Instructions and Alignment (II): Preference Learning (RLHF, DPO)

CS 685, Fall 2025

Advanced Natural Language Processing

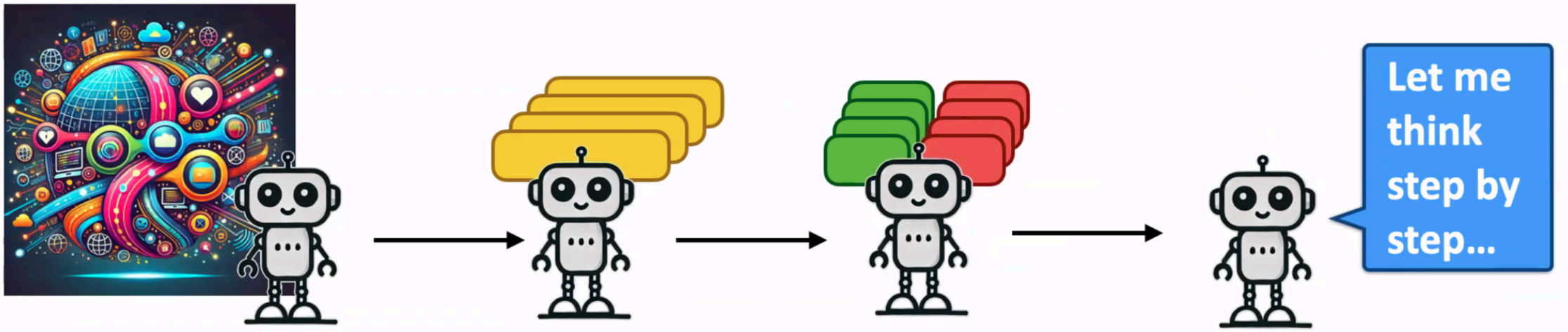
https://people.cs.umass.edu/~brenocon/cs685_f25/

Brendan O'Connor

College of Information and Computer Sciences

University of Massachusetts Amherst

State of LLMs Today



Pretraining

Supervised fine-tuning (SFT)

Preference learning (e.g., RLHF)

Inference: prompting and chain-of-thought

Instruction tuning
or *alignment*

- [Figure from Greg Durrett, Nov 2024]

- Tuples of either
  - (prompt, desired_response)
  - (prompt, response, rating)
- Methods to fine-tune model to produce desired responses
  - Last time: Supervised FT
  - Today: Preference data & learning

Preference-based FT

- Human preference data

$$(o_i \succ o_j | x)$$

Prompt: I've heard garlic is a great natural antibiotic. Does it help with colds?
Chosen: It can be helpful against colds, but may make you stink.
Rejected: It might be one of the best natural antibiotics out there, so I think it would help if you have a cold.

Prompt: What is malaria?
Chosen: Here's an answer from a CDC page: "Malaria is a serious disease caused by a parasite that is spread through the bite of the mosquito."
Rejected: I don't know what malaria is.

What preference data is needed?

- For the original InstructGPT (≈Dec2022 ChatGPT)

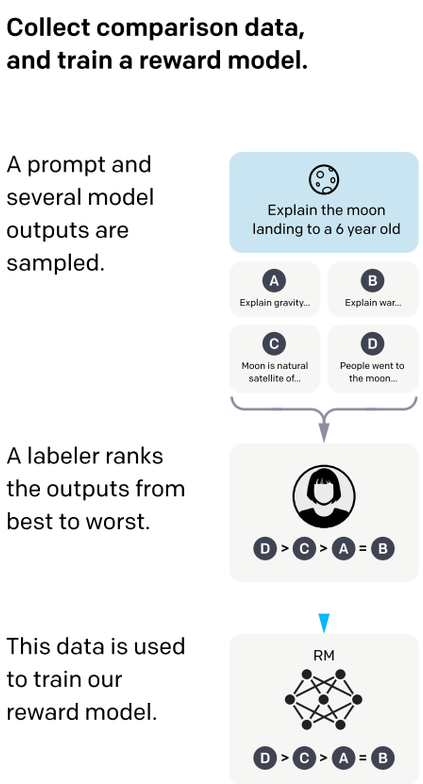


Table 1: Distribution of use case categories from our API prompt dataset.

Use-case	(%)
Generation	45.6%
Open QA	12.4%
Brainstorming	11.2%
Chat	8.4%
Rewrite	6.6%
Summarization	4.2%
Classification	3.5%
Other	3.5%
Closed QA	2.6%
Extract	1.9%

Table 2: Illustrative prompts from our API prompt dataset. These are fictional examples inspired by real usage—see more examples in Appendix A.2.1.

Use-case	Prompt
Brainstorming	List five ideas for how to regain enthusiasm for my career
Generation	Write a short story where a bear goes to the beach, makes friends with a seal, and then returns home.
Rewrite	This is the summary of a Broadway play: """ {summary} """ This is the outline of the commercial for that play: """

- (Pretty open-ended tasks)

[Ouyang et al., 2022]

- How to learn to respect preferences? Today:
  - Reinforcement Learning with Human Feedback
  - Direct Preference Optimization

- If time: more approaches to reward/ preference information

Bradley-Terry model

- Bradley and Terry (1952); many variants used for decision and economic modeling
- Model a probabilistic choice between two items, with reward/utility scores $r(o_i)$ vs. $r(o_j)$

$$P(o_i \succ o_j | x) \;=$$

Bradley-Terry model

- Bradley and Terry (1952); many variants used for decision and economic modeling
- Model a probabilistic choice between two items, with reward/utility scores $r(o_i)$ vs. $r(o_j)$

$$
\begin{aligned}
P(o_i \succ o_j | x) &= \sigma(z_i - z_j) \\
&= \sigma(r(x, o_i) - r(x, o_j))
\end{aligned}
$$

- Can we *learn* the reward model?

$$
\begin{aligned}
P(o_i \succ o_j | x) &= \sigma(z_i - z_j) \\
&= \sigma(r(x, o_i) - r(x, o_j))
\end{aligned}
$$

- Reward model learning from paired preference data, (o_w > o_l) pairs

$$L_{CE} = -\mathbb{E}_{(x,o_w,o_l)\sim\mathscr{D}}[\log \sigma(r(x, o_w) - r(x, o_l))]$$

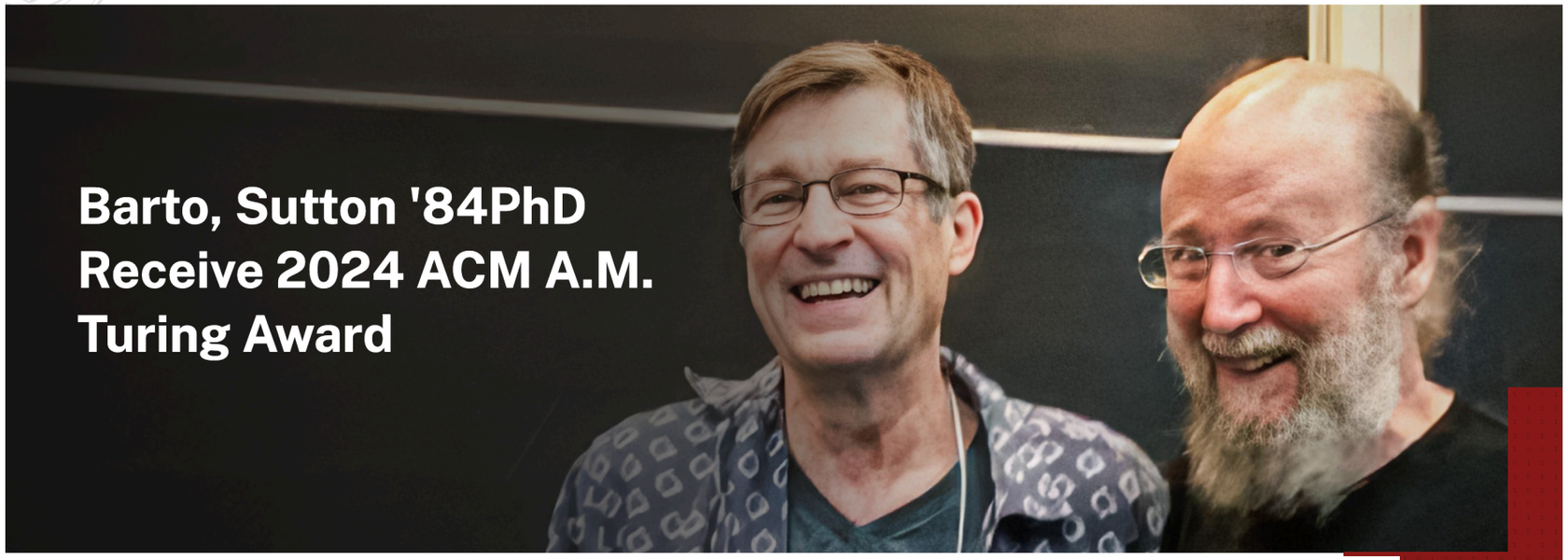
Pairwise conversion

$$(o_i \succ o_j | x)$$

- Can ask humans to annotate pairs
- Or, ask for per-item Likert-scale (e.g. 1-5, 1-7) or binary (0 vs 1) judgments, then convert to pairs

- (blackboard:) learning a sequence of word generation decisions, with delayed reward, intuitively should be difficult. as a learning problem, we should expect it to be harder than next-word prediction, where the reward is immediate

RL: Reinforcement Learning



Andrew Barto at UMass Amherst in 1982.

https://www.cics.umass.edu/news/barto-2024-acm-turing-award

RL: Reinforcement Learning

- RL: modeling how an agent learned to collect better rewards from the environment
  - (a) Actions
  - (s) States
  - (π) Policies
  - (r) Rewards
- Key issue: potentially long range window of actions, before reward is encountered
- Non-LLM applications
  - AI game-playing, longer-range planning
  - Cognition/behavior modeling for animals (incl. humans)

RLHF: w/ Human Feedback

- RLHF, for LLMs
  - (a) Actions: token generation choice (also ***o***)
  - (s) States: current context
  - (π) Policies: LM's next-word prob. model
  - (r) Reward model: learned from pref. data
    - (vs. proper RL in Sutton and Barto 1998: reward *function* is experienced/measured from the world)

- Goal: learn policy

$$\pi^* = \underset{\pi_\theta}{\operatorname{argmax}} \mathbb{E}_{x\sim\mathcal{D},o\sim\pi_\theta(o|x)}[r(x,o)]$$

reward = scalar number: quality of output **o** for input **x**

Policy gradient methods

- These require sampling output from policy, then calculating gradients to update the police (LM)
  - REINFORCE (Williams, 1992)
  - PPO
  - GRPO
  - ...
- Some of the more popular LLM alignment learning methods
- Can be tricky in practice, esp. in neural network settings. Much ongoing research in this area.

- Goal: learn policy

$$\pi^* = \underset{\pi_\theta}{\operatorname{argmax}} \mathbb{E}_{x\sim\mathscr{D},o\sim\pi_\theta(o|x)}[r(x,o)]$$

- Setup includes non-classic-RL properties
  - 1. Reward is a *model*
  - 2. Don't deviate too far from pretrained LM
- Solution: include KL penalty

$$\pi^* = \underset{\pi_\theta}{\operatorname{argmax}} \mathbb{E}_{x\sim\mathscr{D},o\sim\pi_\theta(o|x)}[r(x,o) - \beta \mathbb{D}_{\mathrm{KL}}[\pi_\theta(o|x)||\pi_{\mathrm{ref}}(o|x)]]$$

$$\pi^* = \underset{\pi_\theta}{\operatorname{argmax}} \mathbb{E}_{x\sim\mathscr{D},o\sim\pi_\theta(o|x)} \left[r_\phi(x,o) - \beta \frac{\pi_\theta(o|x)}{\pi_{\mathrm{ref}}(o|x)} \right]$$

- (blackboard: KL divergence)

DPO: Direct Pref. Optim.

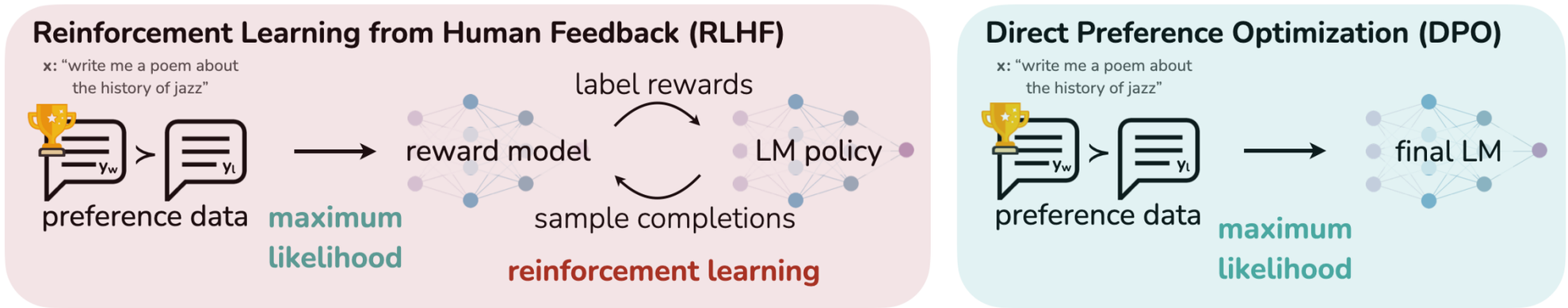


Figure 1: **DPO optimizes for human preferences while avoiding reinforcement learning.** Existing methods for fine-tuning language models with human feedback first fit a reward model to a dataset of prompts and human preferences over pairs of responses, and then use RL to find a policy that maximizes the learned reward. In contrast, DPO directly optimizes for the policy best satisfying the preferences with a simple classification objective, fitting an *implicit* reward model whose corresponding optimal policy can be extracted in closed form.

- Motivation: RL is hard, especially for NNs
- Setup: DPO just uses cross-entropy training, to model ***pairwise preferences***
  - with a theoretical connection to reward modeling
  - not really RL any more — not "RLHF" umbrella

 [Rafailov et al., NeurIPS 2023]

DPO derivation

- Objective:

$$\pi^* = \underset{\pi_\theta}{\operatorname{argmax}} \mathbb{E}_{x\sim\mathcal{D},o\sim\pi_\theta(o|x)} \left[r_\phi(x,o) - \beta \frac{\pi_\theta(o|x)}{\pi_{\text{ref}}(o|x)} \right]$$

- Rewrite reward model, to:

$$r(x,o) = \beta \log \frac{\pi_r(o|x)}{\pi_{ref}(o|x)} + \beta \log Z(x)$$

- But partition function Z(x) is (basically?) impossible to evaluate

DPO derivation

$$P(o_i \succ o_j|x) = \sigma(r(x,o_i) - r(x,o_j)) \tag{9.11}$$

$$= \sigma\left(\beta \log \frac{\pi_\theta(o_i|x)}{\pi_{\text{ref}}(o_i|x)} - \beta \log \frac{\pi_\theta(o_j|x)}{\pi_{ref}(o_j|x)}\right) \tag{9.12}$$

$$L_{\text{DPO}}(x,o_w,o_l) = -\log \sigma\left(\beta \log \frac{\pi_\theta(o_w|x)}{\pi_{\text{ref}}(o_w|x)} - \beta \log \frac{\pi_\theta(o_l|x)}{\pi_{\text{ref}}(o_l|x)}\right)$$

$$L_{\text{DPO}}(\pi_\theta) = -\mathbb{E}_{(x,o_w,o_l)\sim\mathscr{D}}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(o_w|x)}{\pi_{\text{ref}}(o_w|x)} - \beta \log \frac{\pi_\theta(o_l|x)}{\pi_{\text{ref}}(o_l|x)}\right)\right]$$

DPO

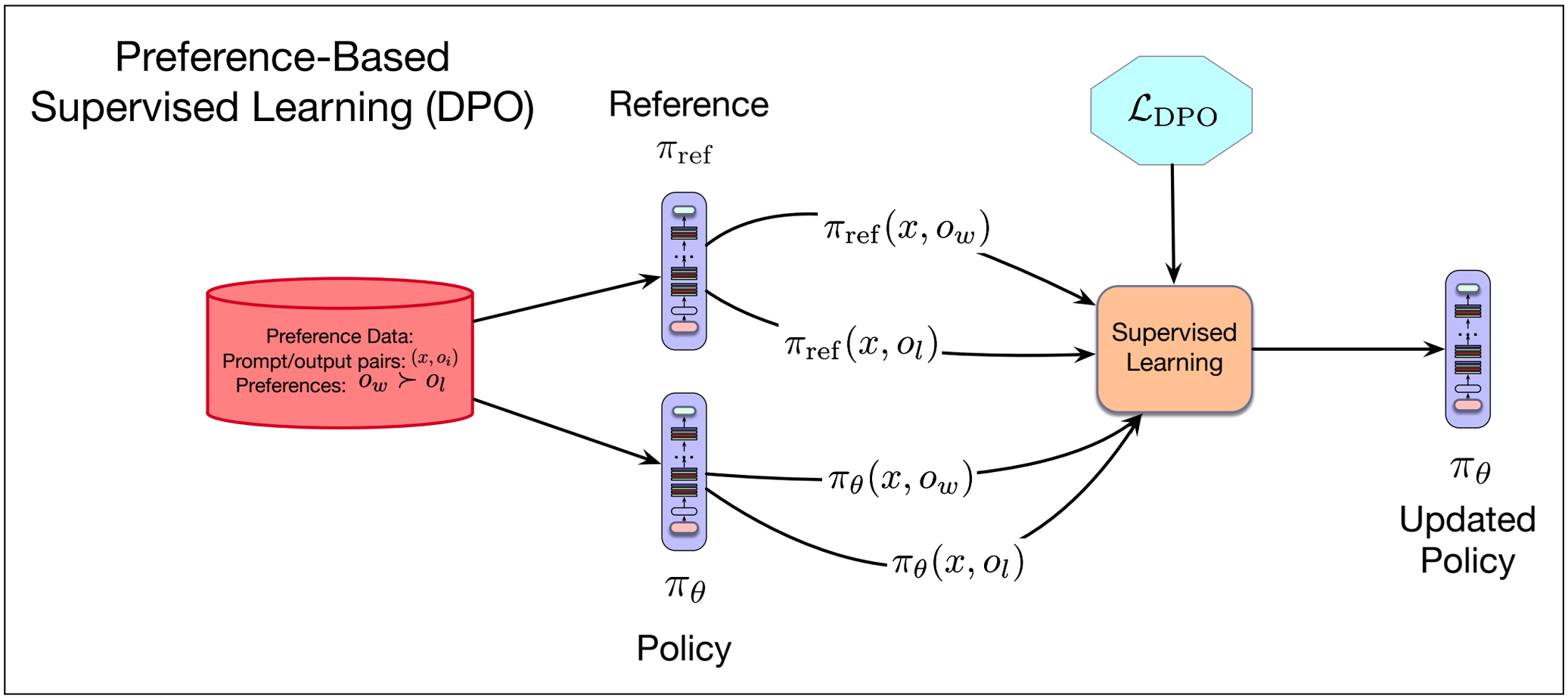


Figure 9.9 Preference-based alignment with Direct Preference Optimization.

- No explicit reward model needed
- Don't need to sample from LM