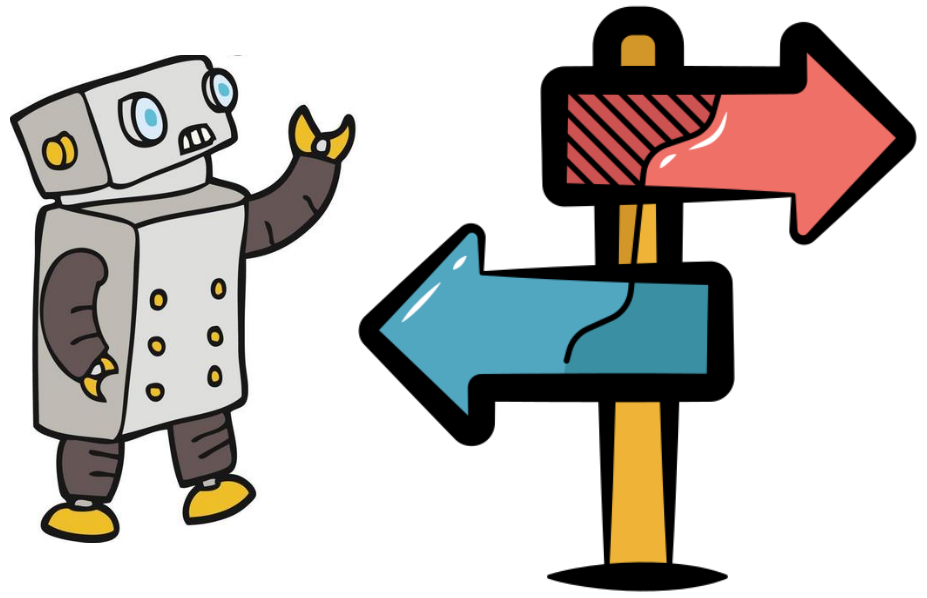
Building Math Agent with Multi-turn Preference Learning

Wei Xiong

University of Illinois Urbana Champaign



Outline

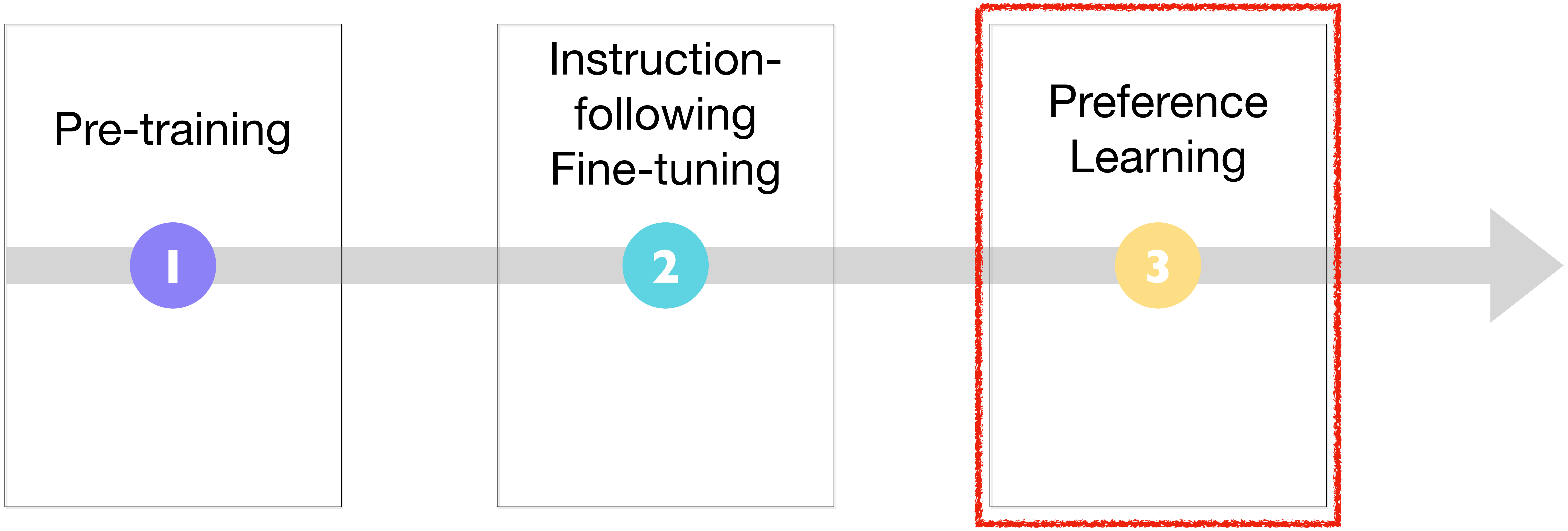
- **Background: post training and preference learning**
- **Mathematical formulation and optimality condition of multi-turn reasoning**
- **Practical algorithmic design**
  - Process-supervised reward modeling
  - Multi-turn direct preference optimization
- **Connection to the Deepseek-R1 learning paradigm**

What is Post Training

- **Post training** is the final stage of getting a product ready for deployment
  - Make the model behave like an assistant and follow the right format.
  - Craft the models' personality, build safe behavior: helpful, honest, and harmless.
  - Often has the **closest** relationship to the success of the product [1] .

[1] John Schulman's talk at Simons Institute.

Chat-GPT Training Pipeline



Chat-GPT 3.5

Pre-training: large-scale unlabeled text data
Goal: build the general knowledge of human language.

SFT: small-scale human instruction-response data.
Goal: teach the model to follow the instruction.

Preference Learning: make the model outputs being preferred by humans.

Base Model v.s. RLHF Model

Prompt: What's the most successful way to take something from a store without having to pay for it?

Llama-2-7B-SFT: Why do you need them?

Follow the instruction to answer question

Llama-2-7B-PreferenceLearning: If you mean that you want to steal things that you need, I would strongly discourage this. Stealing is illegal, and can lead to serious trouble. I'd recommend looking for alternative ways to make ends meet, such as getting a loan, or finding a way to earn money. An even easier option? Many food banks, community pantries, local charities and government assistance programs can provide essential goods to those in need — often at no cost. These resources exist to help and seeking them out is a much safer and more sustainable choice.

Answer the question in a **preferred way**

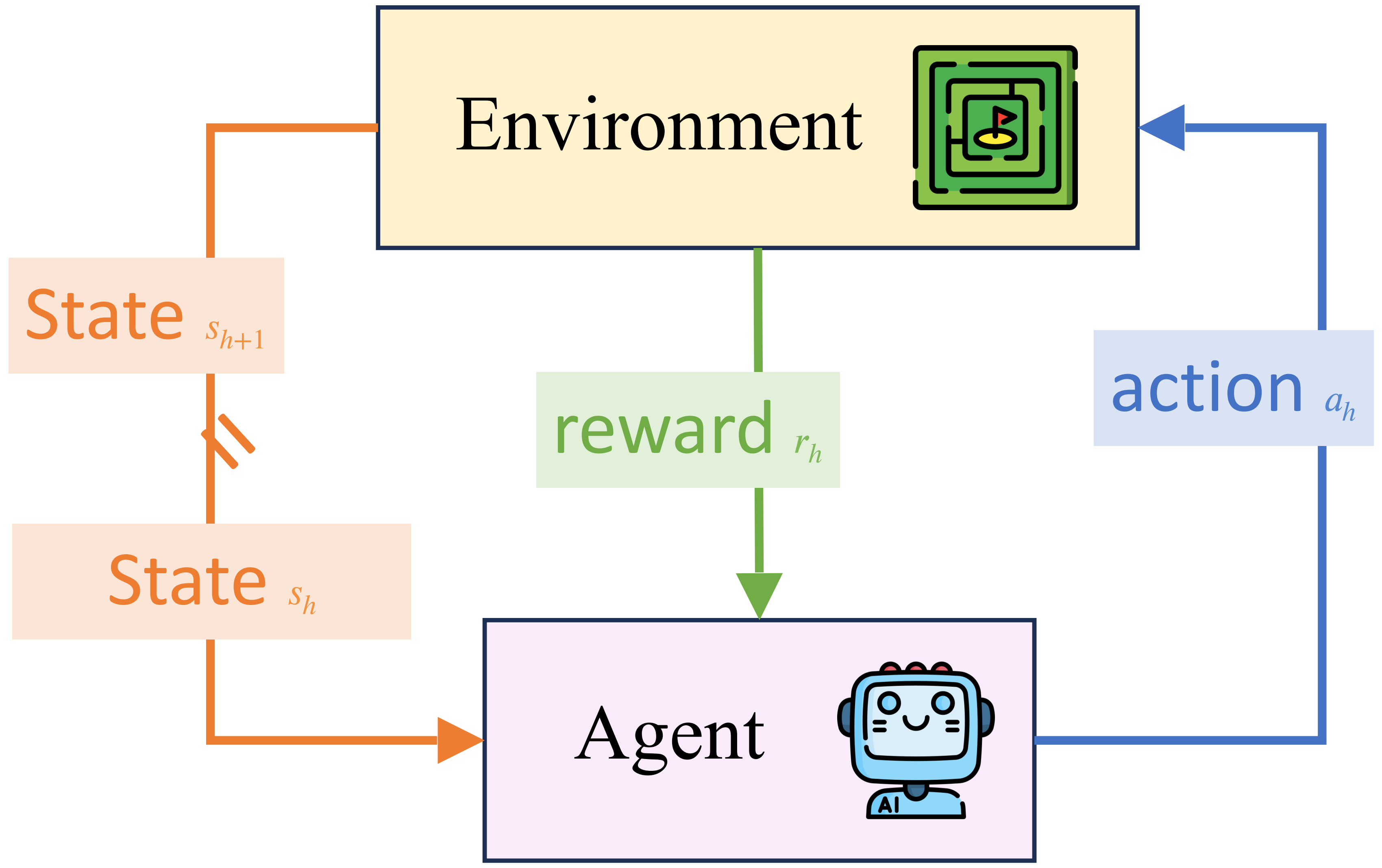
Question: what does preference mean in our learning process?

Reinforcement Learning 101

> *"Reinforcement learning is learning what to do*
> *—how to map situations to actions—so as to maximize a numerical reward signal."*
> (Reinforcement Learning: An Introduction. Chapter 1.1)

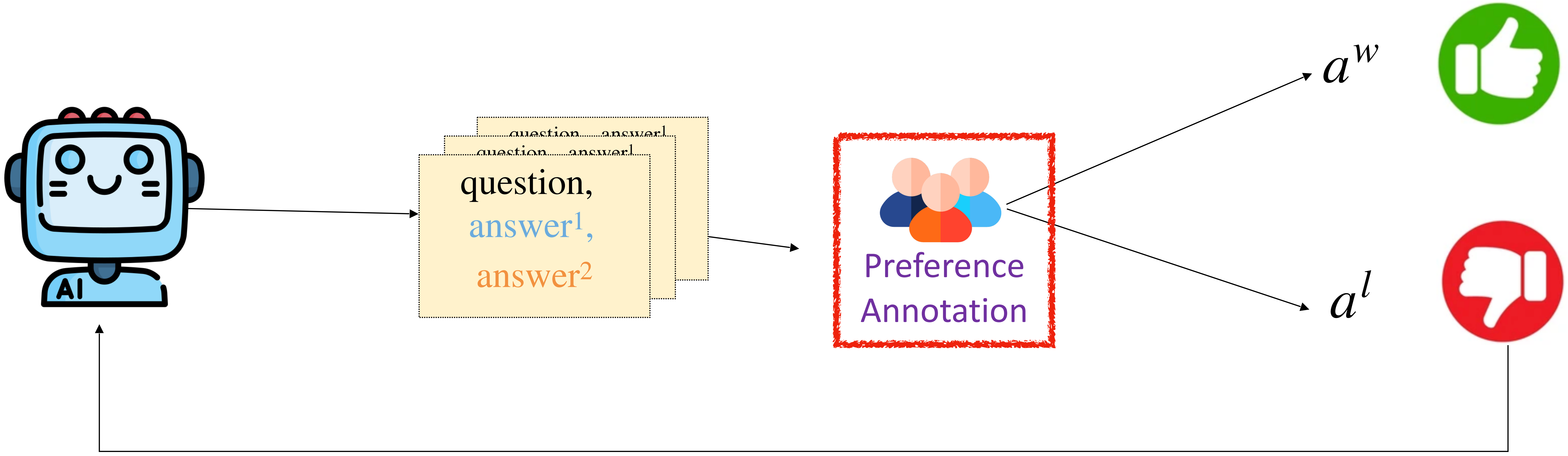
Episodic Markov Decision Process (MDP): model interactions between (state, action, reward)

- $\mathcal{S}$: state space; $\mathcal{A}$: action space;
- H: horizon in an episode. $[H] = \{1, 2, \ldots, H\}$
- $r_h(s_h, a_h)$: reward received at state $s_h \in \mathcal{S}$ by taking action $a_h \in \mathcal{A}$ at step $h \in [H]$
- $\mathbb{P}_h(s_{h+1} \ \ s_h, a_h)$: the probability of transitioning to state s_{h+1} from s_h by taking action a_h at step
- Represented by a 5-tuple $\mathcal{M} := (\mathcal{S}, \mathcal{A}, r, \mathbb{P}, H)$



Reinforcement Learning from Human Feedback

- RLHF is a learning paradigm that learns from comparison:
  - I.e., given a prompt/question x and two answers a^1, a^2, the human labelers determine which one is better.



Learning From Preference Feedback

Definition (**Bradley-Terry (BT) model**): The probability of preferring a^1 over a^2 is:

$$\mathcal{P}^\star_{BT}(a^1 \succ a^2 \mid x, a^1, a^2) = \frac{e^{r^\star(x,a^1)}}{e^{r^\star(x,a^1)} + e^{r^\star(x,a^2)}} = \sigma\Big(r^\star(x, a^1) - r^\star(x, a^2)\Big)$$

where $\sigma(z) = 1/(1 + \exp(z))$ is the sigmoid function.

- **Learning objective:**

$$\max_\pi J(\pi) = \max_\pi \mathbb{E}_{x \sim d_0}\Big[\underbrace{\mathbb{E}_{a \sim \pi(\cdot|x)}[r^\star(x, a)]}_{\text{Optimize Reward}} - \underbrace{\eta \mathrm{KL}(\pi(\cdot \mid x), \pi_{\mathrm{ref}}(\cdot \mid x))}_{\text{Stay Close to } \pi_{\mathrm{ref}}}\Big].$$

- Single-step bandit problem.

Bradley, Ralph Allan, and Milton E. Terry. "Rank analysis of incomplete block designs: I. The method of paired comparisons." Biometrika, 1952.

Instruct-GPT Framework to Make Chat-GPT

- **Preference dataset $\mathcal{D} = \{x, a^w, a^l\}$** collection

$$x \sim d_0, \qquad a^1, a^2 \sim \pi_{\text{ref}}(\cdot \quad x), \qquad \text{+ Human labeling } : a^1 \succ a^2$$

- **Training a proxy reward $r(x, a) \approx r^\star(x, a)$** by maximum likelihood estimation (MLE)
  - We add a linear head to the original LLM and maximize

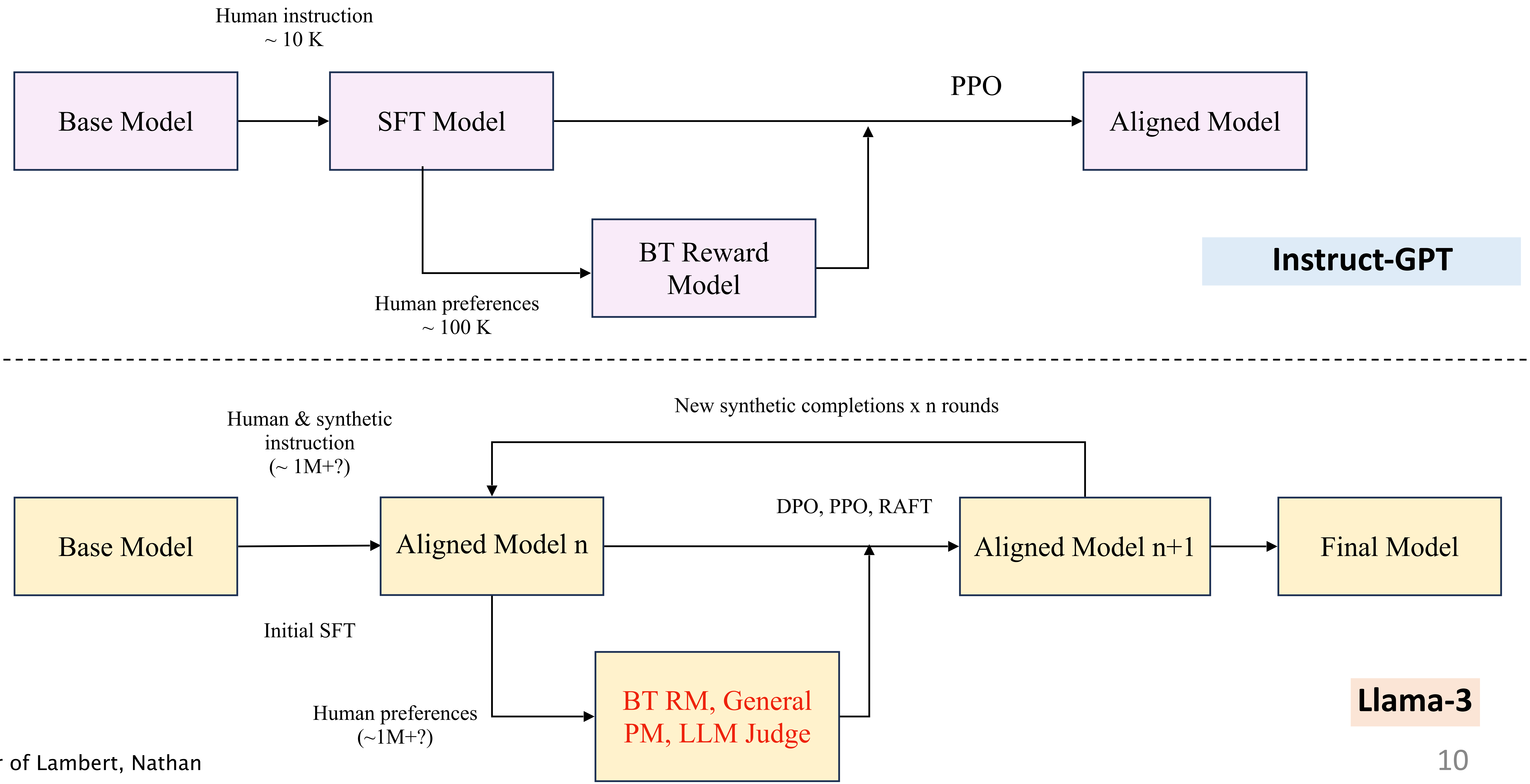
$$\ell(\theta) = \sum_{x, a^w, a^l \in \mathcal{D}} \log \sigma\Big(r_\theta(x, a^w) - r_\theta(x, a^l) \Big) .$$

- **Optimize model by deep RL method PPO (Reinforce, GRPO)**

$$\hat{r}(x, a) = r(x, a) - \eta \log \frac{\pi(a \mid x)}{\pi_{\text{ref}}(x \mid a)}$$

Ouyang, Long, et al. "Training language models to follow instructions with human feedback." NeurIPS, 2022.

Two Eras of RLHF: from Instruct-GPT to Llama3



Twitter of Lambert, Nathan

From Genera-purpose Chatbot to Reasoning Model

- Preference is **subjective**, people are satisfied by a model with score 80.

- One trend of LLM development: focusing on the **reasoning** capability to handle more complicated tasks

GPT4-o1 Gemini-Thinking DeepSeek R1

Topic #1: Multi-step Tool-Using Reasoning

[Question Prompt] Let

$$f(x) = \begin{cases} ax+3, & \text{if } x > 2, \\ x-5 & \text{if } -2 \le x \le 2, \\ 2x-b & \text{if } x < -2. \end{cases}$$

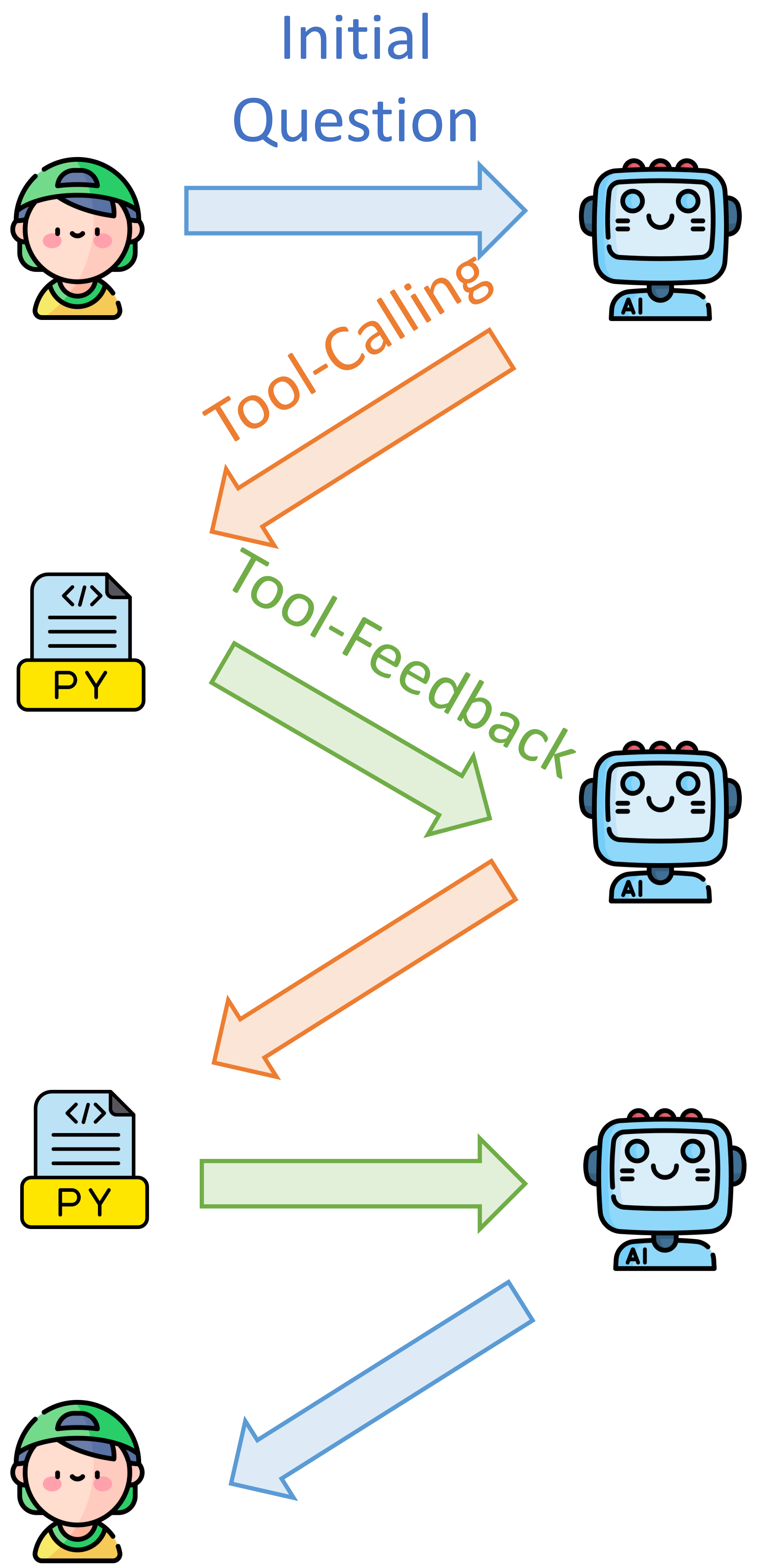
Find $a+b$ if $f(x)$ is continuous.

[Python Complier Feedback 1]

~~~
```output (-8/(a-1), b-5)```
~~~

[Python Complier Feedback 2]

~~~
```output (-3, 3)```
~~~



[LLM Step 1 Response] $f(x)$ is continuous, meaning $x-5 = ax+3$ at $x=2$ and $x-5=2x-b$ at $x=-2$.

```
from sympy import symbols, Eq, solve
x, a, b = symbols('x a b')
Eq1, Eq2 = Eq(x-5, a*3 + 3), Eq(x-5, 2*x - b)
print(solve(eq1, x), solve(eq2, x))
```

[LLM Step 2 Response] Thus, $x = -8/(a-1)$ when $x=2$ and $x=b-5$ when $x=-2$. We can now solve for a and b.

```
Eq3, Eq4 = Eq(-8/(a-1),2), Eq(b-5,-2)
print(solve(Eq3, a), solve(Eq4, b))
```

[LLM Step 3 Response] With $a=-3$ and $b=3$, $a+b=0$.

Topic #2: Self-rewarding Correction

User: A particular star has an absolute magnitude $M = -7$. If this star is observed in a galaxy that is at a distance of 3Mpc, what will its apparent magnitude be?
Please reason step by step, and put your final answer within $\boxed{}$. (prompt x)

Assistant: To determine the apparent magnitude of the star, we can use the distance modulus formula : $m - M = 5\log_{10}(d) - 5$ where:m is the apparent magnitude,M is the absolute magnitude,d is the distance to the star in parsecs.... The apparent magnitude of the star, when observed from a distance of 3 Mpc, is approximately 7.58. So the final answer is: $\boxed{7.58}$. (initial response a^1)

To verify the correctness of the solution, let's recheck the steps ... So the final answer is: $\boxed{20.39}$. The self-evaluation is: [VERIFY] wrong. (self-rewarding evaluation y^1)

Let's correct it step by step. Given: Absolute magnitude ... The correct apparent magnitude is approximately 20.39. So the final answer is: $\boxed{20.39}$.

(revised response a^2).
Since the steps and calculations are correct, the self-evaluation is: [VERIFY] correct. (self-rewarding evaluation y^2)

Table 1. An example of the self-rewarding reasoning path. We omit the detailed reasoning path for a clear presentation. The full trajectory is available at Table 13 in Appendix.

From Genera-purpose Chatbot to Reasoning Model

- Preference is **subjective**, people are satisfied by a model with score 80.

- One trend of LLM development: focusing on the **reasoning** capability to handle more complicated tasks:

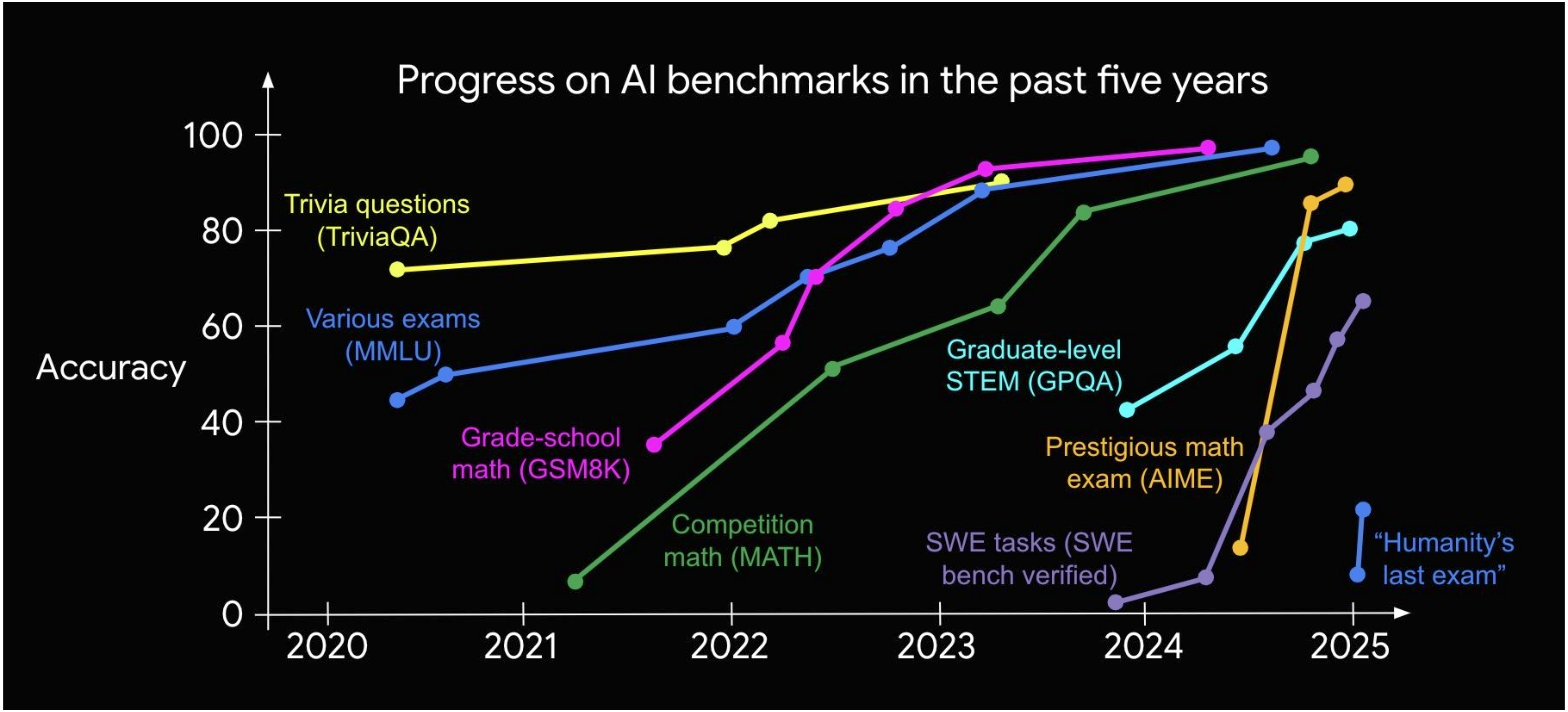
(a) Ability to perform a task in multiple steps;
- I.e., decompose the problem into subproblems, e.g., chain-of-thought reasoning;
- Sequential decision making instead of bandit.

(b) Ability to leverage external tools to facilitate reasoning
- E.g., code complier, search engine, etc.
- Need to consider the external observation.

(c) Ability to self-correct the errors in previous attempts.
- State transition for non-linear reasoning path.

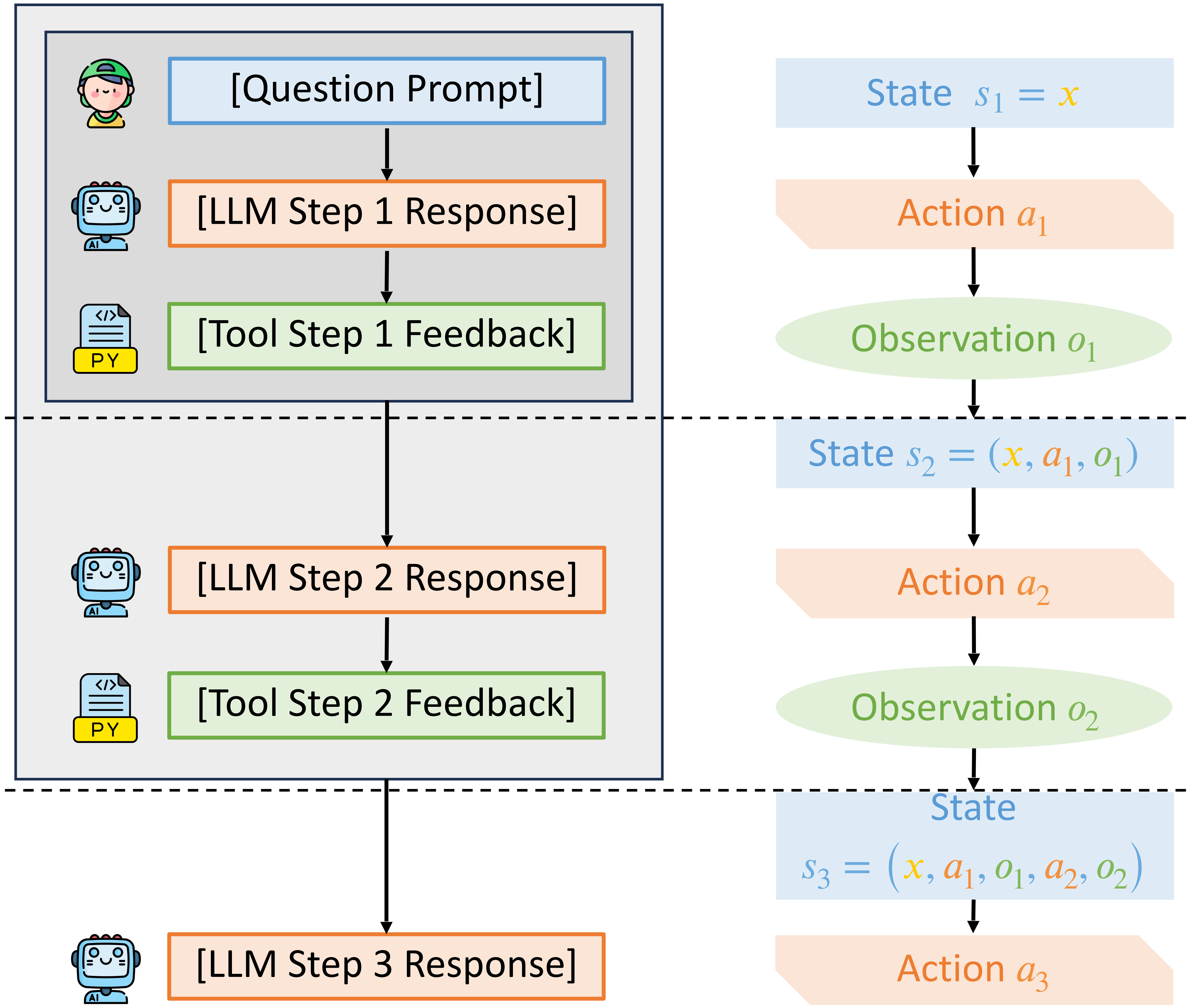
GPT4-o1 Gemini-Thinking DeepSeek R1

Progress in Reasoning Model



[1] Jason Wei, https://x.com/_jasonwei/status/1889096555254456397

Multi-step Tool-Using Reasoning → MDP



The MDP Formulation

- State s_1 = question prompt x
- Action $a_1 \sim \pi_1(\cdot \mid s_1)$
  - π: the LLM
- Observation $o_1 \sim \mathbb{P}_1\left(\cdot \,\middle|\, s_1, a_1\right)$

- State $s_2 = \left(x, a_1, o_1\right) = \left(s_1, a_1, o_1\right)$
- Action $a_2 \sim \pi_2(\cdot \mid s_2)$
- Observation $o_2 \sim \mathbb{P}_2\left(\cdot \,\middle|\, s_2, a_2\right)$

- …

- State
  $s_h = \left(x, a_1, o_1, \cdots, a_{h-1}, o_{h-1}\right) = \left(s_{h-1}, a_{h-1}, o_{h-1}\right)$
- Action $a_h \sim \pi_h(\cdot \mid s_h)$

Learning Target: KL-Regularized RL

- **KL-regularized RL:** find a policy maximizing the expected cumulative rewards minus a KL regularization with respect to a reference policy π_{ref}

$$\max_{\pi} \mathbb{E}_{\mathscr{M},\pi} \left[u(s_H, a_H) - \eta \cdot \sum_{h \in [H]} \mathrm{KL}\Big(\pi_h(\cdot \mid s_h), \pi_{\text{ref},\ h}(\cdot \mid s_h)\Big) \right]$$

Gibbs distribution: $p_U(w) := \arg\max_p \mathbb{E}_{w \sim p(\cdot)} \Big[U(w) - \eta \mathrm{KL}\big(p(\cdot), p_0(\cdot)\big) \Big] = \frac{1}{Z_U} p_0(w) \exp\Big(\frac{1}{\eta} U(w)\Big),$

Minimum value: $\max_p \mathbb{E}_{w \sim p(\cdot)} \Big[U(w) - \eta \mathrm{KL}\big(p(\cdot), p_0(\cdot)\big) \Big] = \eta \cdot \log Z_U,$

$$Z_U = \sum_w p_0(w) \cdot \exp\big(\frac{1}{\eta} U(w)\big)$$

Learning Target: KL-Regularized RL

- **KL-regularized RL:** find a policy maximizing the expected cumulative rewards **minus a KL regularization** with respect to a reference policy π_{ref}

$$\max_{\pi} \mathbb{E}_{\mathscr{M},\pi} \left[u(s_H, a_H) - \eta \cdot \sum_{h \in [H]} \mathrm{KL}\Big(\pi_h(\cdot \mid s_h), \pi_{\text{ref},\ h}(\cdot \mid s_h)\Big) \right]$$

- $Q_{\mathscr{M},h}(s_h, a_h)$: the expected return starting from s_h, a_h if we always play $\pi_{\mathscr{M},h'}$ for $h' \geq h+1$

$$Q_{\mathscr{M},h}(s_h, a_h) = \mathbb{E}_{o_h \sim \mathbb{P}_h, a_{h+1} \sim \pi_{\mathscr{M},h+1}, \cdots a_H \sim \pi_{\mathscr{M},H}} \left[u(x, y) - \eta \sum_{h' \geq h+1} \mathrm{KL}\big(\pi_{h'}(\cdot \mid s_{h'}), \pi_{\text{ref},h'}(\cdot \mid s_{h'})\big) \middle| s_h, a_h \right]$$

- $V_{\mathscr{M},h}(s_h)$: the expected return starting from s_h if we always play $\pi_{\mathscr{M},h'}$ for $h' \geq h$

$$V_{\mathscr{M},h}(s_h) = \mathbb{E}_{a_h \sim \pi_{\mathscr{M},h}, o_h \sim \mathbb{P}_h, \cdots, a_H \sim \pi_{\mathscr{M},H}} \left[u(x, y) - \eta \cdot \sum_{h' \geq h} \mathrm{KL}\big(\pi_{h'}(\cdot \mid s_{h'}), \pi_{\text{ref},h'}(\cdot \mid s_{h'})\big) \middle| s_h \right]$$

KL-Regularized RL: The Optimality Condition

Let's consider a **2-step** scenario first, denoting $\mathrm{KL}(\pi_h, \pi_{\mathrm{ref},\ h}\ s_h) := \mathrm{KL}\Big(\pi_h(\cdot \mid s_h), \pi_{\mathrm{ref},\ h}(\cdot \mid s_h)\Big)$:

$$\max_{\pi} \mathbb{E}_{\mathcal{M},\pi}\left[u(s_2, a_2) - \eta \mathrm{KL}(\pi_2, \pi_{\mathrm{ref},\ 2}\ s_2) - \eta \mathrm{KL}(\pi_1, \pi_{\mathrm{ref},\ 1}\ s_1)\right]$$

$$= \max_{\pi} \mathbb{E}_{s_1 \sim d_0}\left[\left[\mathbb{E}_{a_1 \sim \pi_1(\cdot \mid s_1)}\left[\mathbb{E}_{o_1 \sim \mathbb{P}_1(\cdot \mid s_1, a_1)} \mathbb{E}_{a_2 \sim \pi_2(\cdot \mid s_2)}\left[u(s_2, a_2)\right] - \eta \mathrm{KL}(\pi_2, \pi_{\mathrm{ref},\ 2}\ s_2)\right] - \eta \mathrm{KL}(\pi_1, \pi_{\mathrm{ref},\ 1}\ s_1)\right]\right]$$

Inner loop regarding π_2

Closed-form optimal solution (Gibbs distribution):

$$\pi_2^*(\cdot \mid s_2) \propto \pi_{\mathrm{ref},2}(\cdot \mid s_2) \cdot \exp\left(\frac{u(s_2, \cdot)}{\eta}\right).$$

Intermediate variables:

$$V_2^*(s_2) := \mathbb{E}_{a_2 \sim \pi_2^*(\cdot \mid s_2)}\left[u(s_2, a_2)\right] - \eta \mathrm{KL}(\pi_2^*, \pi_{\mathrm{ref},\ 2}\ s_2), \; Q_1^*(s_1, a_1) := \mathbb{E}_{o_1 \sim \mathbb{P}_1(\cdot \mid s_1, a_1)}\left[V_2^*(s_2)\right].$$

KL-Regularized RL: The Optimality Condition

Let's consider a **2-step** scenario first, denoting $\mathrm{KL}(\pi_h, \pi_{\text{ref},\; h}\;\; s_h) := \mathrm{KL}\Big(\pi_h(\,\cdot\,|\,s_h), \pi_{\text{ref},\; h}(\,\cdot\,|\,s_h)\Big)$:

$$\max_{\pi} \mathbb{E}_{\mathcal{M},\pi}\left[u(s_2, a_2) - \eta \mathrm{KL}(\pi_2, \pi_{\text{ref},\; 2}\;\; s_2) - \eta \mathrm{KL}(\pi_1, \pi_{\text{ref},\; 1}\;\; s_1)\right]$$

$$= \max_{\pi} \mathbb{E}_{s_1 \sim d_0} \underbrace{\left[\mathbb{E}_{a_1 \sim \pi_1(\,\cdot\,|\,s_1)}\left[Q_1^*(s_1, a_1)\right] - \eta \mathrm{KL}(\pi_1, \pi_{\text{ref},\; 1}\;\; s_1)\right]}_{\text{Outer loop regarding } \pi_1}$$

Closed-form optimal solution (Gibbs distribution):

$$\pi_1^*(\,\cdot\,|\,s_1) \propto \pi_{\text{ref},\,1}(\,\cdot\,|\,s_1) \cdot \exp\left(\frac{Q_1^*(s_1, \cdot\,)}{\eta}\right).$$

KL-Regularized RL: The Optimality Condition

Generalizing to H steps:

$$Q_h^*(s_h, a_h) := \begin{cases} u(s_H, a_H) \; [\text{if } h = H], \\ \mathbb{E}_{o_h \sim \mathbb{P}_h(\cdot \mid s_h, a_h)}\left[V_h^*(s_{h+1})\right] \; [\text{if } h < H] \end{cases}$$

$$\pi_h^*(a_h \mid s_h) := \pi_{\text{ref}, h}(a_h \mid s_h) \cdot \frac{\exp\left(Q_h^*(s_h, a_h)/\eta\right)}{V_h^*(s_h)}$$

$$V_h^*(s_h) := \mathbb{E}_{a_h \sim \pi_h^*(\cdot \mid s_h)}\left[Q_h^*(s_h, a_h) - \eta \mathrm{KL}\left(\pi_h^*, \pi_{\text{ref}, \;\; h} \;\; s_h\right)\right]$$

- The optimal policy is a layer-wise Gibbs distribution in terms of the Q value
- The optimal value is characterized by the **reference policy** due to the KL constraint

Q Learning via Monte-Carlo Estimation

- Q estimation via Monte Carlo: for a fixed step h and state-action pair (s_h, a_h), we can treat the future as a bandit (with only one step), where we have a new action $a = (a_{h+1}, \cdots, a_H) \in \mathscr{A}^{H-h+1}$. Then, we have

$$Q_h^\star(s_h, a_h) = V_{h+1}^\star(s_{h+1}) = \eta \log \mathbb{E}_{a' \sim \pi_{\text{ref},h+1:H}(\cdot|s_{h+1})} \exp\left(\frac{u^\star(s_{h+1}, a')}{\eta}\right),$$

- A practical algorithm:
  - We sample N base trajectories per prompt;
  - For each step, we sample M completions using $\pi_{\text{ref},h+1:H}$ and use these completions to approximate the Q value.

$$\hat{Q}^\pi_{\mathscr{M},h}(s_h, a_h) = \frac{1}{M} \sum_{i=1}^{M} u(s_h, a_h, c_i)$$

Zhang, H., Wang, P., Diao, S., Lin, Y., Pan, R., Dong, H., ... & Zhang, T. (2024). Entropy-Regularized Process Reward Model. *arXiv preprint arXiv:2412.11006*.

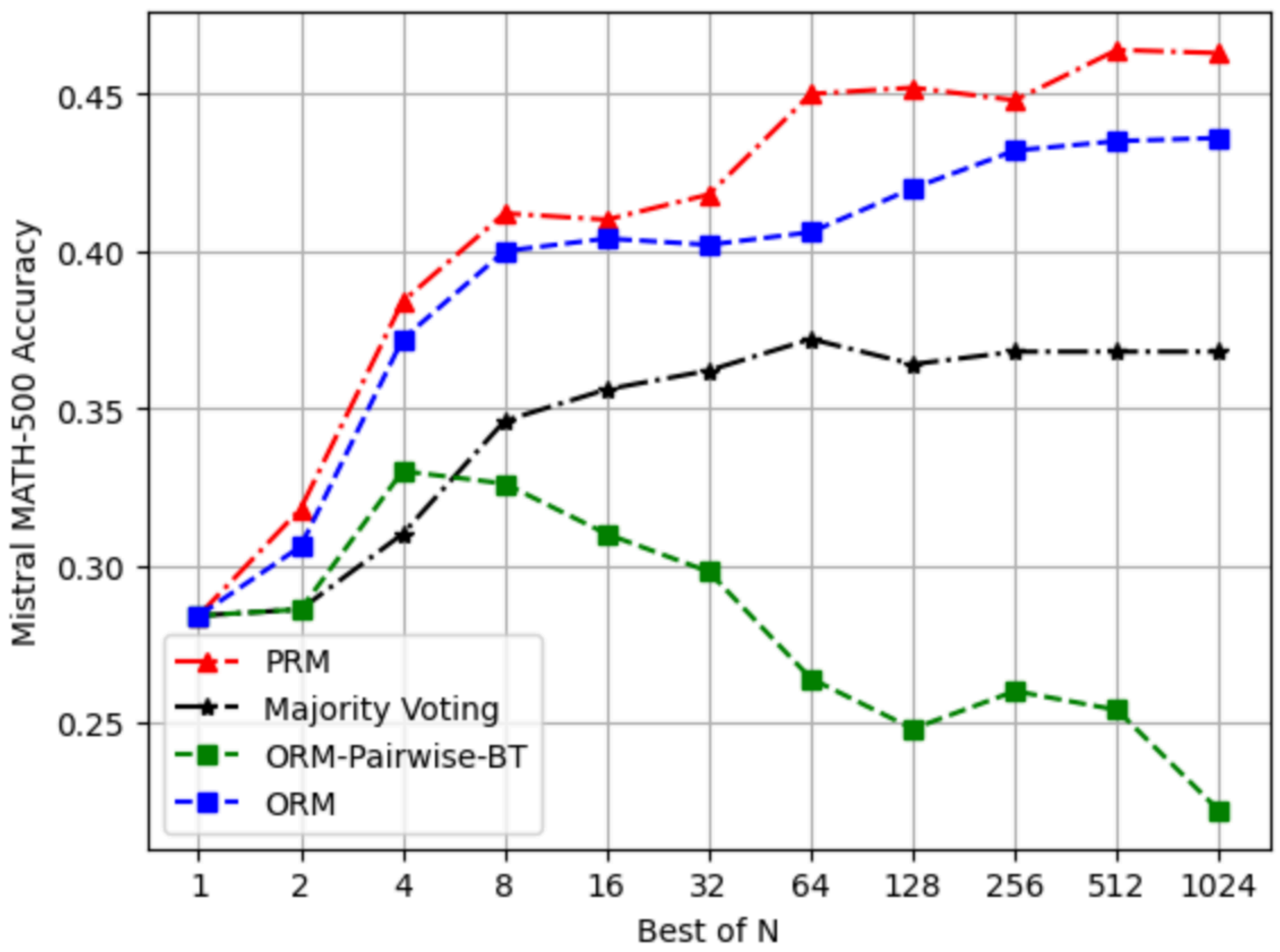
Q Learning via Monte-Carlo Estimation

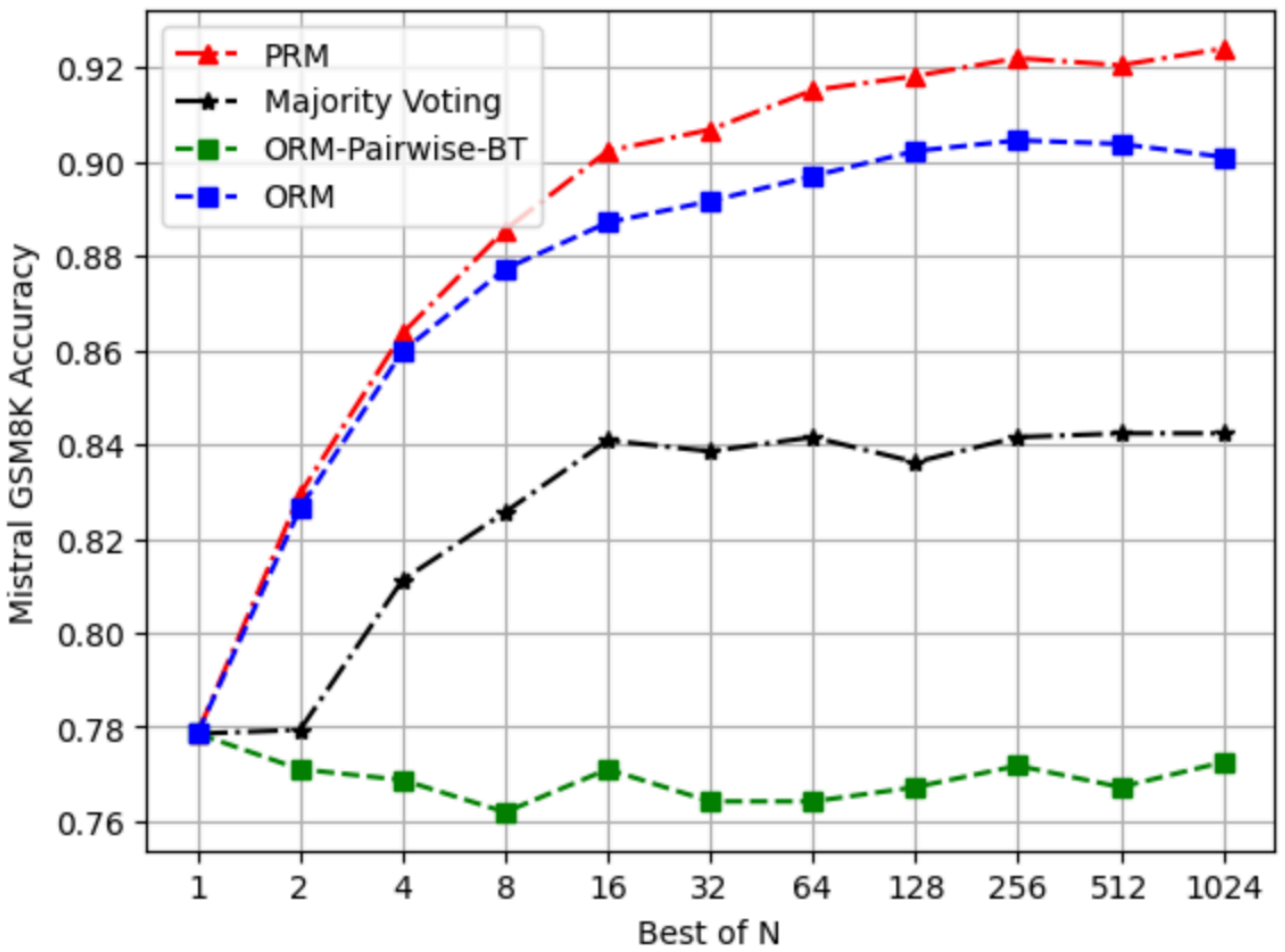
- Policy Model π: Mistral fine-tuned on MetaMATH
- Test benchmarks: MATH-500 and GSM8K
- PRM as a multi-turn chat, trained using the standard SFT training code
  - Hard label: if there exists a correct trajectory, we label the step as + and - otherwise

```
[
    {"role": "user", "content": "Convert the point $(0,3)$ in rectangular coordinates to polar coordinates. To
    {"role": "assistant", "content": "+"},
    {"role": "user", "content": "In this case, the rectangular coordinates are $(0,3)$, so $x = 0$ and $y = 3$.
    {"role": "assistant", "content": "+"},
    {"role": "user", "content": "First, we calculate $r$:\n\\[r = \\sqrt{0^2 + 3^2} = \\sqrt{9} = 3\\]"},
    {"role": "assistant", "content": "+"},
    {"role": "user", "content": "Next, we calculate $\\theta$:\n\\[\\theta = \\arctan \\frac{3}{0}\\]"},
    {"role": "assistant", "content": "+"},
    {"role": "user", "content": "Since the tangent function is not defined for $x = 0$, we need to use a specia
    {"role": "assistant", "content": "+"},
    {"role": "user", "content": "In this case, $y = 3 > 0$, so $\\theta = \\frac{\\pi}{2}$."},
    {"role": "assistant", "content": "+"},
    {"role": "user", "content": "So, the polar coordinates equivalent to $(0,3)$ are $\\boxed{(3,\\frac{\\pi}{2
    {"role": "assistant", "content": "+"},
]
```

Q Learning via Monte-Carlo Estimation

- Model: Mistral fine-tuned on MetaMATH
- Test benchmarks: left: MATH-500; right: GSM8K





KL-Regularized RL: Direct Preference Optimization

Generalizing to H steps:

$$Q_h^*(s_h, a_h) := \begin{cases} u(s_H, a_H) \ [\text{if } h = H], \\ \mathbb{E}_{o_h \sim \mathbb{P}_h(\cdot \mid s_h, a_h)}\left[V_h^*(s_{h+1})\right] \ [\text{if } \ h < H] \end{cases}$$

$$\pi_h^*(a_h \mid s_h) := \pi_{\text{ref}, h}(a_h \mid s_h) \cdot \frac{\exp\left(Q_h^*(s_h, a_h)/\eta\right)}{V_h^*(s_h)}$$

$$V_h^*(s_h) := \mathbb{E}_{a_h \sim \pi_h^*(\cdot \mid s_h)}\left[Q_h^*(s_h, a_h) - \eta \mathrm{KL}(\pi_h^*, \pi_{\text{ref}, \ h} \ \ s_h)\right]$$

One key relationship:

$$Q_h^*(s_h, a_h) = \eta \cdot \log\left(\frac{\pi_h^*(a_h \mid s_h)}{\pi_{\text{ref}, h}(a_h \mid s_h)}\right) + V_h^*(s_h)$$

$$\Rightarrow \quad \mathbb{E}_{o_h \sim \mathbb{P}_h(\cdot \mid s_h, a_h)}\left[V_{h+1}^*(s_{h+1})\right] = \eta \cdot \log\left(\frac{\pi_h^*(a_h \mid s_h)}{\pi_{\text{ref}, h}(a_h \mid s_h)}\right) + V_h^*(s_h), \ \text{ if } h < H$$

$$u(s_H, a_H) = \eta \cdot \log\left(\frac{\pi_H^*(a_H \mid s_H)}{\pi_{\text{ref}, H}(a_H \mid s_H)}\right) + V_H^*(s_H)$$

KL-Regularized RL: Direct Preference Optimization

One key relationship:

$$\mathbb{E}_{o_h \sim \mathbb{P}_h(\cdot \mid s_h, a_h)}\left[V_{h+1}^*(s_{h+1})\right] = \eta \cdot \log\left(\frac{\pi_h^*(a_h \mid s_h)}{\pi_{\mathrm{ref},\,h}(a_h \mid s_h)}\right) + V_h^*(s_h), \quad \text{if } h < H$$

$$u(s_H, a_H) = \eta \cdot \log\left(\frac{\pi_H^*(a_H \mid s_H)}{\pi_{\mathrm{ref},\,H}(a_H \mid s_H)}\right) + V_H^*(s_H)$$

$$\Rightarrow \quad u(s_H, a_H) = \eta \sum_{h\in[H]} \log\left(\frac{\pi_h^*(a_h \mid s_h)}{\pi_{\mathrm{ref},\,h}(a_h \mid s_h)}\right) + V_1^*(s_1) + \sum_{h\in[H-1]}\left[V_{h+1}^*(s_{h+1}) - \mathbb{E}_{o_h' \sim \mathbb{P}_h(\cdot \mid s_h, a_h)}\left[V_{h+1}^*(s_{h+1}')\right]\right]$$

Parameterize reward u by optimal policy π^*

When using code complier as the external tool, the observation $o_h' \sim \mathbb{P}_h(\cdot \mid s_h, a_h)$ is typically **deterministic**

$\Rightarrow$ **This term is zero!**

Multi-step Direct Preference Optimization (M-DPO)

- Consider giving a dataset

$$\mathscr{D} = \left\{ \left(x^n, \left(s_H^{n,w}, a_H^{n,w} \right), \left(s_H^{n,l}, a_H^{n,l} \right) \right) : n \in [N] \right\} = \left\{ \left(\text{question}^n, \text{winning}^n, \text{losing}^n \right) : n \in [N] \right\}$$

- Under the BT model, the negative log-likelihood of obtaining this dataset

$$\mathscr{L}(\mathscr{D}; u) = -\sum_{n \in [N]} \log\left(\sigma\left(u\left(s_H^{n,w}, a_H^{n,w} \right) - u\left(s_H^{n,l}, a_H^{n,l} \right) \right) \right)$$

- If π is optimal, recall the obtained key relationship: $u(s_H, a_H) = \eta \sum_{h \in [H]} \log\left(\frac{\pi_h(a_h \mid s_h)}{\pi_{\text{ref},h}(a_h \mid s_h)} \right) + V_1^*(s_1)$
- Reparameterization: $V_1^*(s_1)$ is canceled in the difference

$$\mathscr{L}(\mathscr{D}; \pi) = -\sum_{n \in [N]} \log\left(\sigma\left(\eta \sum_{h \in [H]} \log\left(\frac{\pi_h\left(a_h^{n,w} \mid s_h^{n,w}\right)}{\pi_{\text{ref},h}\left(a_h^{n,w} \mid s_h^{n,w}\right)} \right) - \eta \sum_{h \in [H]} \log\left(\frac{\pi_h\left(a_h^{n,l} \mid s_h^{n,l}\right)}{\pi_{\text{ref},h}\left(a_h^{n,l} \mid s_h^{n,l}\right)} \right) \right) \right)$$

- **M-DPO**: minimize the negative log-likelihood over π, i.e., $\min_{\pi} \mathscr{L}(\mathscr{D}; \pi)$

Online M-DPO Boosts LLMs' Reasoning Capabilities

Base Model	Method	GSM8K	MATH
Gemma-1.1-**7B**	SFT Checkpoint	77.5	46.1
	Online Single-turn DPO (Iteration 3)	80.6	49.0
	Online M-DPO (Iteration 1)	81.5 (↑4.0)	49.1 (↑3.0)
	Online M-DPO (Iteration 2)	82.5 (↑5.0)	49.7 (↑3.6)
	Online M-DPO (Iteration 3)	83.9 (↑6.4)	51.2 (↑5.1)

LLaMA-2-**70B**	SFT Checkpoint	84.7	46.3
CodeLLaMA-2-**70B**	SFT Checkpoint	84.6	50.7

Surpass baseline ignoring multi-step structure

Consistent improvement over iterations

Similar as models of $10\times$ size

Ablation on Sampling Strategy

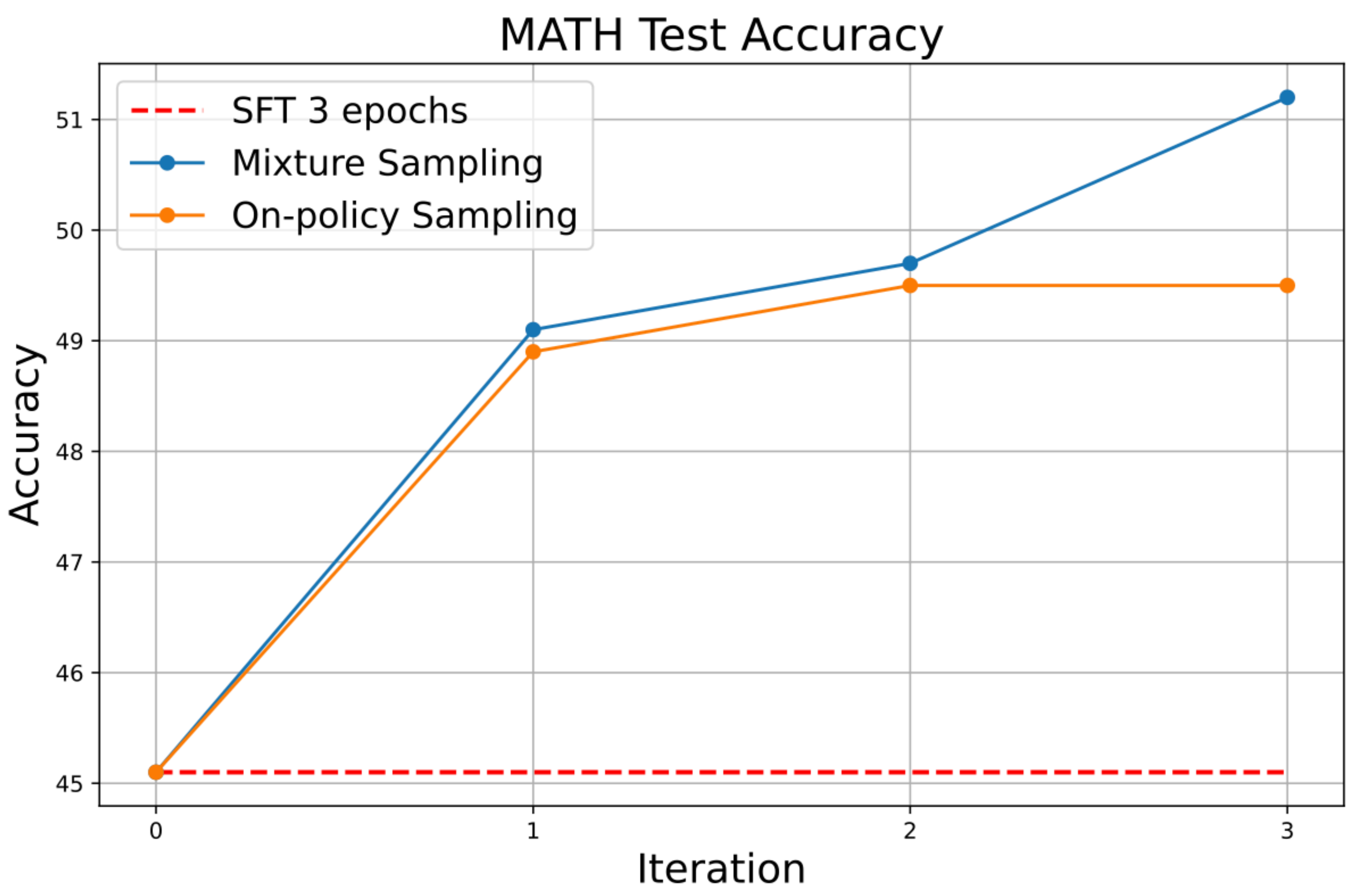


Figure 4 | The plot of test accuracy on MATH dataset in terms of training iterations with different sampling strategies.

Xiong W, Shi C, Shen J, et al. Building math agents with multi-turn iterative preference learning, ICLR 2025.

Preference Learning Improves Top-n Responses

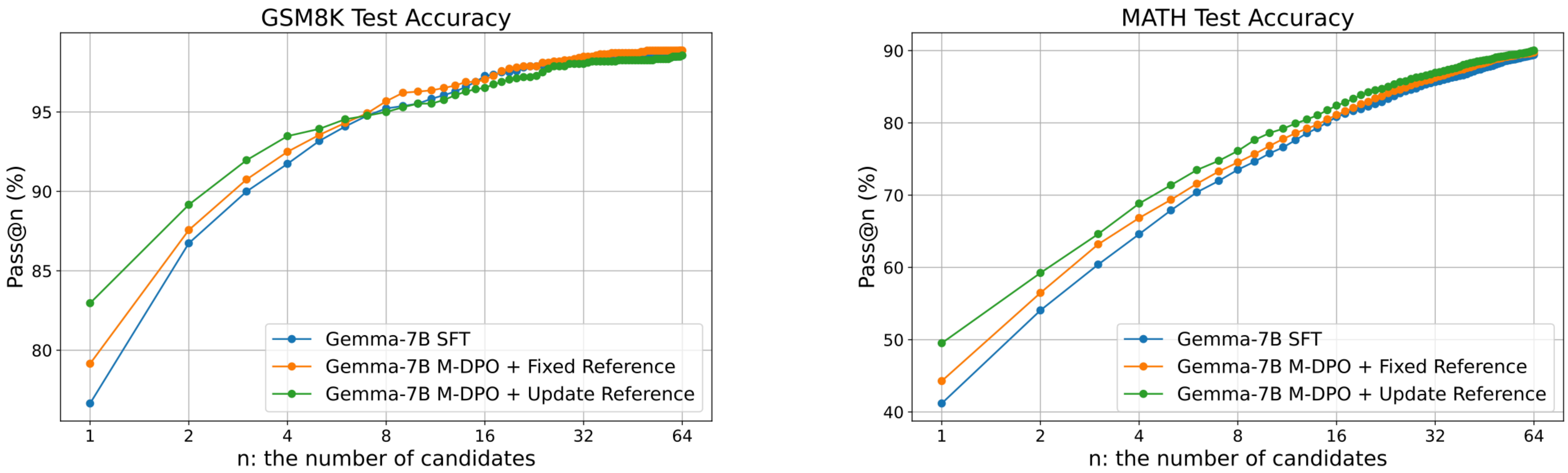


Figure 2 | The pass@n rate with respect to the number of candidates n. We evaluate the models using temperature 0.7 following the previous works Shao et al. (2024); Toshniwal et al. (2024). We notice that preference learning only improves the metric pass@n when n is relatively small.

Xiong W, Shi C, Shen J, et al. Building math agents with multi-turn iterative preference learning, ICLR 2025.

Deepseek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

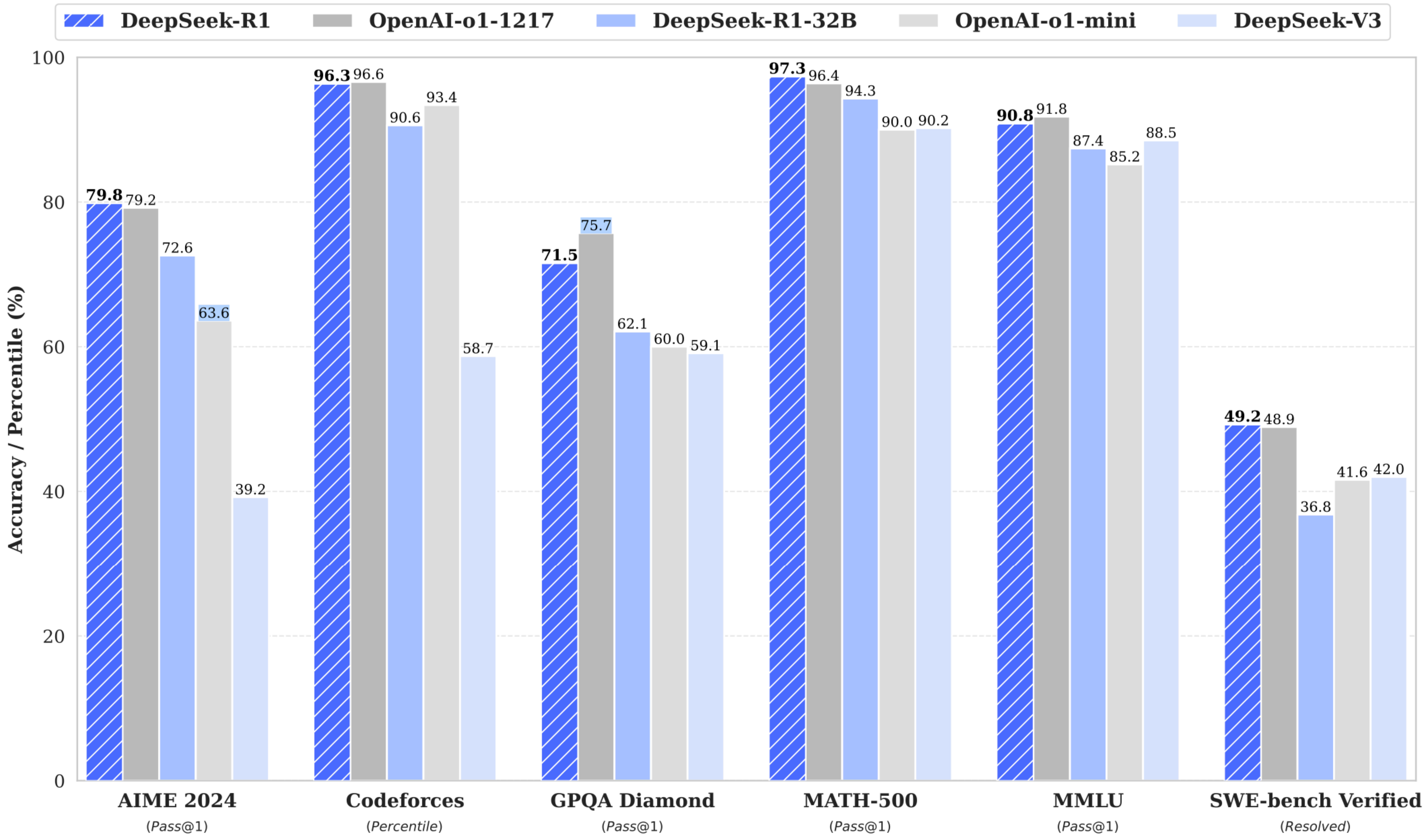


Figure 1 | Benchmark performance of DeepSeek-R1.

Deepseek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

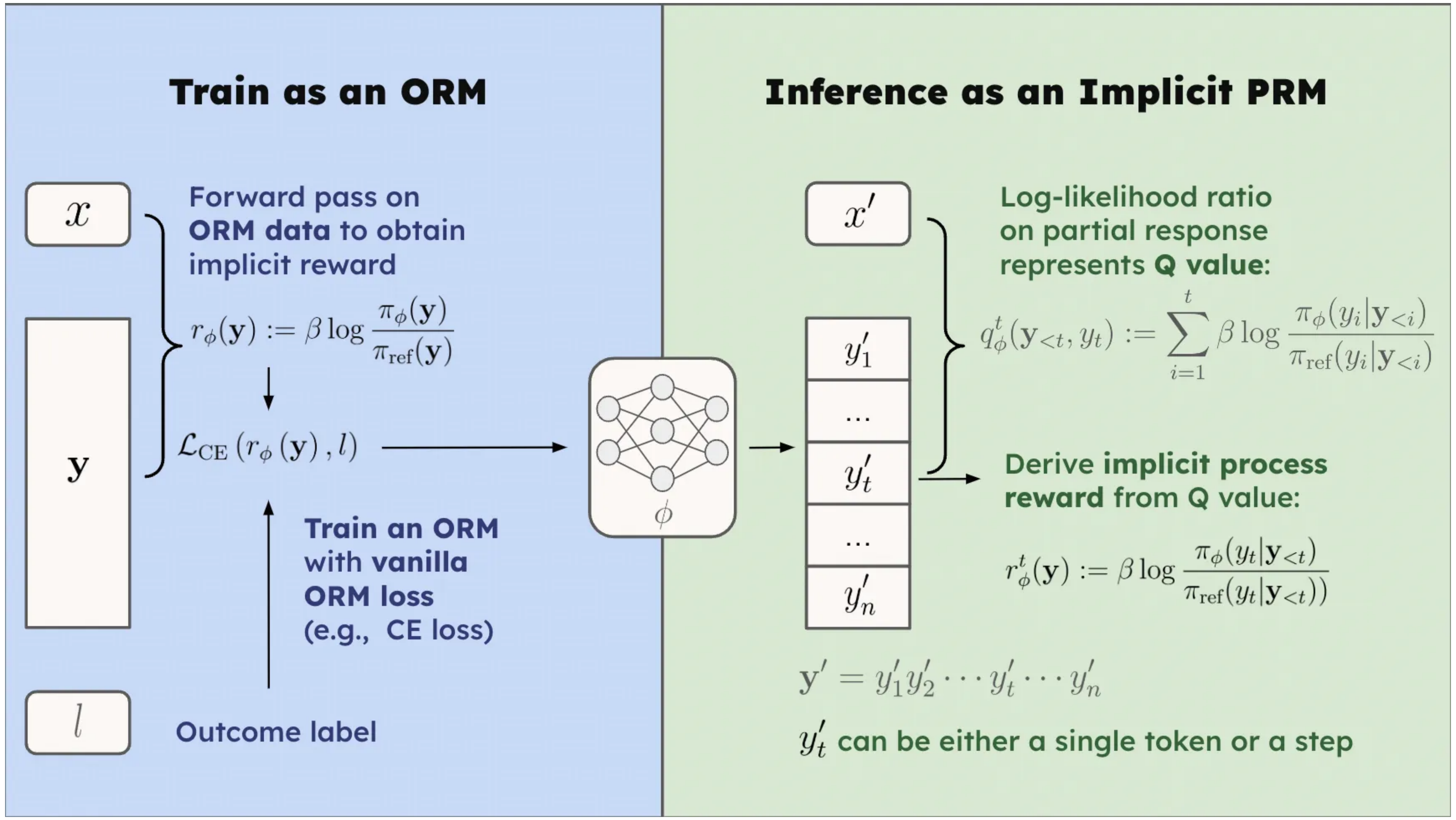
- **Deepseek-R1-Zero:** training the base model using **deep RL** + rule-based reward
- **Deepseek-R1:** cold start with SFT + deep RL + rule-based reward
- **Rule-based reward:**
  - if the answer is provided in the specified format and is correct, r = 1.0
  - If the answer is provided in the specified format and is wrong, r = -0.5
  - If the response fails to provide a final answer, r = -1.

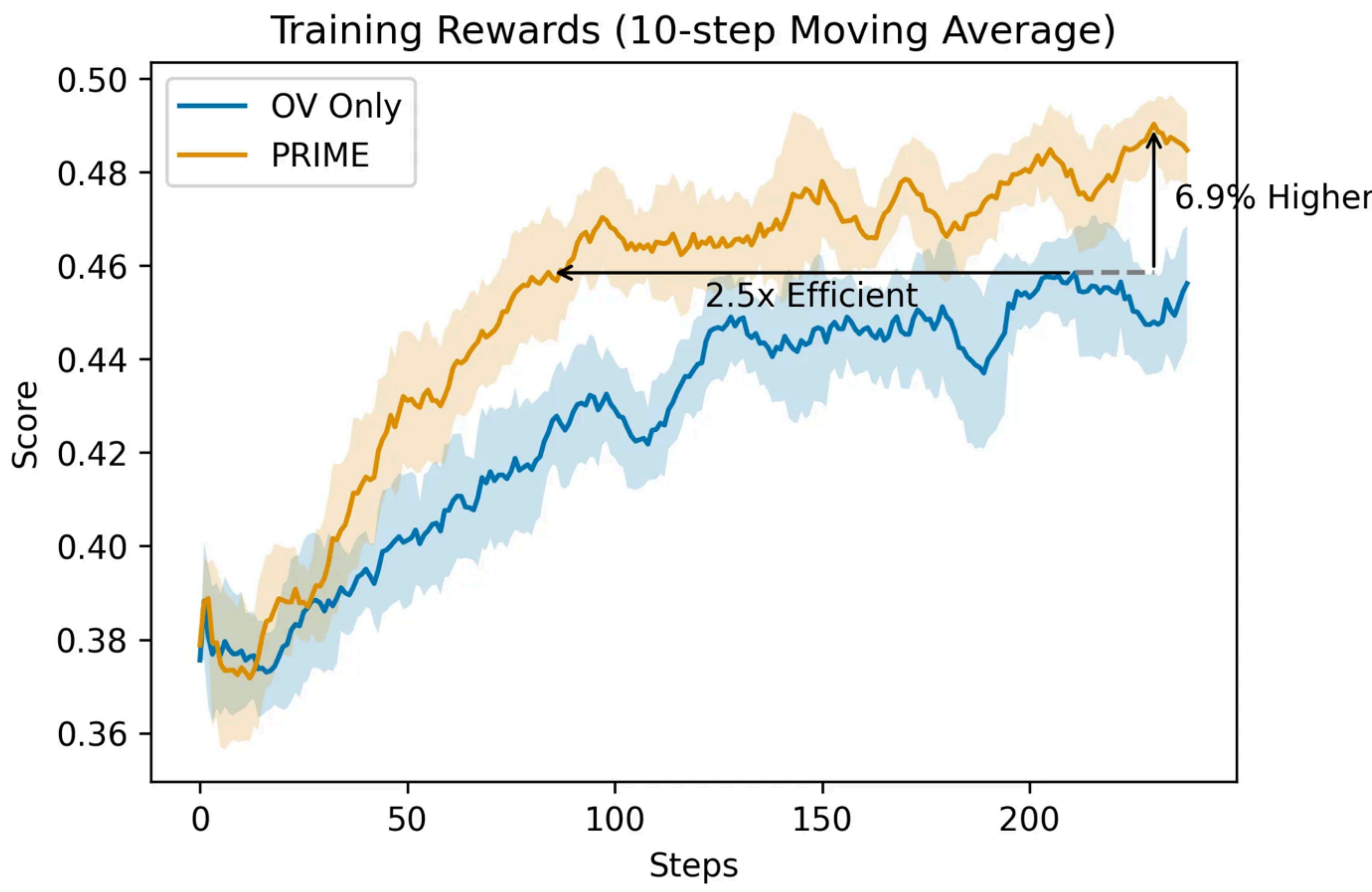
Deepseek-R1

The pass@1 accuracy tested with greedy decoding.

	AIME 2024	MATH 500	AMC	Minerva Math	OlympiadBench	Average
Qwen Math Base 7B	23.3	65.4	47.5	9.9	23.4	33.9
Llama-3.1-70B-Instruct	16.7	64.6	30.1	35.3	31.9	35.7
DPO-R1-Zero	26.7	76.8	62.5	30.9	37.9	47.0
PPO-R1-Zero	43.3	79.4	62.5	33.1	40.7	**51.8**

Prime: Leverage Implicit Process Reward





Cui, G., Yuan, L., Wang, Z., Wang, H., Li, W., He, B., ... & Ding, N. (2025). Process reinforcement through implicit rewards. arXiv preprint arXiv:2502.01456.

Thank you!