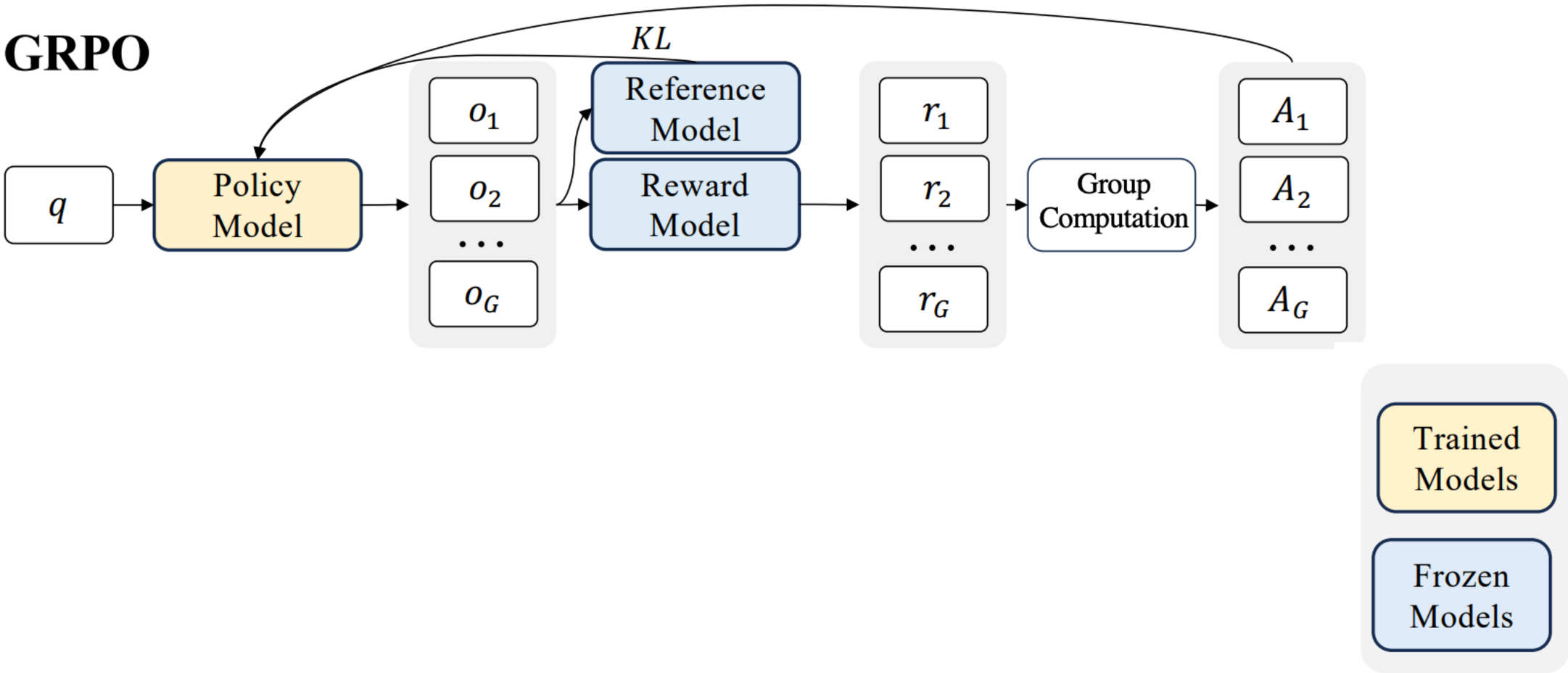
GRPO for LLMs

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

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

- GAE to compute advantage estimate is memory intensive.
- Alternatively, for each prompt, sample G responses and compute advantage as a function of these G responses.
- GRPO uses $A_{i,t} = \frac{r_i - \texttt{mean}(r)}{\texttt{std}(r)}$, i.e., group normalized Monte Carlo return.
- Similar idea to “vine” sampling method from Schulman et al. (2015).

GRPO for LLMs



Learning to “Think”

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?
A: ***Let's think step by step.***

(Output) There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓

- Chain of thought (and many variants) show that outputting reasoning like logic before an answer can improve final response.
- RL can be used to figure out what intermediate outputs will maximize the correctness of the final response.
- “Stop anthropomorphizing intermediate tokens as reasoning/thinking Traces!” Kambhampati et al. (2025).

Brennen’s Presentation

- Slides

Albert’s Presentation

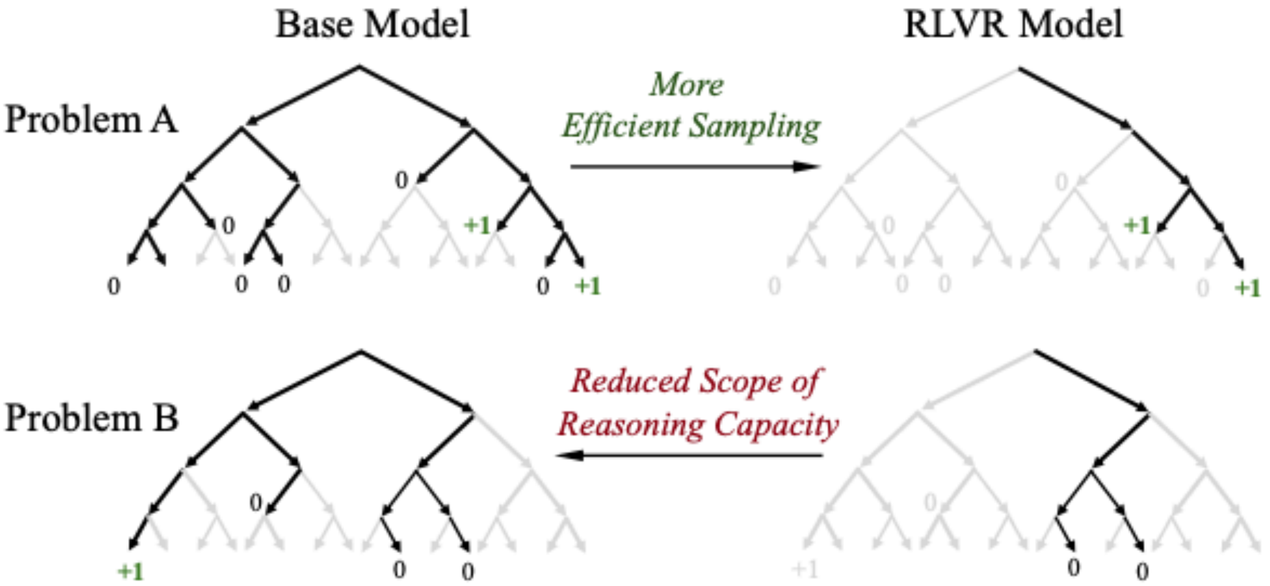
- Slides

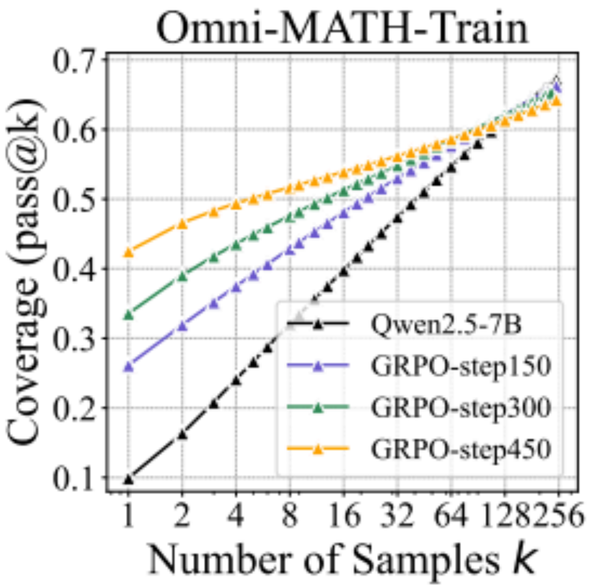
What is Missing?

- RL for LLMs has mainly focused on basic on-policy policy gradient methods and Monte Carlo returns.
- What is missing and what are challenges and opportunities?
  - Bootstrapping and temporal-difference.
  - Advanced exploration.
  - Model learning.
  - Hierarchy.

Recommend: Dale Schuurman's keynote at RLC 2025.

What can RL Learn?





Summary

- Several different RL methods available for fine-tuning LLMs.
- Can be used for RLHF or to discover “reasoning” abilities.
- Much of RL has not been explored well for LLMs (publically).