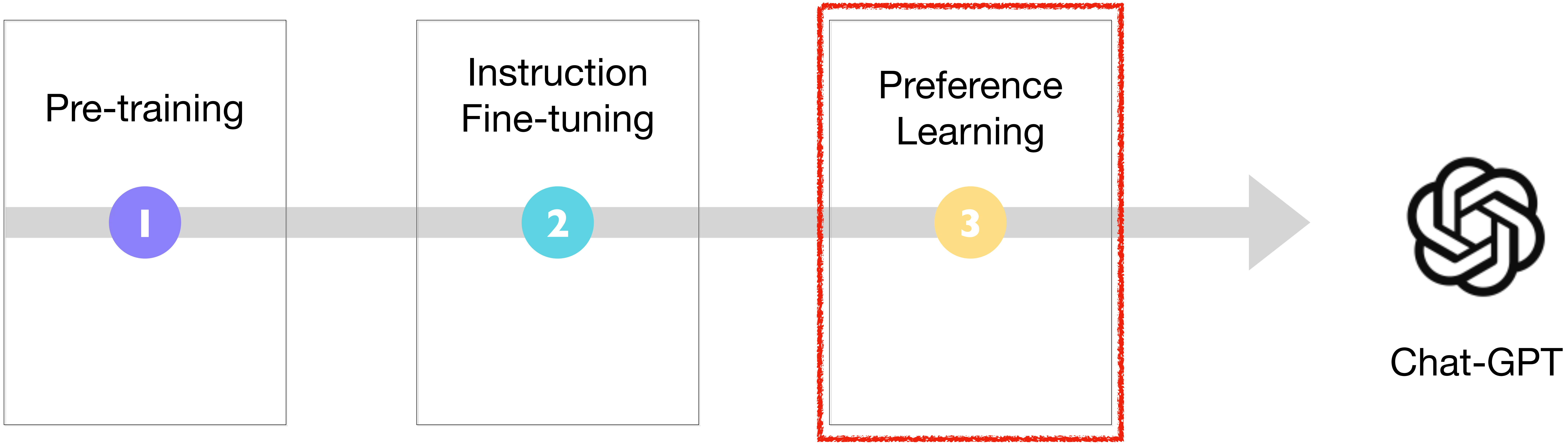
Iterative Preference Learning for Large Language Model Post Training

Wei Xiong

University of Illinois Urbana-Champaign

LLM training pipeline



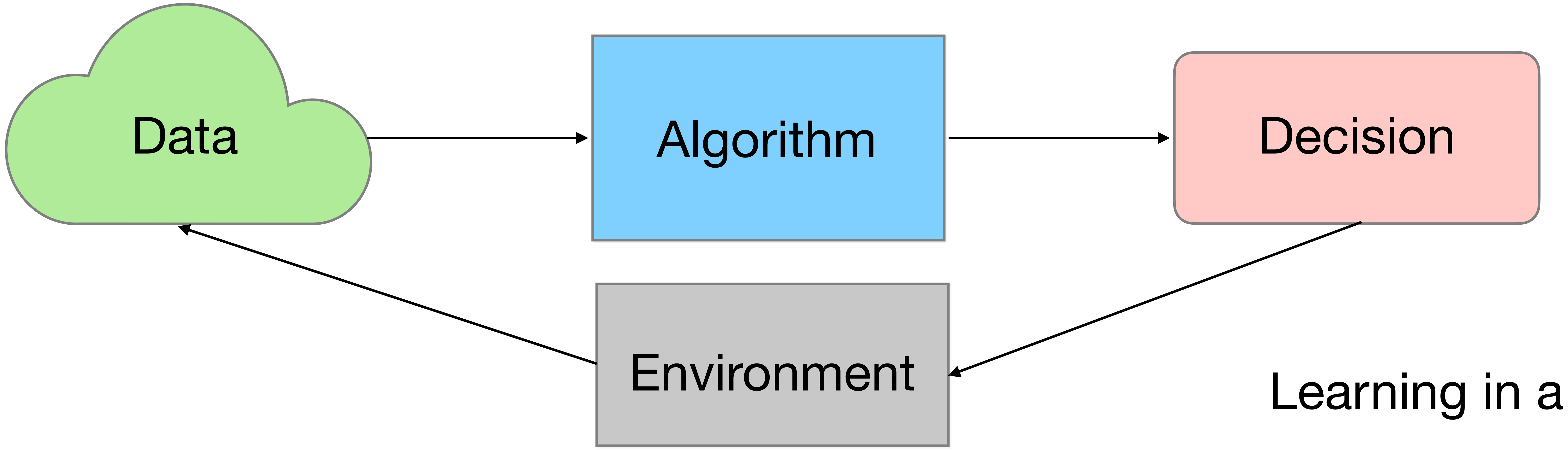
Outline

- Motivation: Preference Learning as **Sequential Decision Making**
- Introduction to Reinforcement Learning from Human Feedback (RLHF)
- Main Results: Online Iterative RLHF Framework
- Practical and Open-source Codebook: RLHFlow

Supervised learning vs decision making

- **Supervised learning** predicts patterns from **passively** observed data
  - Image classification and speech recognition

Supervised learning vs decision making

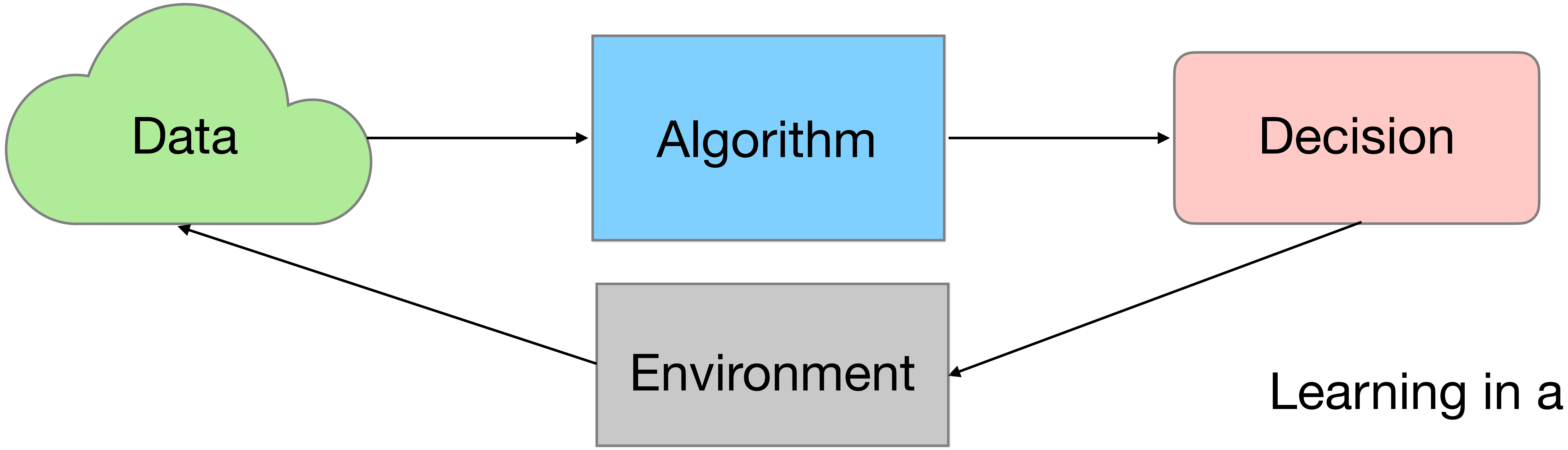


Learning in a feedback loop

- **Supervised learning** predicts patterns from **passively** observed data
  - Image classification and speech recognition
- **Decision making** actively gathers information by **sequential interactions** with the environment
  - Recommendation system, robotics and game playing

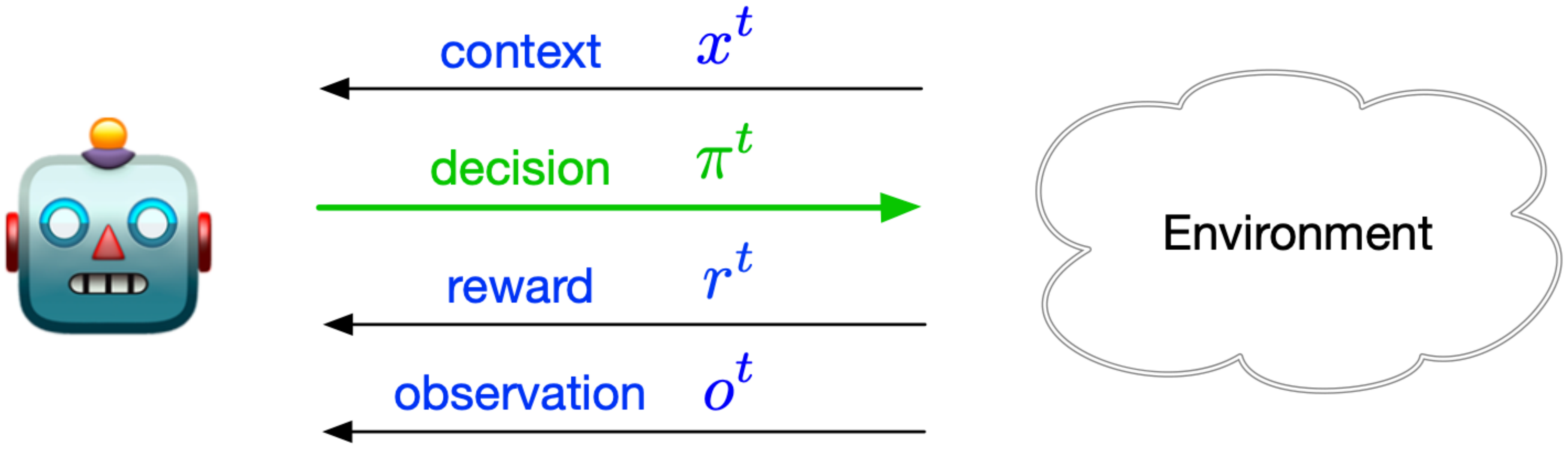
Adapted from tutorial https://dylanfoster.net/slides/bldm.pdf

Preference learning as decision making



Learning in a feedback loop

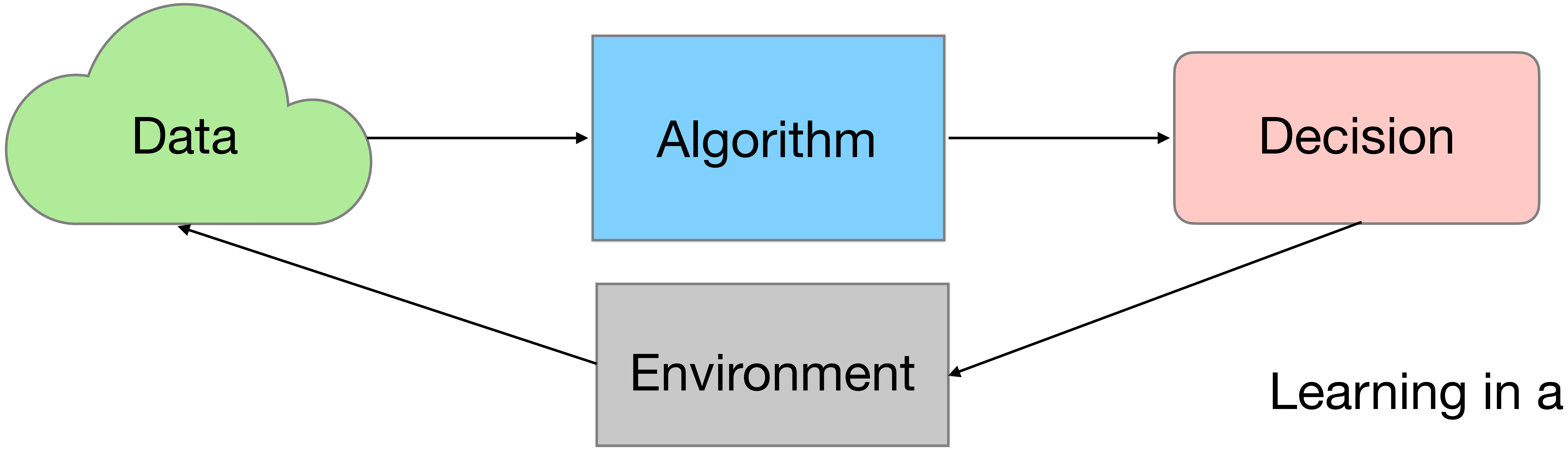
Decision making for T steps



- **Context**: prompt
- **Decision**: (distribution) of response
- **Reward**: human preference (defined later)
- **Observation**: potential external message

Adapted from tutorial https://dylanfoster.net/slides/bldm.pdf

Exploration-exploitation trade-off



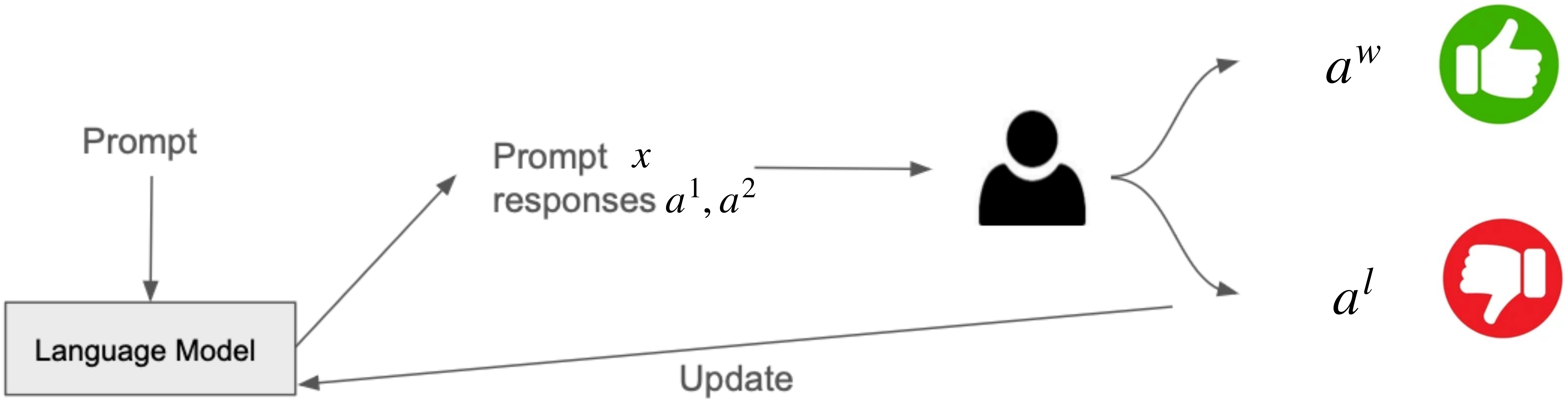
Learning in a feedback loop

- Trade-off between exploration and exploitation in online sequential decision making:
  - want to focus on good decisions based on the history and avoid bad decisions to maximize rewards
  - need to try new decisions to learn the environment

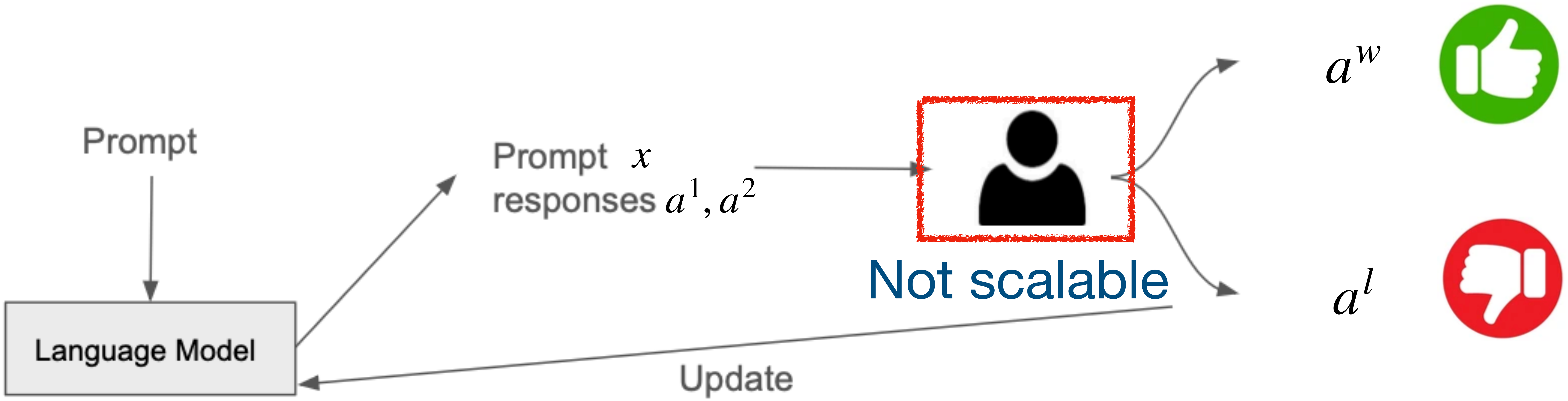
Main research problem: can we design principled preference learning algorithms under this online sequential decision making framework?

Reinforcement Learning from Human Feedback

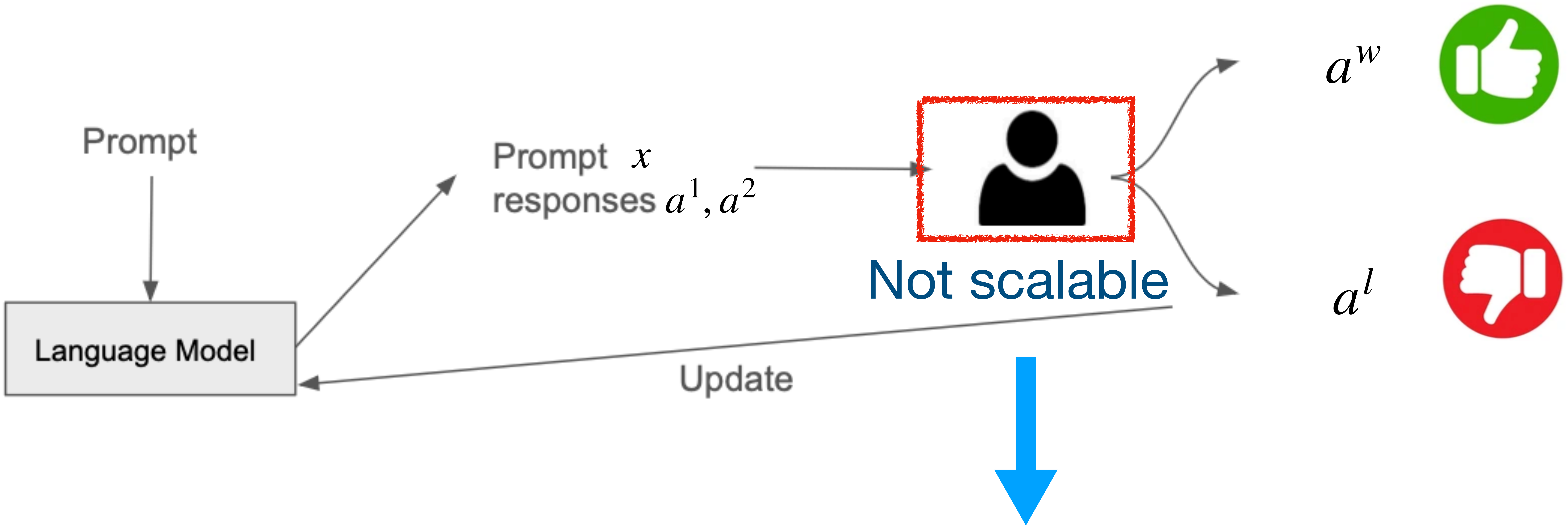
Reinforcement learning from human feedback



Reinforcement learning from human feedback



Reinforcement learning from human feedback



Bradley-Terry (BT) model :

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

Scalable: we can query the reward as many times as we want

Reinforcement learning from human feedback

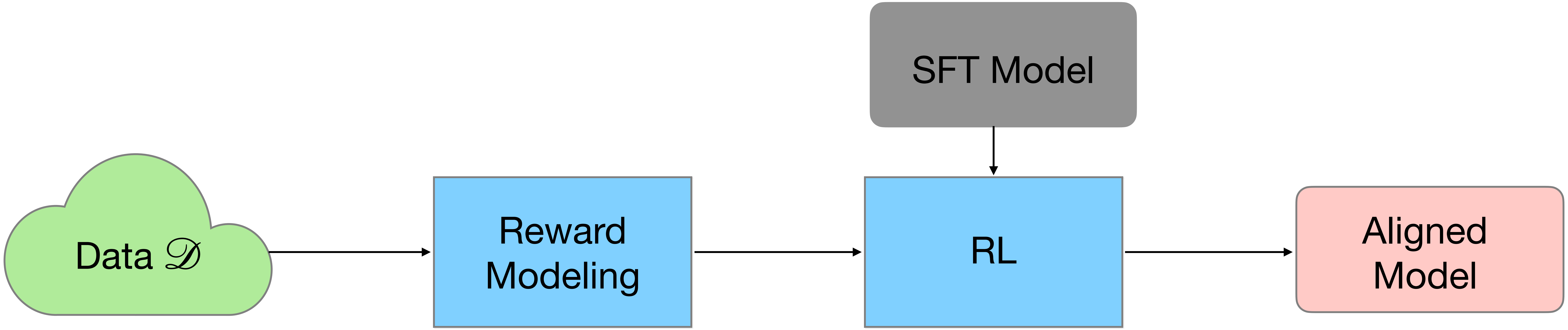
- Learning objective

$$\max_{\pi} J(\pi) = \max_{\pi} \mathbb{E}_{x \sim d_0}\Bigg[\underbrace{\mathbb{E}_{a \sim \pi(\cdot|x)}[r^\star(x, a)]}_{\text{Optimize Reward}} - \underbrace{\eta \mathrm{KL}(\pi(\cdot \mid x), \pi_0(\cdot \mid x))}_{\text{Stay Close to SFT Model } \pi_0} \Bigg].$$

Reinforcement learning from human feedback

- Learning objective

$$\max_{\pi} J(\pi) = \max_{\pi} \mathbb{E}_{x \sim d_0}\Big[\underbrace{\mathbb{E}_{a \sim \pi(\cdot|x)}[r^{\star}(x,a)]}_{\text{Optimize Reward}} - \underbrace{\eta \mathrm{KL}(\pi(\cdot \mid x), \pi_0(\cdot \mid x))}_{\text{Stay Close to SFT Model } \pi_0} \Big].$$



Passively observed data
data from π_0

r_{MLE} = MLE on $\mathscr{D}$

$$r_{\mathrm{MLE}}(x,a) - \eta \cdot \log \frac{\pi(a \mid x)}{\pi_0(a \mid x)}$$

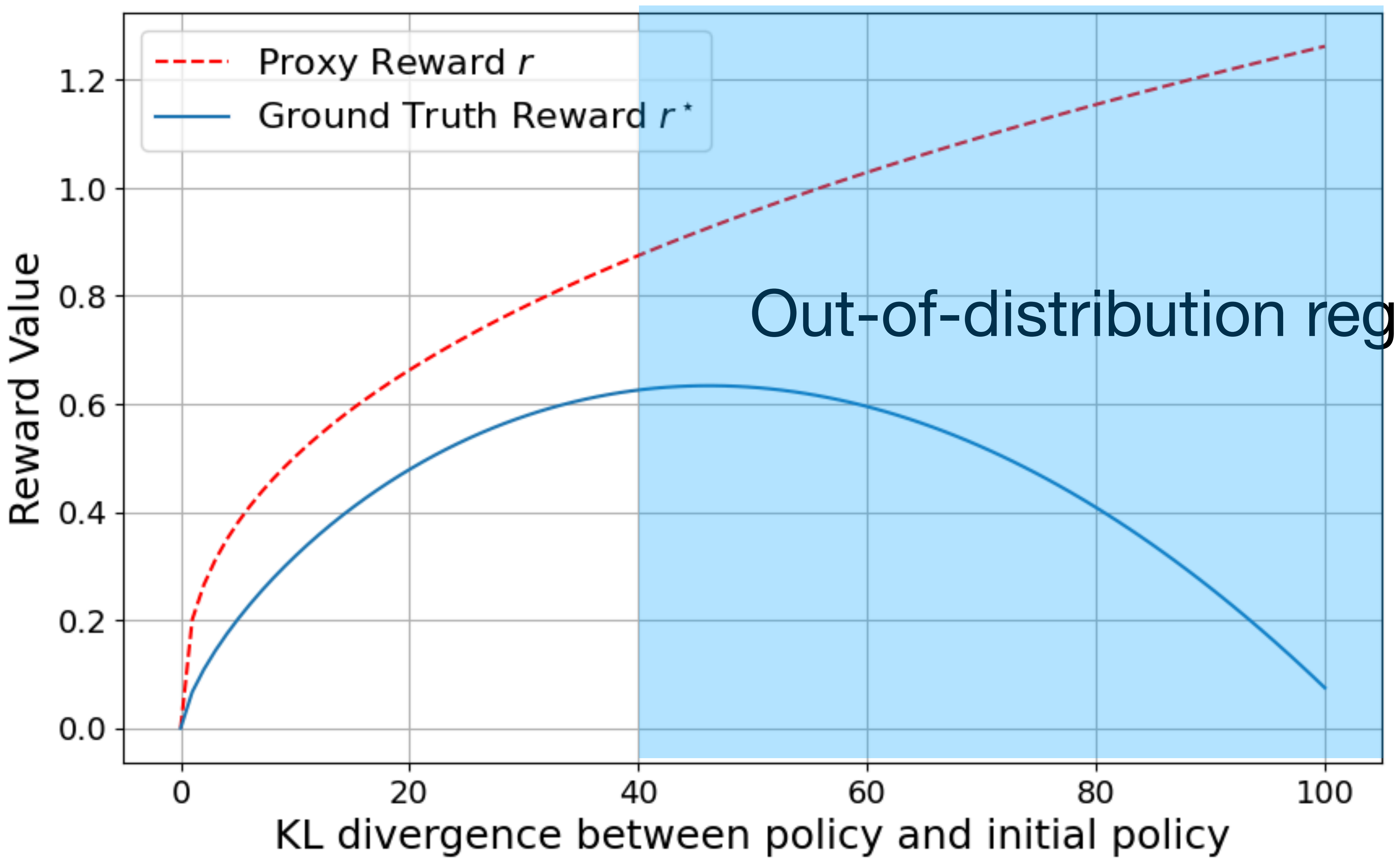
Long, Ouyang, et al. Training language models to follow instructions with human feedback. arxiv, 2022.

Proxy reward over-optimization

A high proxy reward does not necessarily lead to a better performance.

$$\longrightarrow \mathscr{P}^{\star}_{BT}(\cdot \mid x, a^1, a^2), r^{\star} \longrightarrow \quad r \text{ trained from } \mathscr{D}$$

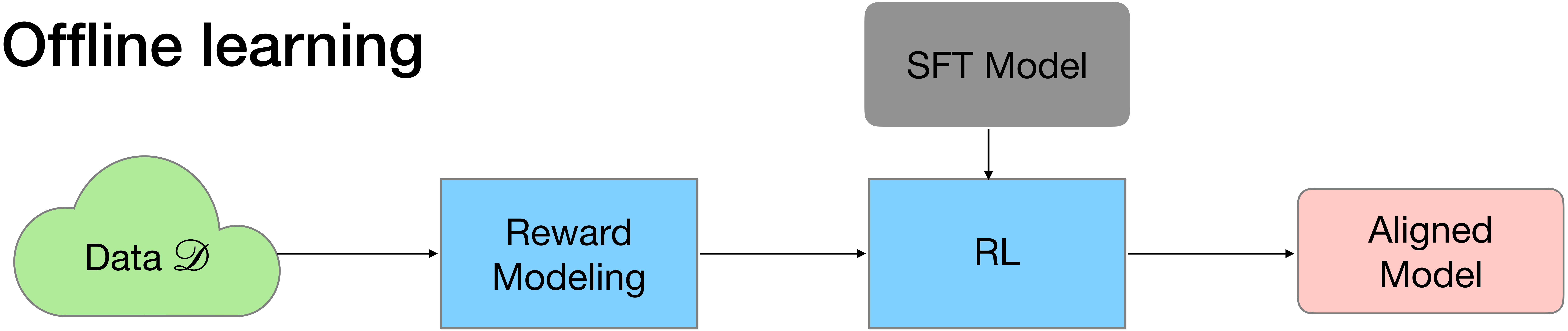
The outputs of LLMs easily fall into OOD regime of proxy reward



Simplified Figure from Gao, Leo, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. ICML, 2023.

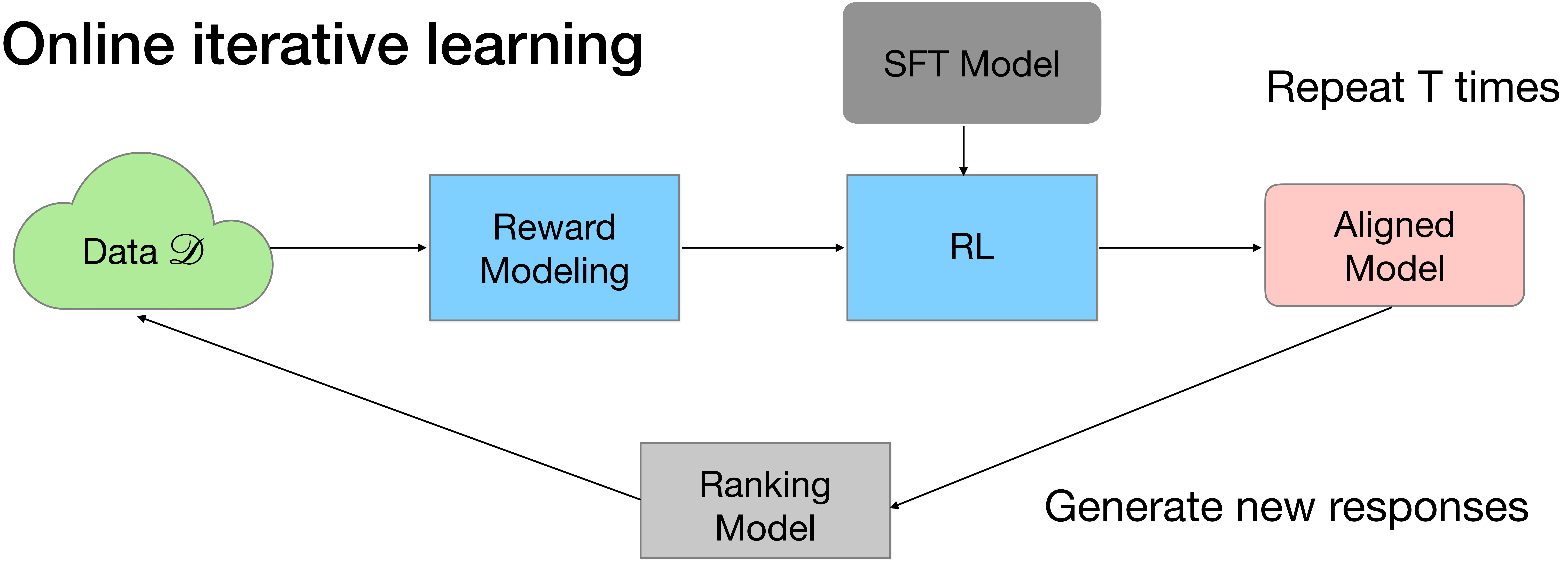
Online Iterative RLHF

Offline learning



Passively observed data
data from π_0

Online iterative learning



Intuition: the new responses and their labels mitigate the OOD issue of proxy reward

Online iterative RLHF with exploration

- For $t = 1,2,3...$ Divide the learning into T batches

  - The main agent **exploits** the historical information: $\pi_t^1 = \pi_{r_{t,\mathrm{MLE}}}$ based on $\mathcal{D}_{1:t-1}$

$$\pi_t^1 = \max_{\pi} \mathbb{E}_{x\sim d_0}\left[\mathbb{E}_{a\sim\pi(\cdot|x)}[r_{t,\mathrm{MLE}}(x,a)] - \eta \mathrm{KL}(\pi(\cdot \mid x), \pi_0(\cdot \mid x))\right].$$

Online iterative RLHF with exploration

- For $t = 1,2,3...$ Divide the learning into T batches
  - The main agent **exploits** the historical information: $\pi_t^1 = \pi_{r_{t,\mathrm{MLE}}}$ based on $\mathcal{D}_{1:t-1}$

$$\pi_t^1 = \max_{\pi} \mathbb{E}_{x\sim d_0}\Big[\mathbb{E}_{a\sim\pi(\cdot|x)}[r_{t,\mathrm{MLE}}(x,a)] - \eta \mathrm{KL}(\pi(\cdot\mid x), \pi_0(\cdot\mid x))\Big].$$

  - The enhancer **explores** the environment by maximizing the uncertainty relative to π_t^1
    - $\pi_t^2 = \arg\max_{\pi'\in\Pi} \Gamma_t(\pi_t^1, \pi')$

Uncertainty estimator

  - Collect m new samples $x_{t,j}, a_{t,j}^1, a_{t,j}^2, y_{t,j} \sim (d_0, \pi_t^1, \pi_t^2, \mathcal{P}_{BT}^\star)$ as $\mathcal{D}_t$

Uncertainty estimator

Definition: uncertainty estimator in linear case

Suppose that $r = \langle \theta, \phi(x,a) \rangle : \theta, \phi(x,a) \in \mathbb{R}^d$. For any two policies π_t^1, π_t^2, we define the information gain as

$$\Gamma_t(\pi_t^1, \pi_t^2) = C_\dagger \| \underbrace{\mathbb{E}_{\pi_t^1}\phi(x, a_t^1) - \mathbb{E}_{\pi_t^2}\phi(x, a_t^2)}_{\text{feature differnece}} \|_{\Sigma_t^{-1}}$$

which is the projection of the new feature difference to historical feature covariance matrix.

$$\Sigma_t = \lambda C_\dagger^2 I + \sum_{s=1}^{t-1} \mathbb{E}_{x\sim d_0, a^1\sim\pi_s^1, a^2\sim\pi_s^2} \big(\phi(x,a^1) - \phi(x,a^2)\big)^\top \big(\phi(x,a^1) - \phi(x,a^2)\big)$$

Theoretical result

Theorem: Guarantee for the online iterative preference learning

If we run the online iterative RLHF with batch size $m = O\left(d/\epsilon^2\right)$ for $T = \tilde{\Omega}(d)$ times, with probability at least $1 - \delta$, we can find a $t_0 \in [T]$ such that

$$J(\pi^\star) - J(\pi_{t_0}^1) + \eta \mathrm{KL}(\pi^\star, \pi_{t_0}^1) \leq \epsilon$$

where $J(\pi) = \mathbb{E}_{d_0,\pi}[r^\star(x,a) - \eta \mathrm{KL}(\pi, \pi_0)]$.

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

Theoretical result

Theorem: Guarantee for the online iterative preference learning

If we run the online iterative RLHF with batch size $m = O\left(d/\epsilon^2\right)$ for $T = \tilde{\Omega}(d)$ times, with probability at least $1 - \delta$, we can find a $t_0 \in [T]$ such that

$$J(\pi^\star) - J(\pi_{t_0}^1) + \eta \mathrm{KL}(\pi^\star, \pi_{t_0}^1) \leq \epsilon$$

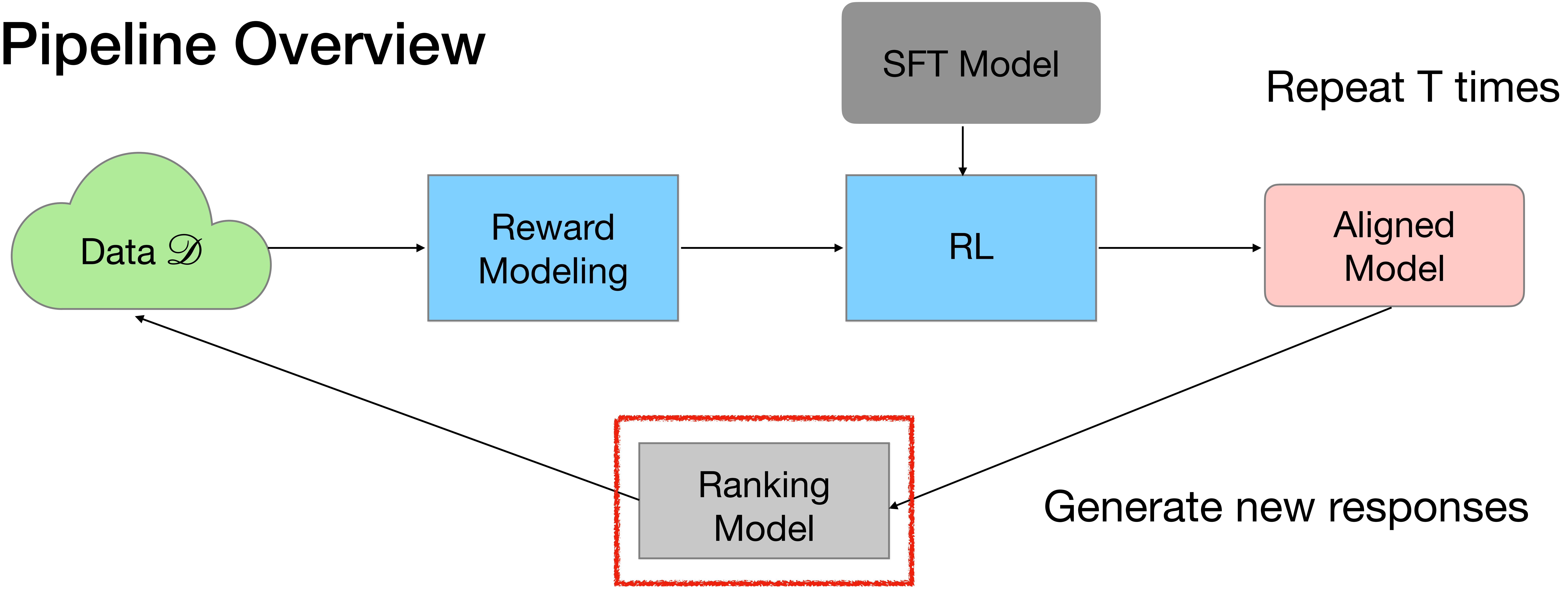
where $J(\pi) = \mathbb{E}_{d_0,\pi}[r^\star(x, a) - \eta \mathrm{KL}(\pi, \pi_0)]$.

- \+ The algorithm is provably efficient
- \- Iterative human feedback is expensive to collect for open-source project
- \- It is not clear how to construct the uncertainty estimator for general neural network

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

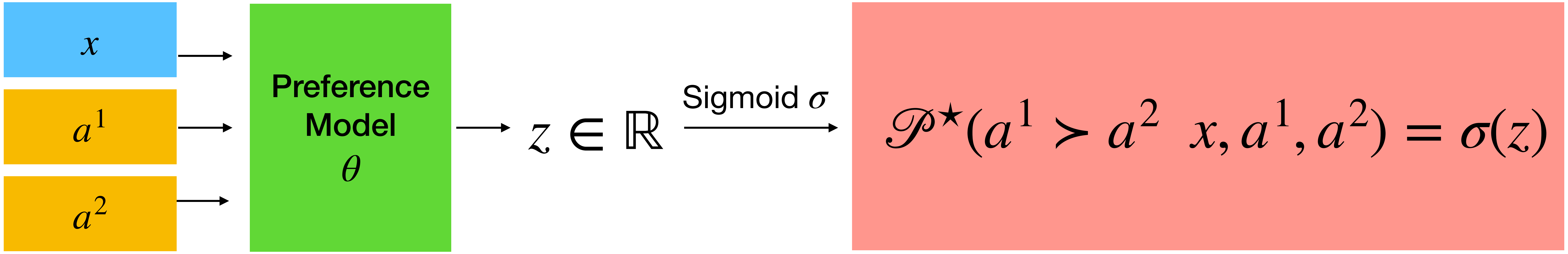
RLHFlow: Open-source Online Iterative RLHF

Pipeline Overview



- A mixture of different types of ranking models on open-source data
- Heuristic rule: length penalty, final result checking for MATH/Coding…

Next-token prediction as pairwise preference model

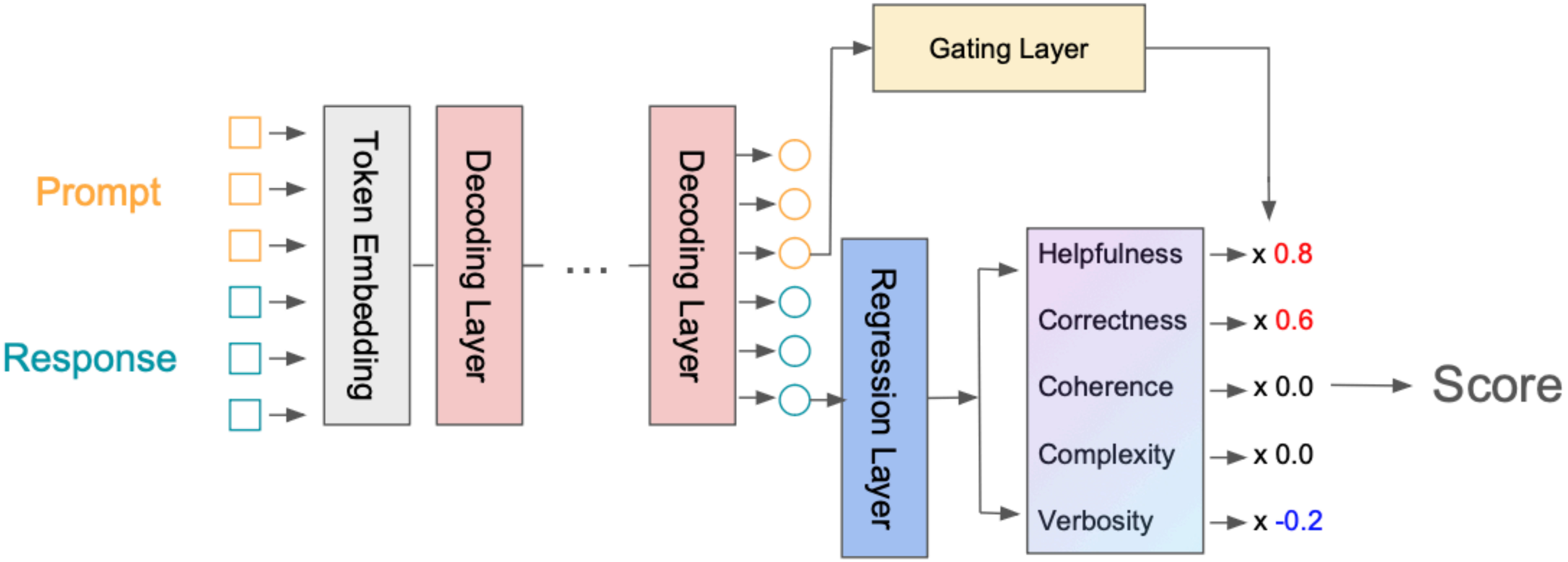


instruction = [CONTEXT] $\{x\}$ [RESPONSE A] $\{a^1\}$ [RESPONSE B] $\{a^2\}$

$$\mathbb{P}(a^1 \succ a^2 \mid x, a^1, a^2) = \mathbb{P}(A \mid \text{instruction})$$

Zhao, Y., Joshi, R., Liu, T., Khalman, M., Saleh, M., & Liu, P. J. (2023). Slic-hf: Sequence likelihood calibration with human feedback. arXiv preprint.

Multi-head reward model with MoE aggregation



- Multi-head reward modeling from different criteria
- MoE aggregation with the coefficient determined by the embedding of the prompt

Wang H*, **Xiong W***, et al. Interpretable Preferences via Multi-Objective Reward Modeling and Mixture-of-Experts. arXiv preprint arXiv:2406.12845, 2024.

Reward modeling: reward benchmark results

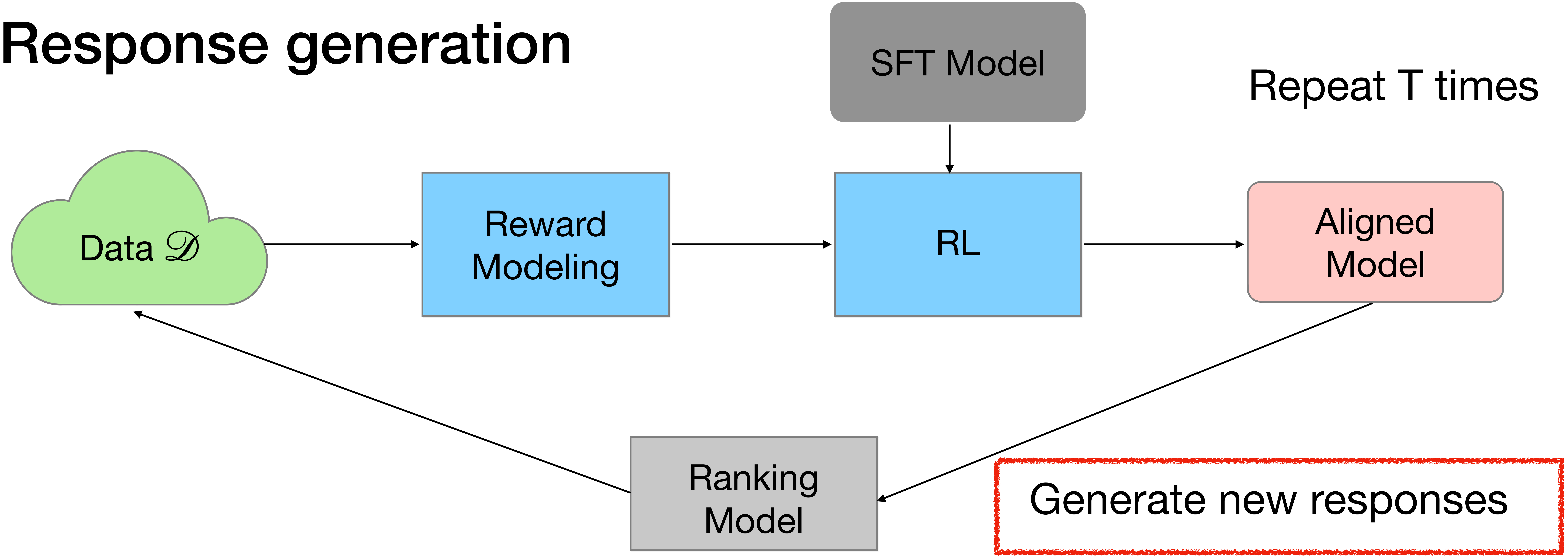
	Model	Model Type	Score	Chat	Chat Hard	Safety	Reasoning
1	nvidia/Nemotron-4-340B-Reward *	Custom Classifier	92.2	95.8	87.1	92.2	93.6
2	RLHFlow/ArmoRM-Llama3-8B-v0.1	Custom Classifier	90.8	96.9	76.8	92.2	97.3
3	Cohere May 2024 *	Custom Classifier	89.5	96.4	71.3	92.7	97.7
4	nvidia/Llama3-70B-SteerLM-RM *	Custom Classifier	89.0	91.3	80.3	93.7	90.6
5	facebook/Self-taught-Llama-3-70B *	Generative	88.7	96.9	84.0	91.5	82.5
6	google/gemini-1.5-pro-0514 *	Generative	88.1	92.3	80.6	87.5	92.0
7	google/flame-1.0-24B-july-2024 *	Generative	88.1	92.2	75.7	90.7	93.8
8	RLHFlow/pair-preference-model-LLaMA3-8B	Custom Classifier	87.1	98.3	65.8	89.7	94.7
9	Cohere March 2024 *	Custom Classifier	87.1	94.7	65.1	90.3	98.2
10	openai/gpt-4o-2024-08-06	Generative	86.7	96.1	76.1	88.1	86.6
11	openai/gpt-4-0125-preview	Generative	85.9	95.3	74.3	87.2	86.9
12	openai/gpt-4-turbo-2024-04-09	Generative	85.1	95.3	75.4	87.1	82.7
13	openai/gpt-4o-2024-05-13	Generative	84.7	96.6	70.4	86.7	84.9

The models serve as the ranking models for 30+ follow-up preference learning research projects.

Screenshot from 8.30, 2024.

Lambert, Nathan, et al. "Rewardbench: Evaluating reward models for language modeling." arXiv preprint 2024.

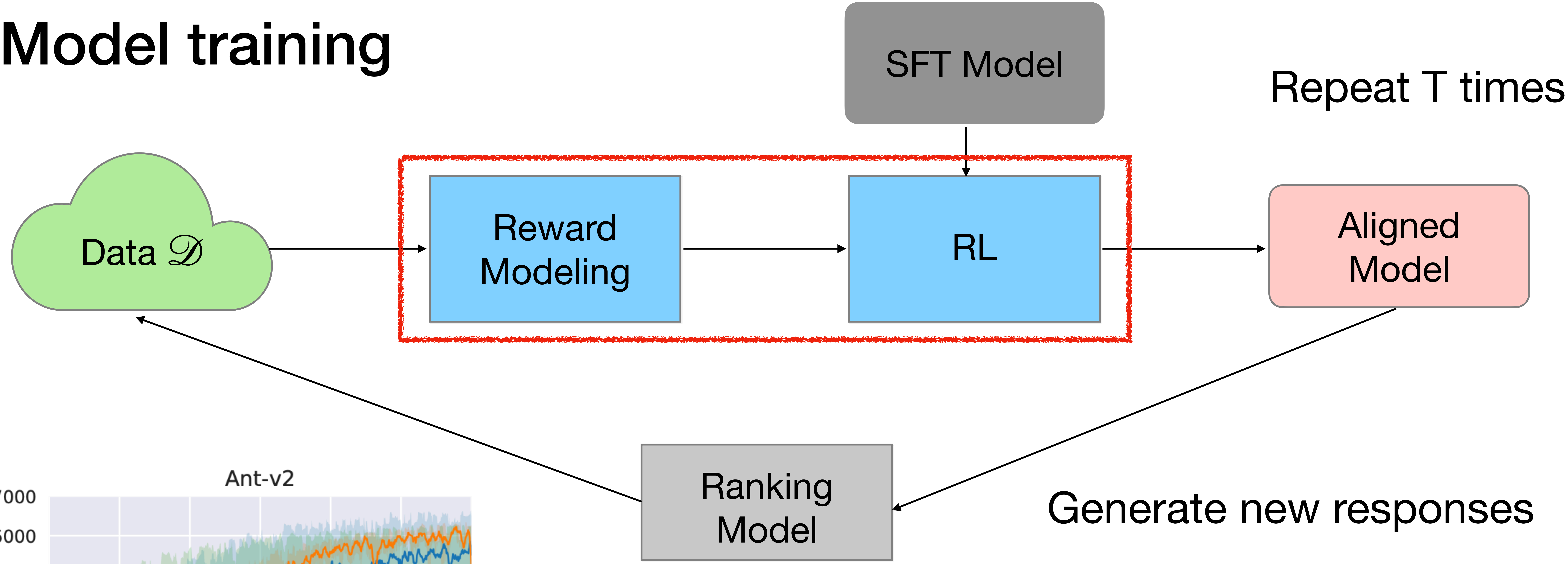
Response generation

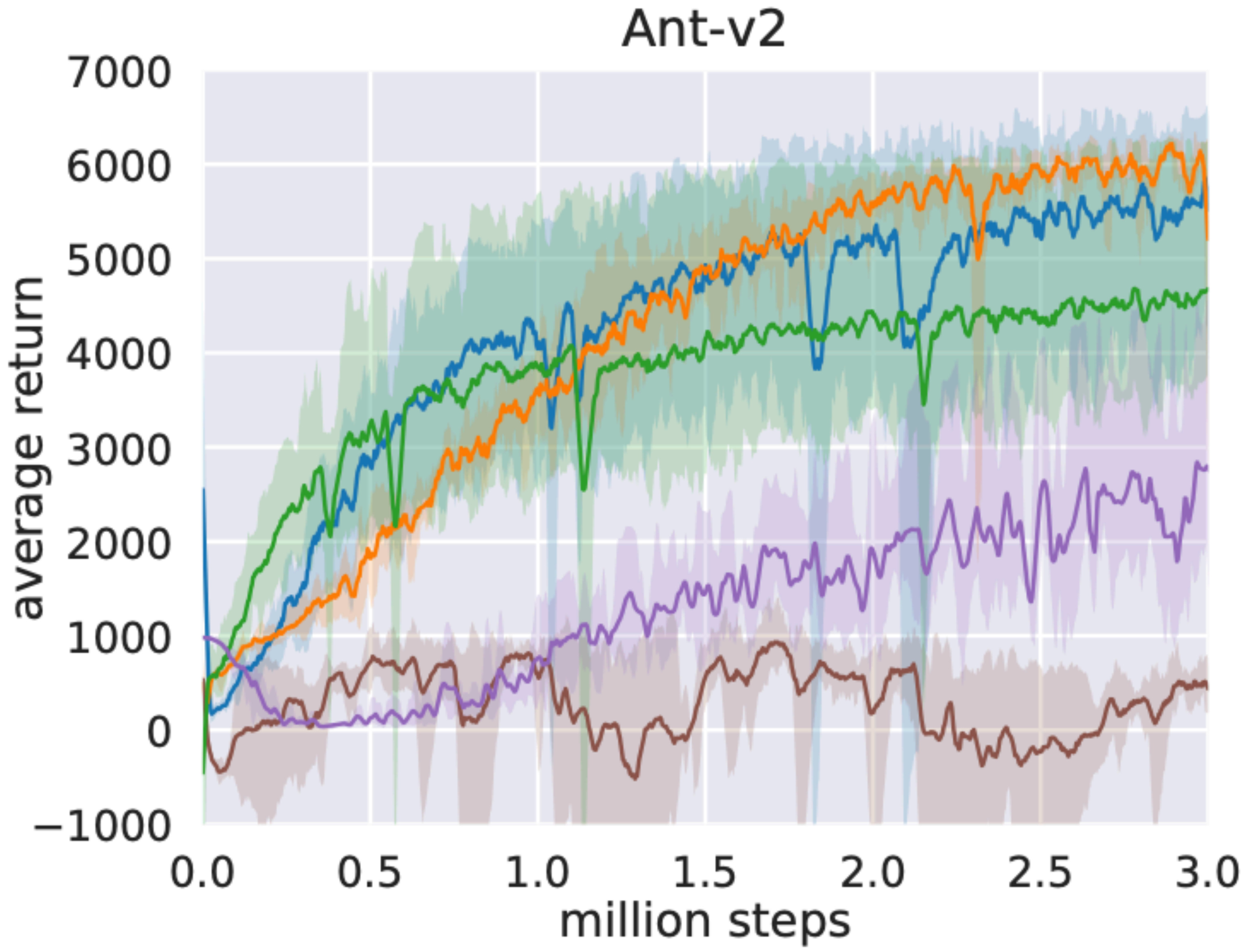


- Heuristic strategies to maximize sample diversity
  - Sample n responses and use the best one and the worst one to construct a pair
  - Tuning sampling parameter like the temperature
  - Collect data by different checkpoints

$$\Gamma_t(\pi_t^1, \pi_t^2) = C_\dagger \| \underbrace{\mathbb{E}_{\pi_t^1}\phi(x, a_t^1) - \mathbb{E}_{\pi_t^2}\phi(x, a_t^2)}_{\text{feature differnece}} \|_{\Sigma_t^{-1}}$$

Model training





Haarnoja, T., Zhou, A., Hartikainen, K., Tucker, G., Ha, S., Tan, J., ... & Levine, S. (2018). Soft actor-critic algorithms and applications. arXiv.

Direct preference optimization (DPO)

- Gibbs distribution

$$\pi_r(\cdot \mid x) = \max_{\pi} \left[\mathbb{E}_{a \sim \pi(\cdot|x)}[r(x,a)] - \eta \mathrm{KL}(\pi(\cdot \mid x), \pi_0(\cdot \mid x)) \right] = \frac{1}{Z(x)} \cdot \pi_0(\cdot \mid x) \cdot \exp\left(\frac{1}{\eta} r(x, \cdot)\right)$$

- Re-parameterize reward by policy:

$$Z(x) = \sum_{a \in \mathcal{A}} \pi_0(a \mid x) \cdot \exp\left(\frac{1}{\eta} r(x,a)\right)$$

$$r(x,a) = \underbrace{\eta \log \frac{\pi_r(a \quad x)}{\pi_0(a \quad x)}}_{\text{Implicit reward}} + \cancel{\eta \log Z(x)}$$

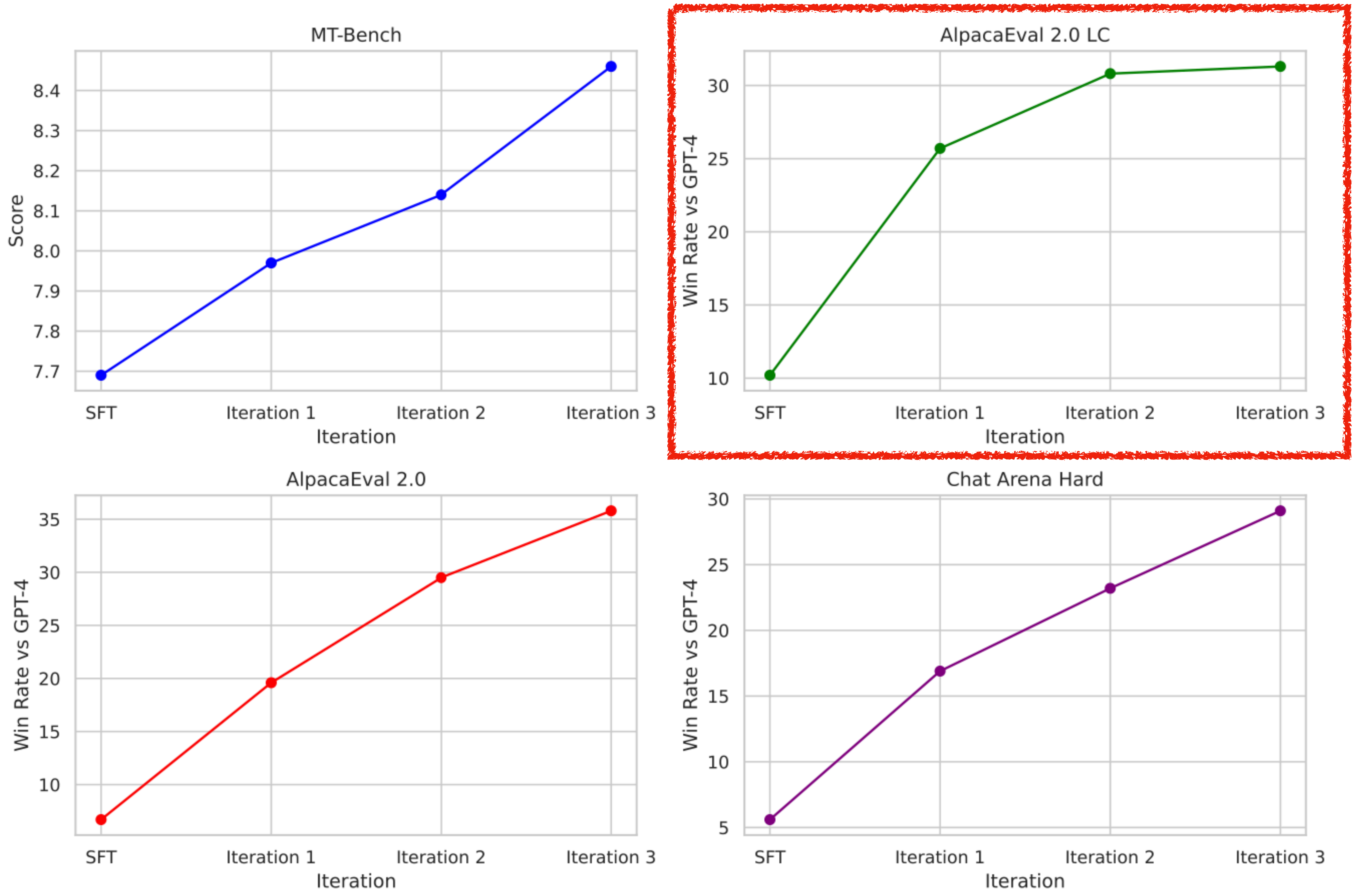
- MLE in reward space -> policy optimization:

$$\ell_{\mathrm{reward}}(r_\theta) = \sum_{(x,a^w,a^l) \in \mathcal{D}} \log\left(\sigma\left(r_\theta(x,a^w) - r_\theta(x,a^l)\right)\right)$$

$$\mathcal{L}_{\mathsf{DPO}}(\pi_\theta) = -\sum_{(x,a^w,a^l) \in \mathcal{D}} \log \sigma\left(\eta\left[\log \frac{\pi_\theta(a^w \mid x)}{\pi_0(a^w \mid x)} - \log \frac{\pi_\theta(a^l \mid x)}{\pi_0(a^l \mid x)}\right]\right).$$

Rafailov, Rafael, et al. Direct preference optimization: Your language model is secretly a reward model. NeurIPS, 2023.

Main result: state-of-the-art chat model

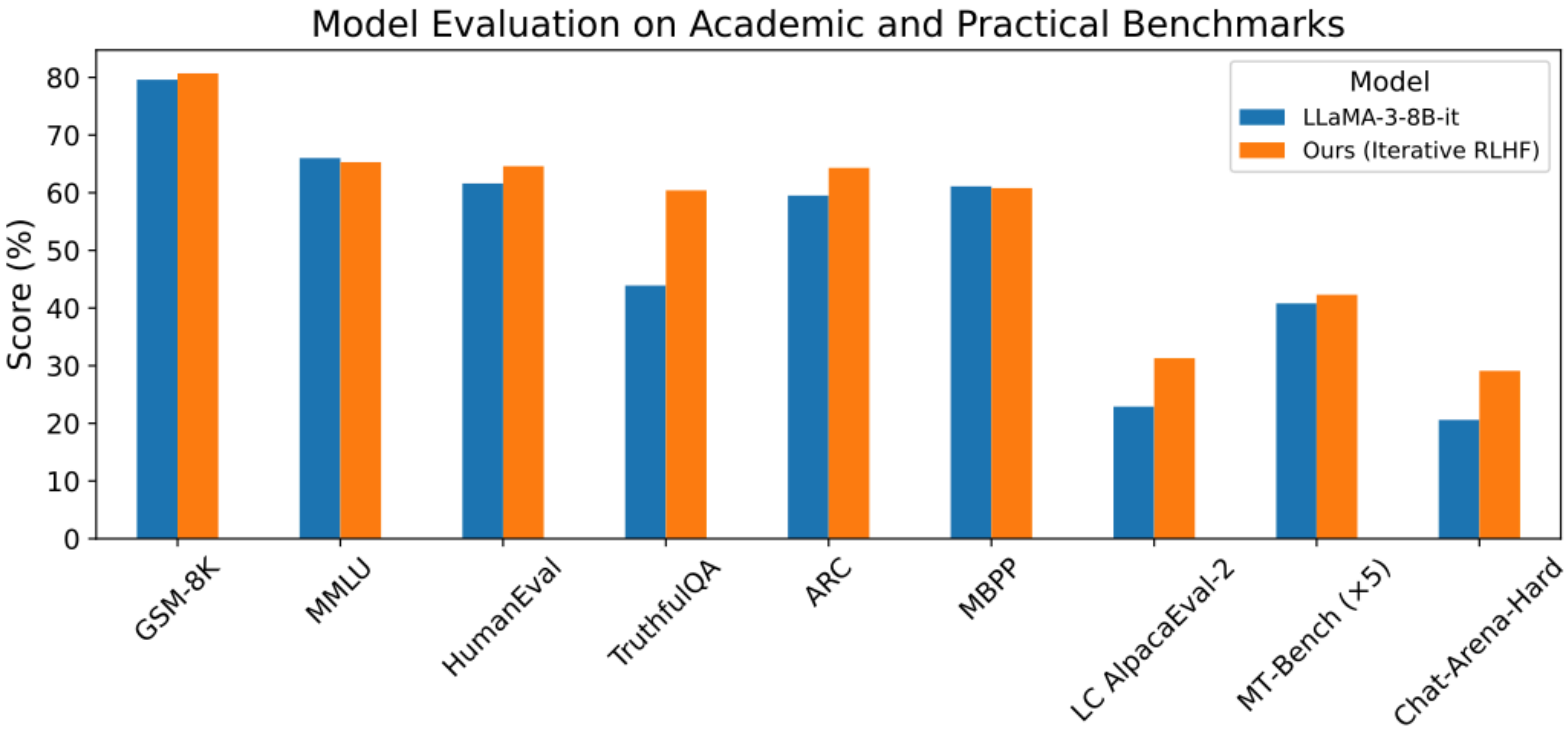


Base model: LLaMA3-8B-SFT

Evaluation results of models at different iterations on instruction-following benchmarks.

(α-β) Dong H, **Xiong W**, Pang B, Wang H, et al. RLHF workflow: From reward modeling to online RLHF, TMLR, 2024.

Main result: state-of-the-art chat model



Evaluation results on standard academic and instruction-following benchmarks.

(α–β) Dong H, **Xiong W**, Pang B, Wang H, et al. RLHF workflow: From reward modeling to online RLHF, TMLR, 2024.

Multi-turn tool-integrated reasoning

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 tool-integrated reasoning

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.

```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)

Multi-turn tool-integrated reasoning

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.

```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)

$$s_h = (x, a_1, o_1, a_2, o_2, \cdots, a_{h-1}, o_{h-1})$$

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

Preference learning for multi-turn reasoning task

- Trajectory: x-prompt, a-action, o-external observation, s_h: history at the beginning of step h

$$\tau = (x, \underbrace{a_1, o_1, \cdots, o_{H-1}, a_H}_{y}) .$$

- Trajectory preference

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

- Learning objective

$$\arg\max_{\pi} J(\pi; \mathcal{M}^{\star}, \pi_0) = \mathbb{E}_{x \sim d_0} \mathbb{E}_{a_h \sim \pi_h(\cdot|s_h), o_h \sim \mathbb{P}_h(\cdot|s_h, a_h)} \left[r^{\star}(x, y) - \eta \sum_{h=1}^{H} \mathrm{KL}\big(\pi_h(\cdot \mid s_h), \pi_{0,h}(\cdot \mid s_h)\big) \right] .$$

Multi-turn direct preference learning

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

$$r(s_H, a_H) = \underbrace{\eta \sum_{h=1}^{H} \log \frac{\pi_{\mathcal{M},h}(a_h \mid s_h)}{\pi_{0,h}(a_h \mid s_h)}}_{\text{term (A)}} + \cancel{\underbrace{Z(x)}_{\text{term (B)}}} + \underbrace{\sum_{h=1}^{H-1} \left[V_{\mathcal{M},h+1}(s_{h+1}) - \mathbb{E}_{o_h \sim \mathbb{P}_h(\cdot \mid s_h, a_h)} V_{\mathcal{M},h+1}(s_{h+1}) \right]}_{\text{term (C)}}.$$

- Term (C) is not zero except for

$V_{\mathcal{M},h}$: optimal V value function under $\mathcal{M} = (r, \mathbb{P})$

  - $H = 1$: original DPO

  - o_h is deterministic given the history

Implementation: run DPO but mask out the external messages.

Xiong W, Shi C, Shen J, et al. Building Math Agents with Multi-Turn Iterative Preference Learning, arXiv, 2024.

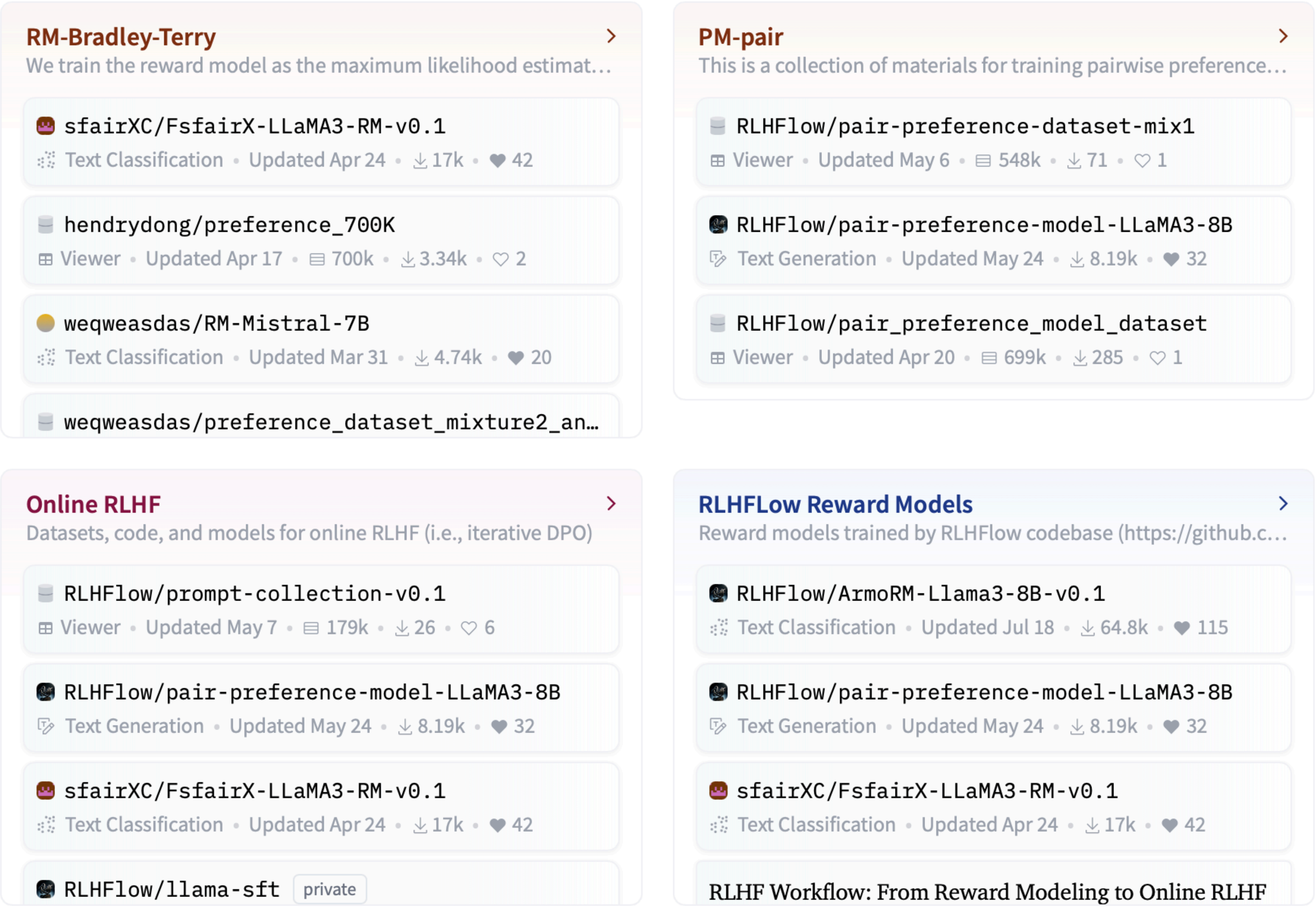
Main result: improving reasoning ability

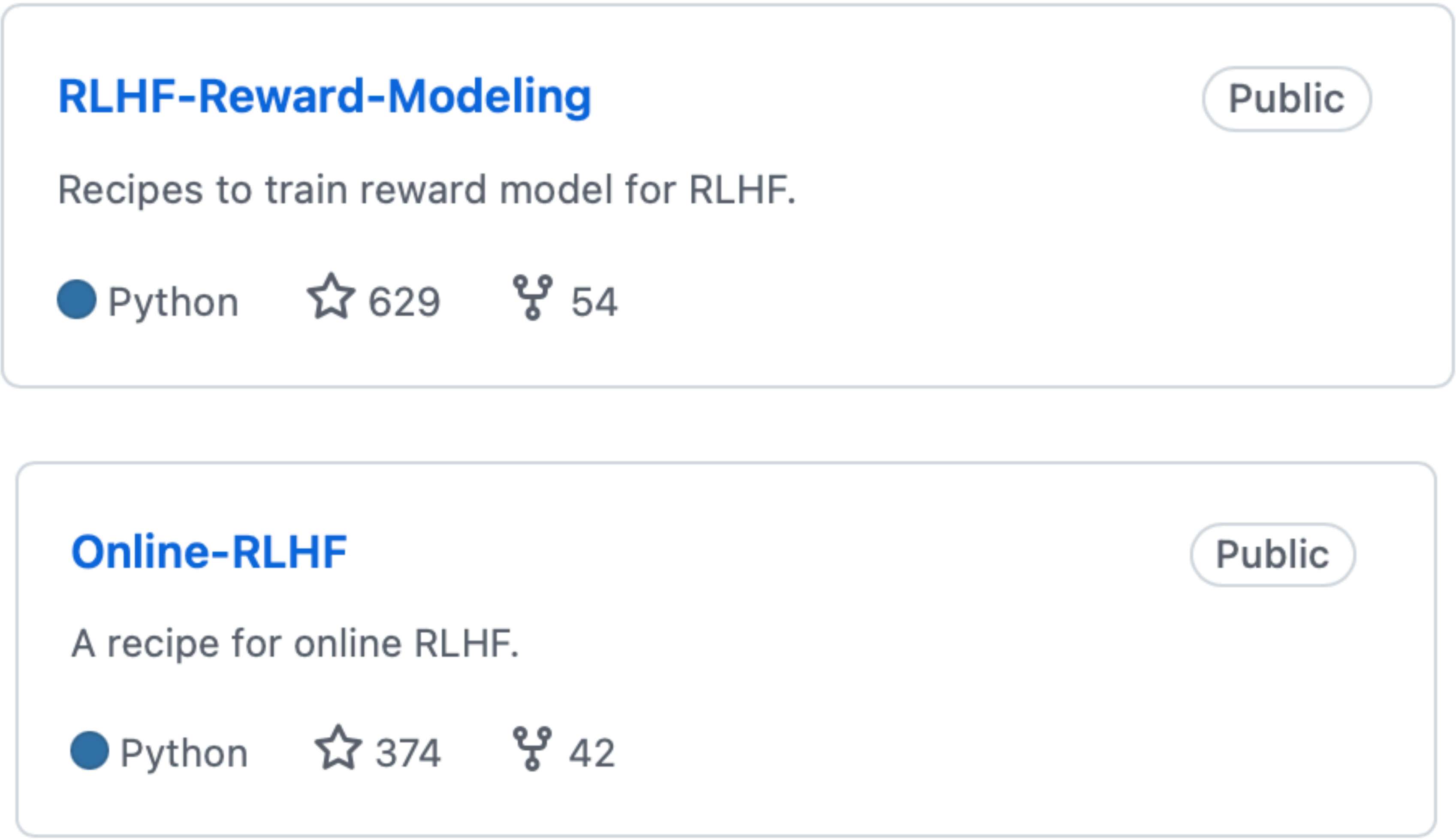
Base Model	Method	with Tool	GSM8K	MATH	AVG
Gemma-1.1-it-7B	SFT†	✓	77.5	46.1	61.8
Gemma-1.1-it-7B	RAFT	✓	79.2	47.3	63.3
Gemma-1.1-it-7B	Iterative Single-turn DPO	✓	81.7	48.9	65.3
Gemma-1.1-it-7B	Iterative M-DPO + fixed reference	✓	79.9	48.0	64.0
Gemma-1.1-it-7B	M-DPO Iteration 1	✓	81.5	49.1	65.3
Gemma-1.1-it-7B	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
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 M-DPO	✓	81.5 ↑4.2	50.1 ↑3.7	65.8
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 M-DPO	✓	82.3 ↑4.5	47.5 ↑4.8	64.9
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

Prompt: training set MATH and GSM8K
Reward: binary reward by checking the answer

Xiong W, Shi C, Shen J, et al. Building Math Agents with Multi-Turn Iterative Preference Learning, arXiv, 2024.

A practical and open-source codebook





Dataset

Training code

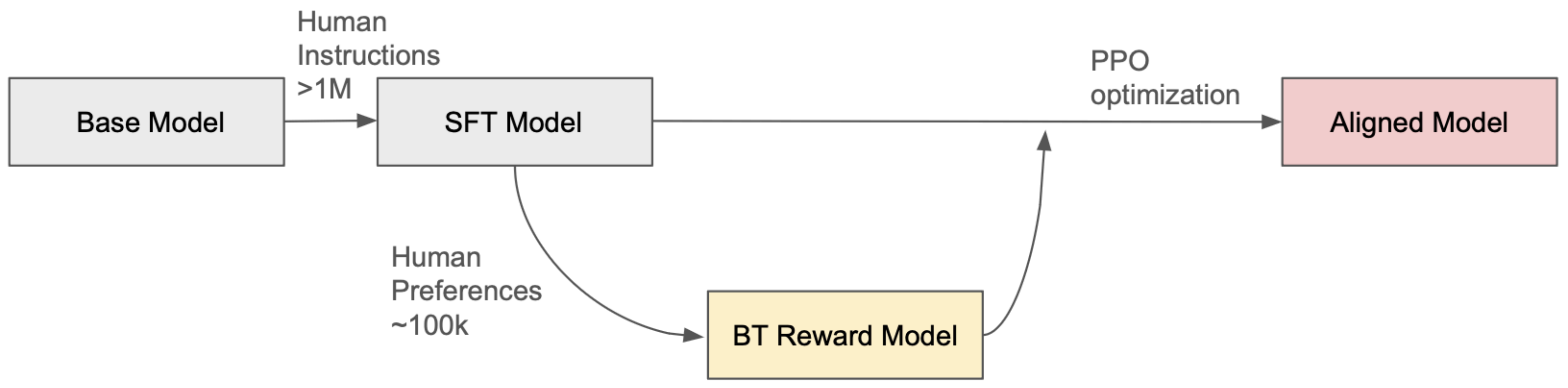
Hyper-parameters

Final models

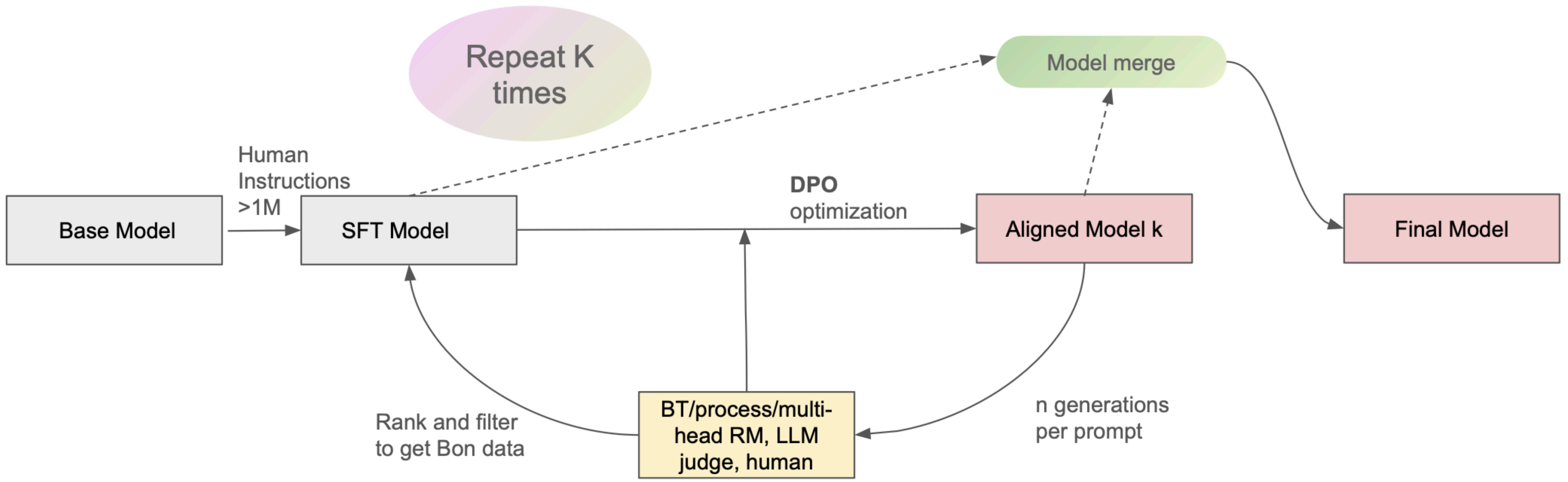
RLHFlow, Github.

Takeaway

- RLHF benefits from continuous online exploration through interactions with the rater
- Online iterative direct preference learning is a robust recipe to make good chatbot



Instruct GPT



LLaMA 3.1

Thanks for listening!