Building Math Agents with Multi-turn Iterative Preference Learning

Wei Xiong

Compute Science, University of Illinois Urbana-Champaign

Outline

- Direct preference optimization for multi-turn math problem solving with external tools
- Optimality condition and Algorithmic development
- Experimental results
- Discussion

Multi-turn math problem solving with external tool

Prompt:
User: 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 the piecewise function is continuous (which means that its graph can be drawn without lifting your pencil from the paper) $(s_1 = x)$

Multi-turn math problem solving with external tool

Prompt:
User: 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 the piecewise function is continuous (which means that its graph can be drawn without lifting your pencil from the paper) $(s_1 = x)$

Assistant: The function must be continuous at 2 and -2. That means $x-5 = ax+3$ at $x = 2$ and $x-5 = 2x-b$ at $x = -2$. Let's now solve for x in each case.

Multi-turn math problem solving with external tool

Prompt:
User: Let

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

Find $a + b$ if the piecewise function is continuous (which means that its graph can be drawn without lifting your pencil from the paper) $(s_1 = x)$

Assistant: The function must be continuous at 2 and -2. That means $x - 5 = ax + 3$ at $x = 2$ and $x - 5 = 2x - b$ at $x = -2$. Let's now solve for x in each case.

```
from sympy import symbols, Eq, solve
## define the variables
x, a, b = symbols('x a b')


## the function must be continuous at 2
eq1 = Eq(x-5, a*x + 3)
eq1_sol = solve(eq1, x)

## the function must be continuous at -2
eq2 = Eq(x-5, 2*x - b)
eq2_sol = solve(eq2, x)

## print the solutions
print(eq1_sol)
print(eq2_sol)
```

(a_1)

Multi-turn math problem solving with external tool

Prompt:
User: Let

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

Find $a+b$ if the piecewise function is continuous (which means that its graph can be drawn without lifting your pencil from the paper) $(s_1 = x)$

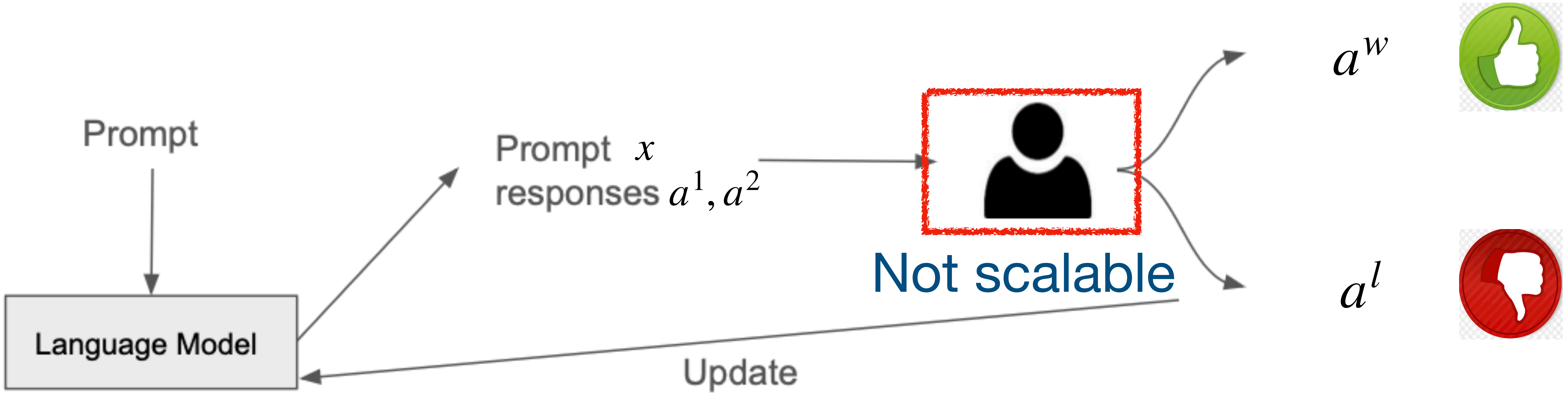
Assistant: The function must be continuous at 2 and -2. That means $x-5 = ax+3$ at $x = 2$ and $x-5 = 2x-b$ at $x = -2$. Let's now solve for x in each case.

```python
from sympy import symbols, Eq, solve
## define the variables
x, a, b = symbols('x a b')


## the function must be continuous at 2
eq1 = Eq(x-5, a*x + 3)
eq1_sol = solve(eq1, x)

## the function must be continuous at -2
eq2 = Eq(x-5, 2*x - b)
eq2_sol = solve(eq2, x)

## print the solutions
print(eq1_sol)
print(eq2_sol)
```

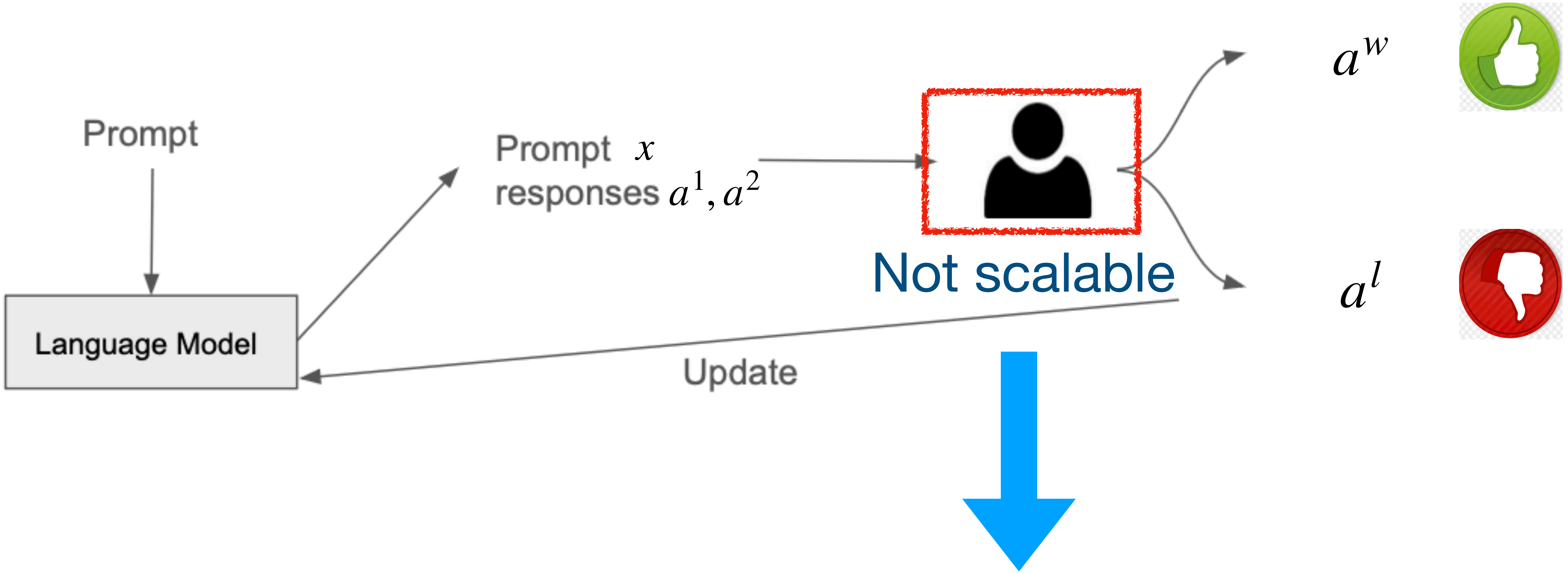
(a_1)

User: “‘output [-8/(a - 1)] [b - 5]‘“ (o_1)

Reinforcement Learning from Human Feedback (RLHF)



Reinforcement Learning from Human Feedback (RLHF)



Bradley-Terry (BT) model :

$$\mathscr{P}^{\star}_{BT}(a^1 \succ a^2 \mid x, a^1, a^2) = \frac{e^{r^{\star}(x,a^1)}}{e^{r^{\star}(x,a^1)} + e^{r^{\star}(x,a^2)}}$$

Imperfect but scalable

Direct preference optimization (DPO)

- Gibbs distribution

$$\pi_r(\cdot \mid x) = \max_{\pi} \left[\mathbb{E}_{a\sim\pi(\cdot|x)}[r(x,a)] - \eta \mathrm{KL}(\pi(\cdot \mid x), \pi_{\mathrm{ref}}(\cdot \mid x)) \right] = \frac{1}{Z(x)} \cdot \pi_{\mathrm{ref}}(\cdot \mid x) \cdot \exp\left(\frac{1}{\eta} r(x,\cdot)\right)$$

- Re-parameterize reward by policy:

$$r(x,a) = \underbrace{\eta \log \frac{\pi_r(a \mid x)}{\pi_{\mathrm{ref}}(a \mid x)}}_{\text{Implicit reward}} + \cancel{\eta \log Z(x)}$$

$\log Z(x)$ will be cancelled in reward difference

- MLE in reward space -> policy optimization:

$$\mathscr{L}_{\mathrm{DPO}}(\theta) = -\sum_{(x,a^w,a^l)\in\mathscr{D}} \log \sigma\left(\eta\left[\log \frac{\pi_\theta(a^l \mid x)}{\pi_{\mathrm{ref}}(a^l \mid x)} - \log \frac{\pi_\theta(a^w \mid x)}{\pi_{\mathrm{ref}}(a^w \mid x)}\right]\right).$$

Fact: If no approximation error + no optimization error:

DPO admits the same optimal policy as RLHF

Rafailov, Rafael, et al. Direct preference optimization: Your language model is secretly a reward model. NeurIPS 2024.

Multi-turn math problem solving with external tool

- LLM as a math agent in tool-integrated reasoning $\tau = (x, \underbrace{a_1, o_1, \cdots, o_{H-1}, a_H}_{y})\,.$

$$s_1 = x \sim d_0, a_1 \sim \pi_1(\cdot \mid s_1), o_1 \sim \mathbb{P}_1(\cdot \mid s_1, a_1), s_2 = (s_1, a_1, o_1) \ldots;$$

- Trajectory preference model

$$\mathscr{P}^\star_{BT}(y^1 \succ y^2 \mid x, y^1, y^2) = \frac{e^{u^\star(x,y^1)}}{e^{u^\star(x,y^1)} + e^{u^\star(x,y^2)}}$$

- Learning objective

$$\arg\max_\pi J(\pi; \mathscr{M}^\star, \pi_{\text{ref}}) = \mathbb{E}_{x\sim d_0}\mathbb{E}_{a_h\sim\pi_h(\cdot|s_h), o_h\sim\mathbb{P}_h(\cdot|s_h,a_h)}\left[u^\star(x,y) - \eta\sum_{h=1}^{H}\text{KL}\big(\pi_h(\cdot \mid s_h), \pi_{\text{ref},h}(\cdot \mid s_h)\big)\right].$$

Optimality condition: layer-wise Q-Gibbs distributions

Initialize: $Q_{\mathscr{M},H}(s_H, a_H) = u(s_H, a_H)$

Step H: single-step decision making, similar to the original DPO

$$\pi_{\mathscr{M},H}(\cdot \mid s_H) = \arg\max_{\pi_H} \mathbb{E}_{a_H \sim \pi_H(\cdot|s_H)} \Big(Q_{\mathscr{M},H}(s_H, a_H) - \eta \cdot \mathrm{KL}\big(\pi_H(\cdot \mid s_H), \pi_{\mathrm{ref},H}(\cdot \mid s_H)\big) \Big) \propto \boxed{\pi_{\mathrm{ref},H}(\cdot \mid s_H) \cdot \exp\left(\frac{Q_{\mathscr{M},H}(s_H, \cdot)}{\eta}\right)}.$$

$$V_{\mathscr{M},H}(s_H) = \mathbb{E}_{a_H \sim \pi_{\mathscr{M},H}(\cdot|s_H)} \Big[Q_{\mathscr{M},H}(s_H, a_H) - \eta \cdot \mathrm{KL}\big(\pi_H(\cdot \mid s_H), \pi_{\mathrm{ref},H}(\cdot \mid s_H)\big) \Big].$$

Gibbs distribution

Optimality condition: layer-wise Q-Gibbs distributions

Initialize: $Q_{\mathscr{M},H}(s_H, a_H) = u(s_H, a_H)$

Step H: single-step decision making, similar to the original DPO

$$\pi_{\mathscr{M},H}(\cdot \mid s_H) = \arg\max_{\pi_H} \mathbb{E}_{a_H \sim \pi_H(\cdot \mid s_H)} \Big(Q_{\mathscr{M},H}(s_H, a_H) - \eta \cdot \mathrm{KL}\big(\pi_H(\cdot \mid s_H), \pi_{\mathrm{ref},H}(\cdot \mid s_H)\big) \Big) \propto \pi_{\mathrm{ref},H}(\cdot \mid s_H) \cdot \exp\left(\frac{Q_{\mathscr{M},H}(s_H, \cdot)}{\eta} \right).$$

$$V_{\mathscr{M},H}(s_H) = \mathbb{E}_{a_H \sim \pi_{\mathscr{M},H}(\cdot \mid s_H)} \Big[Q_{\mathscr{M},H}(s_H, a_H) - \eta \cdot \mathrm{KL}\big(\pi_H(\cdot \mid s_H), \pi_{\mathrm{ref},H}(\cdot \mid s_H)\big) \Big].$$

Step H-1: treat the future as a meta step

$$Q_{\mathscr{M},H-1}(s_{H-1}, a_{H-1}) = \mathbb{E}_{o_{H-1} \sim \mathbb{P}_{H-1}(\cdot \mid s_{H-1}, a_{H-1})} \Big[V_{\mathscr{M},H}(s_H) \Big].$$

$$\pi_{\mathscr{M},H-1}(\cdot \mid s_{H-1}) \propto \pi_{\mathrm{ref},H-1}(\cdot \mid s_{H-1}) \cdot \exp\left(\frac{Q_{\mathscr{M},H-1}(s_{H-1}, \cdot)}{\eta} \right).$$

$$V_{\mathscr{M},H-1}(s_{H-1}) = \mathbb{E}_{a_{H-1} \sim \pi_{\mathscr{M},H-1}(\cdot \mid s_{H-1})} \Big[Q_{\mathscr{M},H-1}(s_{H-1}, a_{H-1}) - \eta \cdot \mathrm{KL}\big(\pi_{H-1}(\cdot \mid s_{H-1}), \pi_{\mathrm{ref},H-1}(\cdot \mid s_{H-1})\big) \Big].$$

……

Multi-turn direct preference learning: re-parameterize

- Re-parameterization trick to connect the **model** with the **policy**

Term (B) will be cancelled in reward difference

$$u(s_H, a_H) = \underbrace{\eta \sum_{h=1}^{H} \log \frac{\pi_{\mathcal{M},h}(a_h \mid s_h)}{\pi_{\mathrm{ref},h}(a_h \mid s_h)}}_{\text{term (A)}} + \cancel{\underbrace{V_{\mathcal{M},1}(s_1)}_{\text{term (B)}}} + \underbrace{\sum_{h=1}^{H-1} \left[V_{\mathcal{M},h+1}(s_{h+1}) - \mathbb{E}_{o_h \sim \mathbb{P}_h(\cdot|s_h,a_h)} V_{\mathcal{M},h+1}(s_{h+1}) \right]}_{\text{term (C)}}.$$

- Term (C) cannot be directly computed except for

$$V_{\mathcal{M},h}(s_h) = \mathbb{E}_{a_{h'} \sim \pi_{\mathcal{M},h'}(\cdot|s_{h'}), h' \geq h} \left[u(s_H, a_H) - \eta \cdot \sum_{h' \geq h} \mathrm{KL}\left(\pi_{h'}(\cdot \mid s_{h'}), \pi_{\mathrm{ref},h'}(\cdot \mid s_{h'})\right) \right].$$

  - $H = 1$: original DPO

Implementation: mask out the external messages.

  - o_h is deterministic given the history

$$\ell_{\mathcal{D}}(\theta) = \sum_{(x,\tau^w,\tau^l) \in \mathcal{D}} \log\Big(\sigma\big(u_\theta(x, y^w) - u_\theta(x, y^l)\big)\Big)$$

$$\mathcal{L}_{\text{M-DPO}}(\theta) = -\sum_{(x,\tau^w,\tau^l) \in \mathcal{D}} \log \sigma\left(\eta \sum_{h=1}^{H} \left[\log \frac{\pi_{\theta,h}(a_h^l \mid s_h^l)}{\pi_{\mathrm{ref},h}(a_h^l \mid s_h^l)} - \log \frac{\pi_{\theta,h}(a_h^w \mid s_h^w)}{\pi_{\mathrm{ref},h}(a_h^w \mid s_h^w)}\right]\right).$$

Multi-turn direct preference learning: re-parameterize

- Re-parameterization trick to connect the **model** with the **policy**

Term (B) will be cancelled in reward difference

$$u(s_H, a_H) = \underbrace{\eta \sum_{h=1}^{H} \log \frac{\pi_{\mathscr{M},h}(a_h \mid s_h)}{\pi_{\text{ref},h}(a_h \mid s_h)}}_{\text{term (A)}} + \cancel{\underbrace{V_{\mathscr{M},1}(s_1)}_{\text{term (B)}}} + \underbrace{\sum_{h=1}^{H-1} \left[V_{\mathscr{M},h+1}(s_{h+1}) - \mathbb{E}_{o_h \sim \mathbb{P}_h(\cdot|s_h,a_h)} V_{\mathscr{M},h+1}(s_{h+1}) \right]}_{\text{term (C)}}.$$

- Term (C) can be estimated if
  - we train a value network simultaneously
  - use MC estimation

Direct learning from dataset seems to be impossible.

Motivating example

- Force model to predict the low-quality external message can hurt the model performance

An example of external messages returned by the Python interpreter

Prompt:

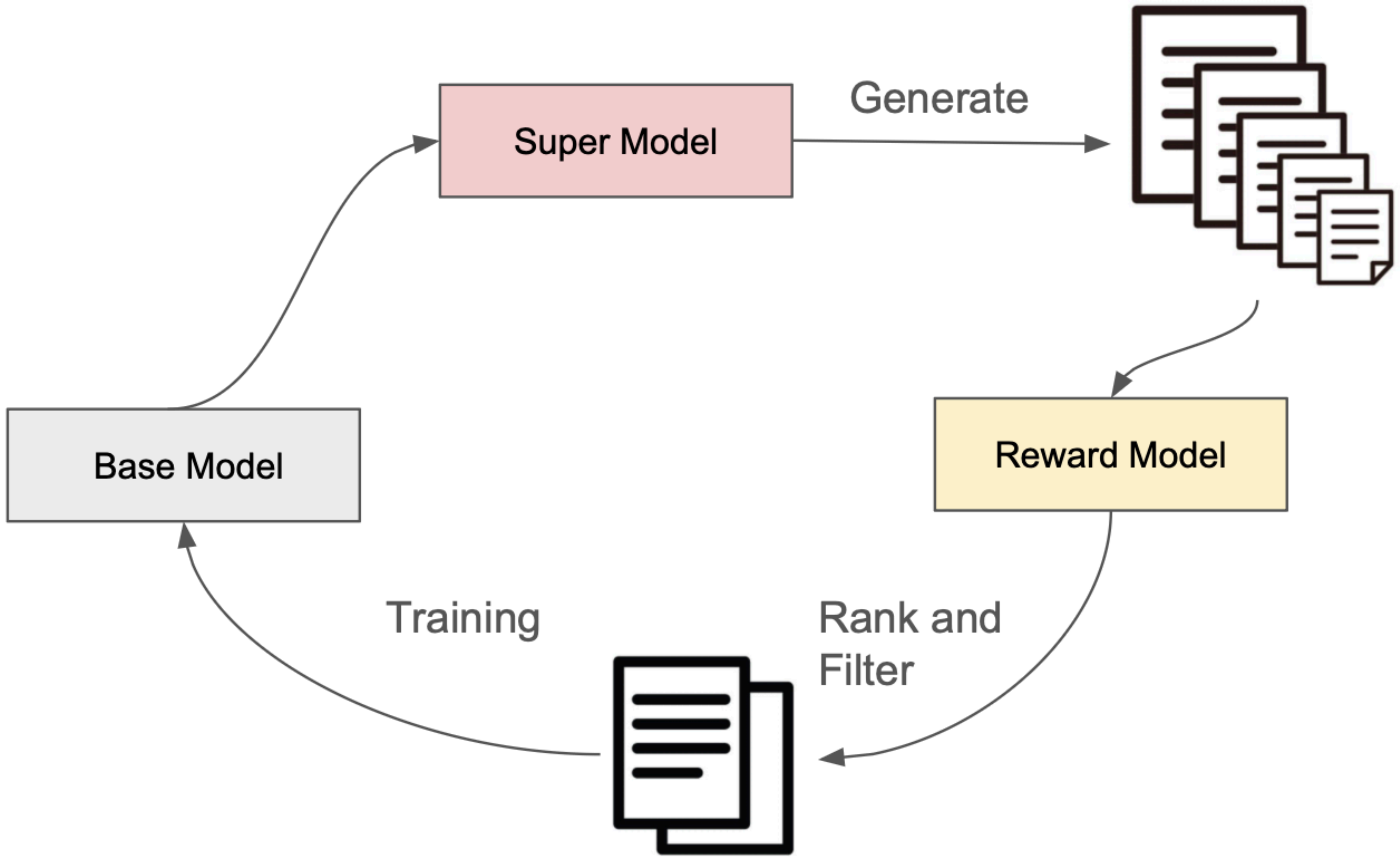
User: Convert the periodic decimal notation 0.42 followed by an infinite sequence of 42s into a reduced fraction $\frac{p}{q}$, such that p and q have no common factors other than 1. Find the sum of p and q.

```python
import sympy as sp

# Define the number as a sympy number with infinite repeating digits
digits = '42'
num = sp.N(0.4242424242, 1000000)  # 1000000 digits are accurate
num
```

User: “‘output 0.42424242420000000234043000091332942247390747070312500000000000000000 (omit other 320 ’0’)
File "/site-packages/timeout_decorator/timeout_decorator.py", line 69, in handler_raise_exception(timeout_exception, exception_message)
File "../anaconda3/envs/inference/lib/python3.10/site-packages/timeout_decorator/timeout_decorator.py", line 45, in _raise_exception
    raise exception()
timeout_decorator.timeout_decorator.TimeoutError: ’Timed Out’ “‘

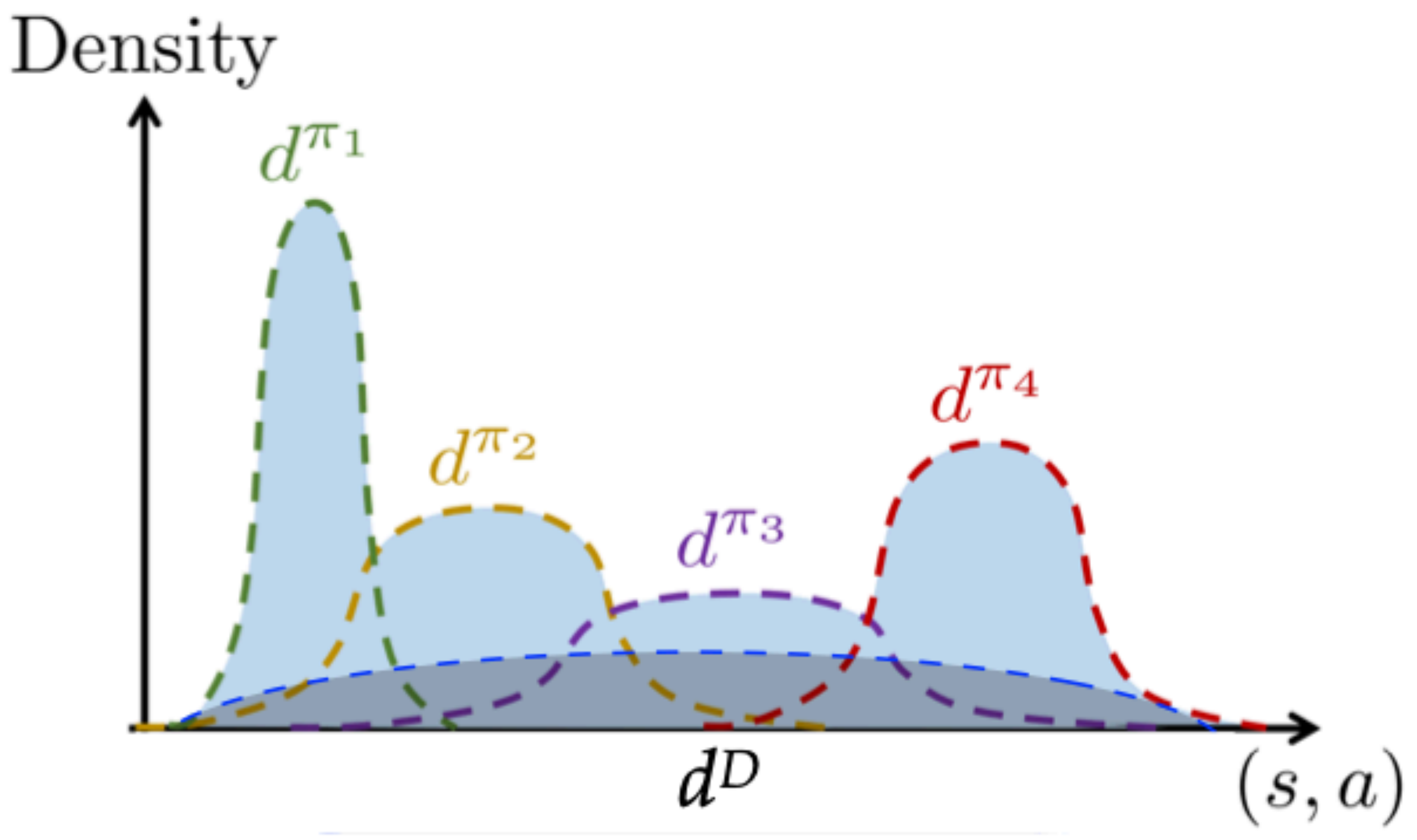
Warm-up SFT: Reward-rAnked Fine-Tuning (RAFT)

- We use an open-source dataset collected by best-of-n sampling and final result checking
- 510K correct trajectories on MATH and GSM8K

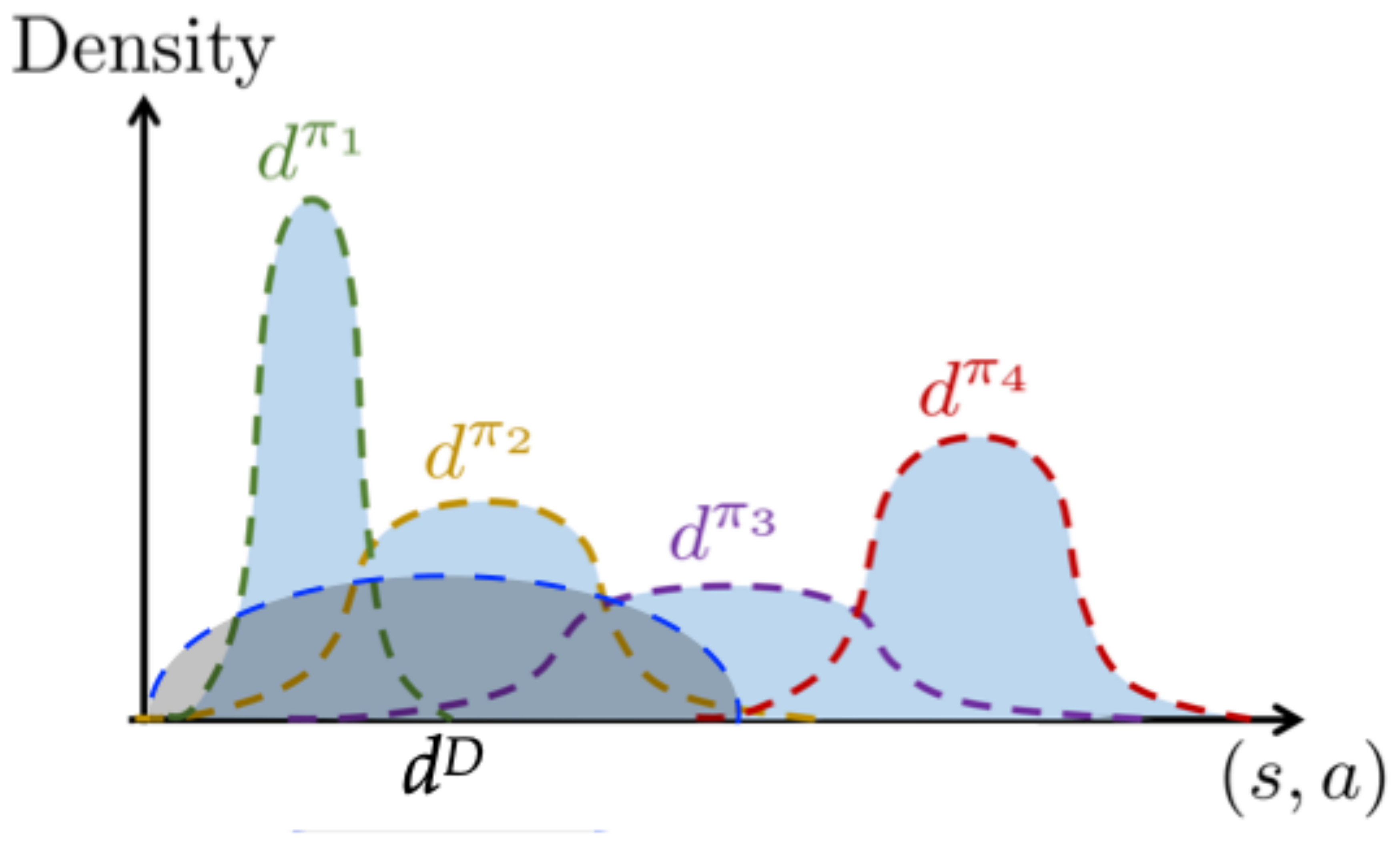


(α–β) Dong H, **Xiong W**, et al. Raft: Reward ranked finetuning for generative foundation model alignment. TMLR, 2023.

Learning with a fixed preference dataset is suboptimal



All policy coverage: MLE is efficient.



Data only covers π_1, π_2: **pessimistic** MLE policy can compete with the best among them.

- Along the way of PPO training, the KL divergence can be > 30

We generally cannot expect a good coverage from a fixed dataset.

Related studies on the role of pessimism in offline learning

Xiong W*, Zhong H*, She C, et al. Nearly minimax optimal offline reinforcement learning with linear function approximation: Single-agent MDP and Markov game. ICLR 2023.

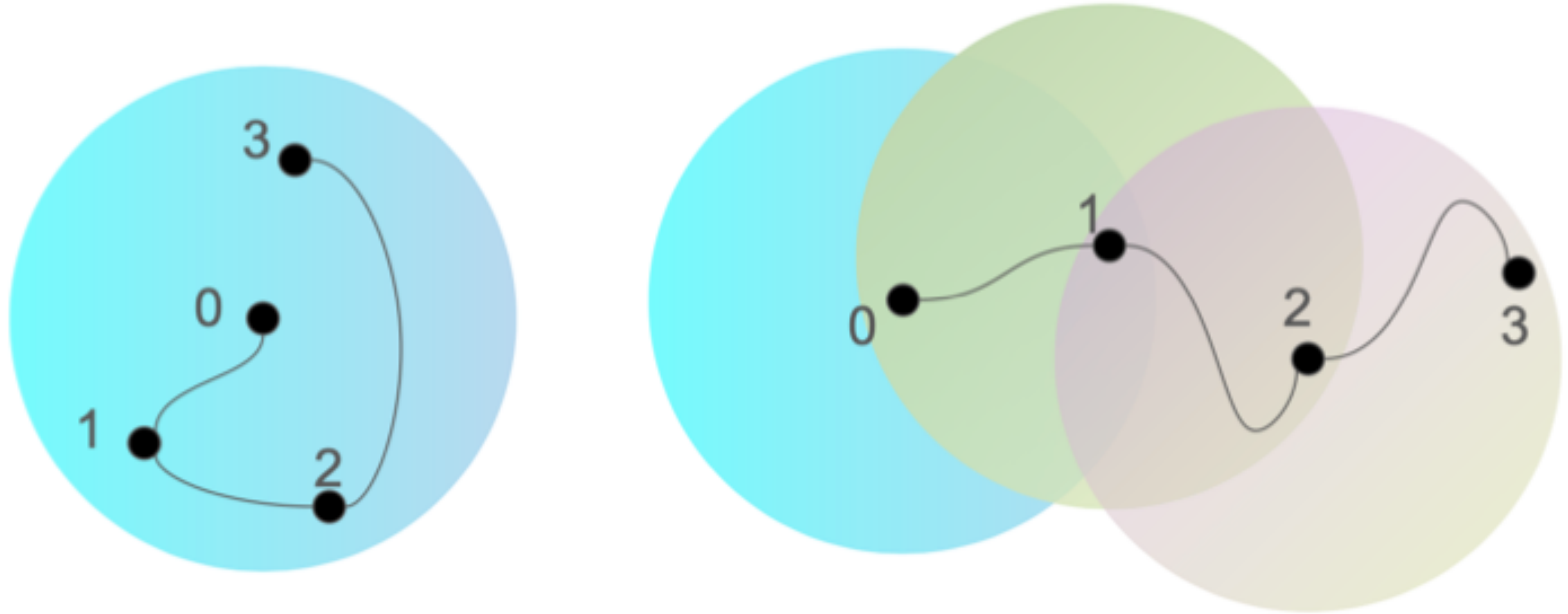
Zhong H*, **Xiong W***, Tan J*, et al. Pessimistic minimax value iteration: Provably efficient equilibrium learning from offline datasets. ICML 2022.

Batch online exploration

- For $t = 1,2,3...$
  - **Exploit** the historical information to get π_t^1 by running M-DPO based on $\mathscr{D}_{1:t-1}$ with $\pi_{\mathrm{ref}}^t(\cdot \mid x) = \pi_{t-1}^1$
  - **Explore:** maximize the data diversity by π_t^1 variant
    - Use checkpoints at different training steps
    - Use more advanced sampling strategy (bon sampling/MCTS)
  - Generate m pairs as $\mathscr{D}_t$
    - For each prompt we generate a pair with a correct trajectory and wrong trajectory

Batch online exploration

- For $t = 1,2,3...$
  - **Exploit** the historical information to get π_t^1 by running M-DPO based on $\mathscr{D}_{1:t-1}$ with $\boxed{\pi_{\text{ref}}^t(\cdot \mid x) = \pi_{t-1}^1}$



Left: fixed reference model. Right: $\pi_{\text{ref}}^t = \pi_{t-1}^1$

Xiong W, Dong H, Ye C, et al. Iterative preference learning from human feedback: Bridging theory and practice for RLHF under KL-constraint, ICML 2024

Multi-turn direct preference learning improves math

Base Model	Method	with Tool	GSM8K	MATH	AVG
WizardMath-7B	SFT for CoT	✗	54.9	10.7	32.8
WizardMath-13B	SFT for CoT	✗	63.9	14.0	39.0
WizardMath-70B	SFT for CoT	✗	81.6	22.7	52.2
CodeLLaMA-2-7B	SFT	✓	75.9	43.6	59.8
CodeLLaMA-2-13B	SFT	✓	78.8	45.5	62.2
CodeLLaMA-2-34B	SFT	✓	80.7	48.3	64.5
LLaMA-2-70B	SFT	✓	84.7	46.3	65.5
CodeLLaMA-2-70B	SFT	✓	84.6	50.7	67.7
Gemma-1.1-it-7B	SFT†	✓	77.5	46.1	61.8
Gemma-1.1-it-7B	RAFT	✓	79.2	47.3	63.3
Gemma-1.1-it-7B	Iterative Single-turn DPO	✓	81.7	48.9	65.3
Gemma-1.1-it-7B	Iterative Single-turn KTO	✓	80.6	49.0	64.8
Gemma-1.1-it-7B	Iterative M-DPO + fixed reference	✓	79.9	48.0	64.0
Gemma-1.1-it-7B	M-DPO Iteration 1	✓	81.5	49.1	65.3
Gemma-1.1-it-7B	M-DPO Iteration 2	✓	82.5	49.7	66.1
Gemma-1.1-it-7B	M-DPO Iteration 3	✓	83.9 ↑6.4	51.2 ↑5.1	67.6 ↑5.8
Gemma-1.1-it-7B	Iterative M-KTO	✓	82.1 ↑4.6	49.5 ↑3.4	65.8 ↑4.0
CodeGemma-1.1-it-7B	SFT†	✓	77.3	46.4	61.9
CodeGemma-1.1-it-7B	RAFT	✓	78.8	48.4	63.6
CodeGemma-1.1-it-7B	Iterative Single-turn DPO	✓	79.1	48.9	64.0
CodeGemma-1.1-it-7B	Iterative Single-turn KTO	✓	80.2	48.6	64.4
CodeGemma-1.1-it-7B	Iterative M-DPO	✓	81.5 ↑4.2	50.1 ↑3.7	65.8 ↑4.0
CodeGemma-1.1-it-7B	Iterative M-KTO	✓	81.6 ↑4.3	49.6 ↑3.2	65.6 ↑3.8
Mistral-7B-v0.3	SFT†	✓	77.8	42.7	60.3
Mistral-7B-v0.3	RAFT	✓	79.8	43.7	61.8
Mistral-7B-v0.3	Iterative Single-turn DPO	✓	79.8	45.1	62.5
Mistral-7B-v0.3	Iterative Single-turn KTO	✓	81.3	46.3	63.8
Mistral-7B-v0.3	Iterative M-DPO	✓	82.3 ↑4.5	47.5 ↑4.8	64.9 ↑4.7
Mistral-7B-v0.3	Iterative M-KTO	✓	81.7 ↑3.9	46.7 ↑4.0	64.2 ↑4.0
Gemma-2-it-9B	SFT†	✓	84.1	51.0	67.6
Gemma-2-it-9B	RAFT	✓	84.2	52.6	68.4
Gemma-2-it-9B	Iterative Single-turn DPO	✓	85.2	53.1	69.2
Gemma-2-it-9B	Iterative Single-turn KTO	✓	85.4	52.9	69.2
Gemma-2-it-9B	Iterative M-DPO	✓	**86.3** ↑2.2	**54.5** ↑3.5	70.4 ↑2.9
Gemma-2-it-9B	Iterative M-KTO	✓	86.1 ↑2.0	54.5 ↑3.5	70.3 ↑2.8

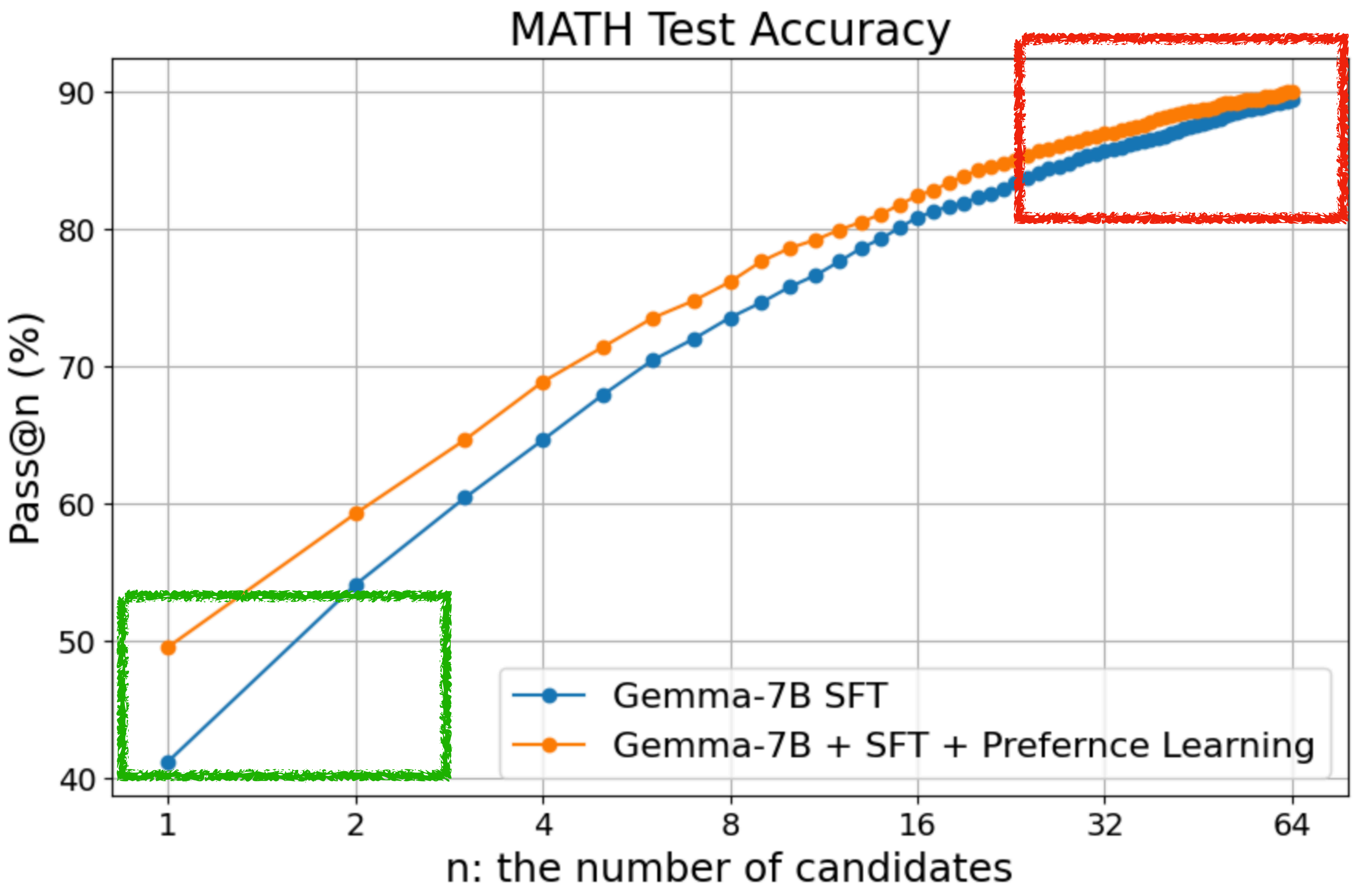
Multi-turn direct preference learning improves math

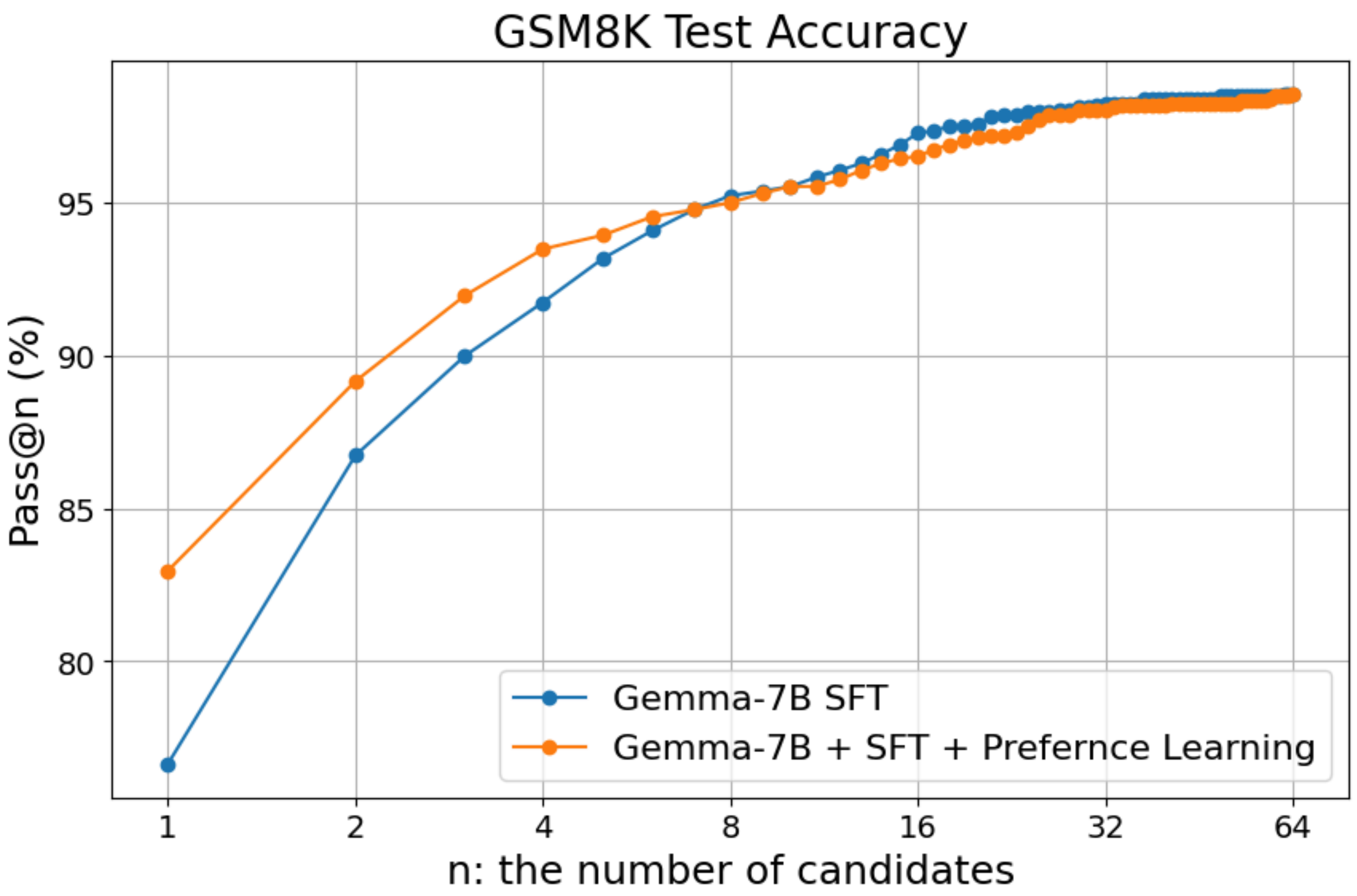
Base Model	Method	with Tool	GSM8K	MATH	AVG
WizardMath-7B	SFT for CoT	✗	54.9	10.7	32.8
WizardMath-13B	SFT for CoT	✗	63.9	14.0	39.0
WizardMath-70B	SFT for CoT	✗	81.6	22.7	52.2
CodeLLaMA-2-7B	SFT	✓	75.9	43.6	59.8
CodeLLaMA-2-13B	SFT	✓	78.8	45.5	62.2
CodeLLaMA-2-34B	SFT	✓	80.7	48.3	64.5
LLaMA-2-70B	SFT	✓	84.7	46.3	65.5
CodeLLaMA-2-70B	SFT	✓	84.6	50.7	67.7
Gemma-1.1-it-7B	SFT†	✓	77.5	46.1	61.8
Gemma-1.1-it-7B	RAFT	✓	79.2	47.3	63.3
Gemma-1.1-it-7B	Iterative Single-turn DPO	✓	81.7	48.9	65.3
Gemma-1.1-it-7B	Iterative Single-turn KTO	✓	80.6	49.0	64.8
Gemma-1.1-it-7B	Iterative M-DPO + fixed reference	✓	79.9	48.0	64.0
Gemma-1.1-it-7B	M-DPO Iteration 1	✓	81.5	49.1	65.3
Gemma-1.1-it-7B	M-DPO Iteration 2	✓	82.5	49.7	66.1
Gemma-1.1-it-7B	M-DPO Iteration 3	✓	83.9 ↑6.4	51.2 ↑5.1	67.6 ↑5.8
Gemma-1.1-it-7B	Iterative M-KTO	✓	82.1 ↑4.6	49.5 ↑3.4	65.8 ↑4.0
CodeGemma-1.1-it-7B	SFT†	✓	77.3	46.4	61.9
CodeGemma-1.1-it-7B	RAFT	✓	78.8	48.4	63.6
CodeGemma-1.1-it-7B	Iterative Single-turn DPO	✓	79.1	48.9	64.0
CodeGemma-1.1-it-7B	Iterative Single-turn KTO	✓	80.2	48.6	64.4
CodeGemma-1.1-it-7B	Iterative M-DPO	✓	81.5 ↑4.2	50.1 ↑3.7	65.8 ↑4.0
CodeGemma-1.1-it-7B	Iterative M-KTO	✓	81.6 ↑4.3	49.6 ↑3.2	65.6 ↑3.8
Mistral-7B-v0.3	SFT†	✓	77.8	42.7	60.3
Mistral-7B-v0.3	RAFT	✓	79.8	43.7	61.8
Mistral-7B-v0.3	Iterative Single-turn DPO	✓	79.8	45.1	62.5
Mistral-7B-v0.3	Iterative Single-turn KTO	✓	81.3	46.3	63.8
Mistral-7B-v0.3	Iterative M-DPO	✓	82.3 ↑4.5	47.5 ↑4.8	64.9 ↑4.7
Mistral-7B-v0.3	Iterative M-KTO	✓	81.7 ↑3.9	46.7 ↑4.0	64.2 ↑4.0
Gemma-2-it-9B	SFT†	✓	84.1	51.0	67.6
Gemma-2-it-9B	RAFT	✓	84.2	52.6	68.4
Gemma-2-it-9B	Iterative Single-turn DPO	✓	85.2	53.1	69.2
Gemma-2-it-9B	Iterative Single-turn KTO	✓	85.4	52.9	69.2
Gemma-2-it-9B	Iterative M-DPO	✓	**86.3** ↑2.2	**54.5** ↑3.5	70.4 ↑2.9
Gemma-2-it-9B	Iterative M-KTO	✓	86.1 ↑2.0	54.5 ↑3.5	70.3 ↑2.8

Multi-turn direct preference learning improves math

Base Model	Method	with Tool	GSM8K	MATH	AVG
WizardMath-7B	SFT for CoT	✗	54.9	10.7	32.8
WizardMath-13B	SFT for CoT	✗	63.9	14.0	39.0
WizardMath-70B	SFT for CoT	✗	81.6	22.7	52.2
CodeLLaMA-2-7B	SFT	✓	75.9	43.6	59.8
CodeLLaMA-2-13B	SFT	✓	78.8	45.5	62.2
CodeLLaMA-2-34B	SFT	✓	80.7	48.3	64.5
LLaMA-2-70B	SFT	✓	84.7	46.3	65.5
CodeLLaMA-2-70B	SFT	✓	84.6	50.7	67.7
Gemma-1.1-it-7B	SFT†	✓	77.5	46.1	61.8
Gemma-1.1-it-7B	RAFT	✓	79.2	47.3	63.3
Gemma-1.1-it-7B	Iterative Single-turn DPO	✓	81.7	48.9	65.3
Gemma-1.1-it-7B	Iterative Single-turn KTO	✓	80.6	49.0	64.8
Gemma-1.1-it-7B	Iterative M-DPO + fixed reference	✓	79.9	48.0	64.0
Gemma-1.1-it-7B	M-DPO Iteration 1	✓	81.5	49.1	65.3
Gemma-1.1-it-7B	M-DPO Iteration 2	✓	82.5	49.7	66.1
Gemma-1.1-it-7B	M-DPO Iteration 3	✓	83.9 ↑6.4	51.2 ↑5.1	67.6 ↑5.8
Gemma-1.1-it-7B	Iterative M-KTO	✓	82.1 ↑4.6	49.5 ↑3.4	65.8 ↑4.0
CodeGemma-1.1-it-7B	SFT†	✓	77.3	46.4	61.9
CodeGemma-1.1-it-7B	RAFT	✓	78.8	48.4	63.6
CodeGemma-1.1-it-7B	Iterative Single-turn DPO	✓	79.1	48.9	64.0
CodeGemma-1.1-it-7B	Iterative Single-turn KTO	✓	80.2	48.6	64.4
CodeGemma-1.1-it-7B	Iterative M-DPO	✓	81.5 ↑4.2	50.1 ↑3.7	65.8 ↑4.0
CodeGemma-1.1-it-7B	Iterative M-KTO	✓	81.6 ↑4.3	49.6 ↑3.2	65.6 ↑3.8
Mistral-7B-v0.3	SFT†	✓	77.8	42.7	60.3
Mistral-7B-v0.3	RAFT	✓	79.8	43.7	61.8
Mistral-7B-v0.3	Iterative Single-turn DPO	✓	79.8	45.1	62.5
Mistral-7B-v0.3	Iterative Single-turn KTO	✓	81.3	46.3	63.8
Mistral-7B-v0.3	Iterative M-DPO	✓	82.3 ↑4.5	47.5 ↑4.8	64.9 ↑4.7
Mistral-7B-v0.3	Iterative M-KTO	✓	81.7 ↑3.9	46.7 ↑4.0	64.2 ↑4.0
Gemma-2-it-9B	SFT†	✓	84.1	51.0	67.6
Gemma-2-it-9B	RAFT	✓	84.2	52.6	68.4
Gemma-2-it-9B	Iterative Single-turn DPO	✓	85.2	53.1	69.2
Gemma-2-it-9B	Iterative Single-turn KTO	✓	85.4	52.9	69.2
Gemma-2-it-9B	Iterative M-DPO	✓	**86.3** ↑2.2	**54.5** ↑3.5	70.4 ↑2.9
Gemma-2-it-9B	Iterative M-KTO	✓	86.1 ↑2.0	54.5 ↑3.5	70.3 ↑2.8

Preference learning improves top-k responses





Discussion

+ Easy to implement, stable training

- The DPO is not equivalent to RLHF in practice

    Easy to approximate u (by transformer) but not $\log \pi/\pi_{\mathrm{ref}}$

    Optimization error exists in practice

- Bradley Terry model may not be a reasonable assumption beyond the chat style

- DPO cannot be scaled: the best model is achieved at ~30K-50K samples

    The best practice of DPO may be focusing on improving the data quality

Thanks for listening!

Check out more details in our paper!