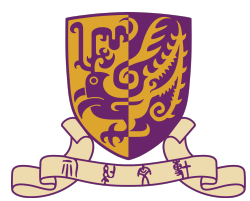
RL vs. DPO in RLHF

FONG, Shi Yuk

Email: syfong1@cse.cuhk.edu.hk

The Institute of Theoretical Computer Science and Communications
The Chinese University of Hong Kong

August 9, 2024



Selected topics in RLHF

- Dealing with reward collapse in RLHF
- **Direct preference optimization (DPO)**
- Proximal policy optimization (PPO)
- Fine-grained reward model

Outline

1. Reviewing RLHF
2. Direct preference optimization (DPO)
3. RL vs. DPO in RLHF

Part 1. Reviewing RLHF

Models trained at RLHF

Let $\mathcal{D}_{\text{pref}} = \{(x^{(i)}, y_+^{(i)}, y_-^{(i)})\}_{i=1}^N$, RLHF aims to learn a reward model $r_\phi(x, y)$ first and utilize it to fine-tune the policy π_θ with a RL process.

$$\phi^* = \arg\min_\phi -\mathbb{E}_{(x,y_+,y_-)\sim\mathcal{D}_{\text{pref}}}[\log \sigma(r_\phi(x, y_+) - r_\phi(x, y_-))]$$

$$\theta^* = \arg\max_\theta \mathbb{E}_{x\sim\rho}[\mathbb{E}_{y\sim\pi_\theta(\cdot|x)}[r_\phi(x, y)] - \beta\mathbb{D}_{\text{KL}}(\pi_\theta \,||\, \pi_{\text{ref}})]$$

where $\mathbb{D}_{\text{KL}}$ is the KL divergence between learned distribution π_θ and a reference distribution π_{ref}.

Part 2. Direct preference optimization (DPO)

R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn, "Direct Preference Optimization: Your Language Model is Secretly a Reward Model," in (Oral Presentation) Advances in Neural Information Processing Systems, 2023, vol. 36, pp. 53728-53741.

Why DPO?

Training a reward model r_ϕ is time-consuming, so does the RL process. It would be beneficial if we can directly optimize the policy π_θ with the preference data $\mathcal{D}_{\text{pref}}$. Such method is called **reward-model-free policy optimization** (as opposed to the reward-model-based p.o.).

Idea: parameterize the reward model to turn the problem into a simple classification problem.

DPO

- Intractable **closed-form** optimal RLHF policy:

$$\pi_\theta(y \mid x) = \frac{1}{Z(x)} \pi_{\text{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right)$$

Every **reward function** r induces an **optimal policy** π_θ.

- Another view of this identity (reparameterization):

$$r_\pi(x, y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x)$$

For all **policy** π_θ, there exists an **induced reward function** r_π, such that π_θ is the **optimal policy**.

- Observation: bijection between reward functions r_π and optimal policies π_θ.
- Idea: Train π_θ so that r_π fits human preferences!

DPO

- Intractable **closed-form** optimal RLHF policy:

$$\pi_\theta(y \mid x) = \frac{1}{Z(x)} \pi_{\text{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right)$$

Every **reward function** r induces an **optimal policy** π_θ.

- Another view of this identity (reparameterization):

$$r_\pi(x, y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x) \quad \longleftarrow$$

but this is computationally infeasible!
(summing over all possible sequences)

For all **policy** π_θ, there exists an **induced reward function** r_π, such that π_θ is the **optimal policy**.

- Observation: bijection between reward functions r_π and optimal policies π_θ.
- Idea: Train π_θ so that r_π fits human preferences!

DPO: the overall framework

Observe that the reward modeling loss on preference data is built upon difference in rewards:

$$\mathcal{L}_{RM}(r, \mathcal{D}) = -\mathbb{E}_{(x,y_+,y_-)\sim\mathcal{D}}[\log \sigma(r_\phi(x, y_+) - r_\phi(x, y_-))]$$

The **induced reward difference** is:

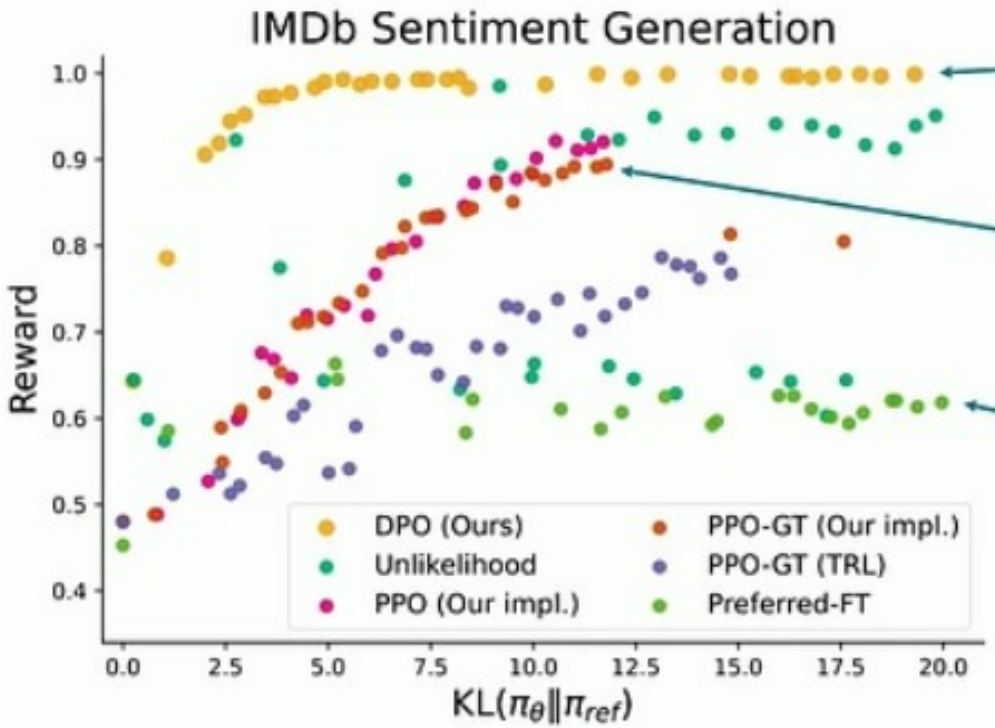
$$r_{\pi_\theta}(x, y_+) - r_{\pi_\theta}(x, y_-) = \beta \left(\log \frac{\pi_\theta(y_+ \mid x)}{\pi_{\text{ref}}(y_+ \mid x)} - \log \frac{\pi_\theta(y_+ \mid x)}{\pi_{\text{ref}}(y_+ \mid x)} \right)$$

We can see that the intractable term $\log Z(x)$ cancels out in the difference.

Simplified loss for the RLHF objective:

$$\boxed{\mathcal{L}_{DPO}(\pi_\theta, \mathcal{D}) = -\mathbb{E}_{(x,y_+,y_-)\sim\mathcal{D}}[\log \sigma(r_{\pi_\theta}(x, y_+) - r_{\pi_\theta}(x, y_-))]}$$

Experiments: Reward-KL trade-off



DPO provides the **best reward-KL tradeoff**

PPO **doesn't provide optimal reward**, even when provided with ground truth

DPO provides a **substantial improvement over** preferred-FT

Experiments: Benchmark

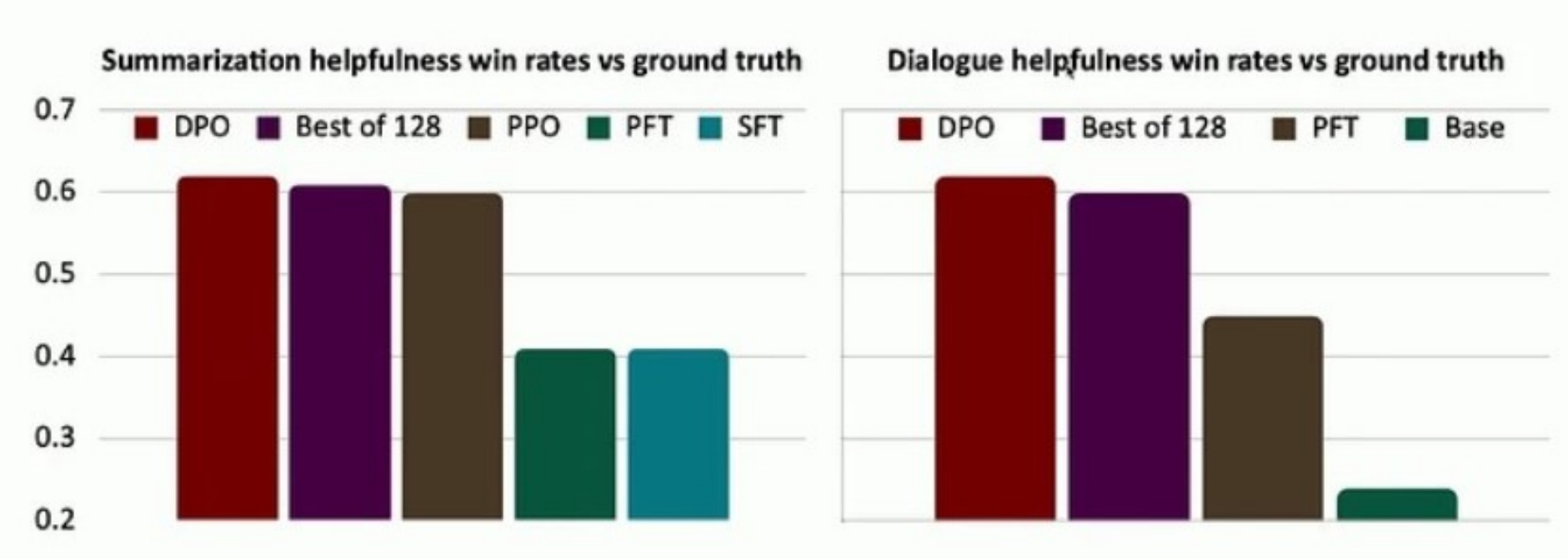


Figure: DPO performs similarly to other RL-based methods, while being **substantially simpler, computationally cheaper, and stabler**.

Part 3. RL vs. DPO in RLHF

Z. Li, T. Xu, and Y. Yu, "Policy Optimization in RLHF: The Impact of Out-of-preference Data," in (Oral Presentation) The Second Tiny Papers Track at International Conference on Learning Representations (ICLR), 2024

Comparison: DPO vs. RL in RLHF

DPO significantly improves the efficiency of RLHF, but at what cost?

There are no free lunches!

Main Idea

- DPO is inferior to RL when the representation is misspecified;
- Sufficient Online Update is Crucial for RL (Online RLHF)

Facts

- (Preference) data is expensive and limited. Thus we can only approximate the reward function in RLHF;
- Distribution of prompts $\rho(\cdot)$ is unknown, we can only obtain finite samples from it;
- Taking expectation over policy distribution $\mathbb{E}_{y\sim\pi(\cdot|x)}[\cdot]$ is infeasible, we can only obtain finite samples from π.

DPO is inferior to RL when the representation is misspecified

Reward learned in RLHF:

$$\hat{r} = \arg\min_{r \in \mathcal{R}} -\mathbb{E}_{(x,y_+,y_-)\sim\mathcal{D}_{\text{pref}}}[\log \sigma(r(x, y_+) - r(x, y_-))]$$

where we hope $\mathcal{R} = \{r : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}\}$ contains the optimal reward function.

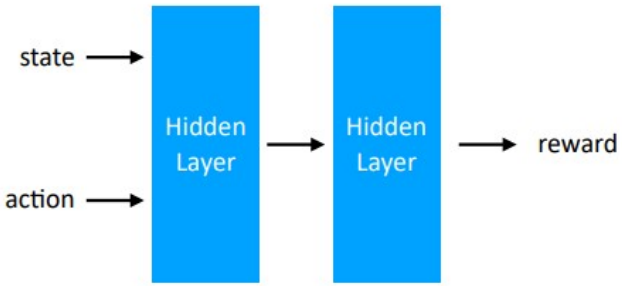
(Latent) reward learned in DPO:

$$\mathcal{R}_{\text{DPO}} = \{r : \mathcal{X} \times \mathcal{Y} \to \mathbb{R} : r \propto \log \pi_\theta(y \mid x) - \log \pi_{\text{ref}}(y \mid x)\}$$

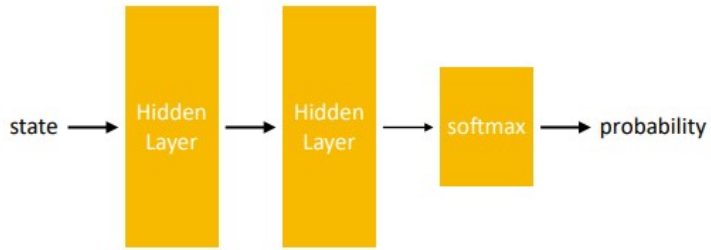
Reward quality of DPO is limited by the expressiveness of learned policy model. If $\mathcal{R}_{\text{DPO}} \neq \mathcal{R}$, the reward and associated optimal policy in DPO may be of poor quality.

Structure of reward and policy models differ

More details can be found in **Theorem 1**, *T. Korbak, H. Elsahar, G. Kruszewski, and M. Dymetman, "On Reinforcement Learning and Distribution Matching for Fine-Tuning Language Models with no Catastrophic Forgetting," in Advances in Neural Information Processing Systems, 2022, vol. 35, pp. 16203-16220.*



(a) Reward neural network.



(b) Policy neural network.

Figure: Architectures of the reward and policy models used in practice differ.The reward neural network learns the joint representation of state and action, while the policy model first learns the state representation. As a result, the representations for these two models are different.

Experiment results

Consider the optimality gap $V_r(\pi^*) - \mathbb{E}[V_r(\bar{\pi})]$, where $\pi^* = \arg\max_\pi V_r(\pi) = \arg\max_\pi \mathbb{E}_{x\sim\rho(\cdot),y\sim\pi(\cdot|x)}[r(s,a)]$, and $\bar{\pi} = \frac{1}{T}\sum_{i=1}^{T} \pi^{(i)}$, where $\pi^{(i)}$ is the policy learned at i-th iteration of the (stochastic) policy optimization process.

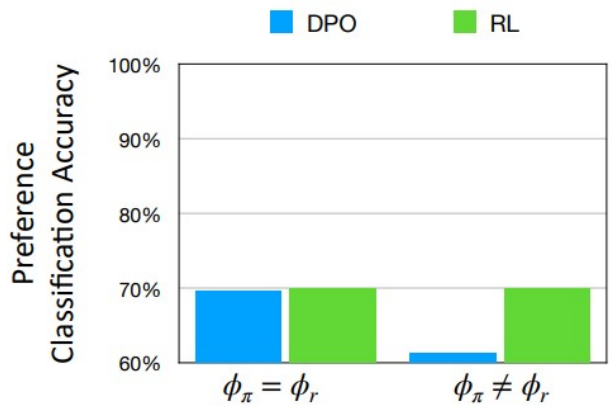


Figure 1: Preference classification accuracy (the larger, the better).

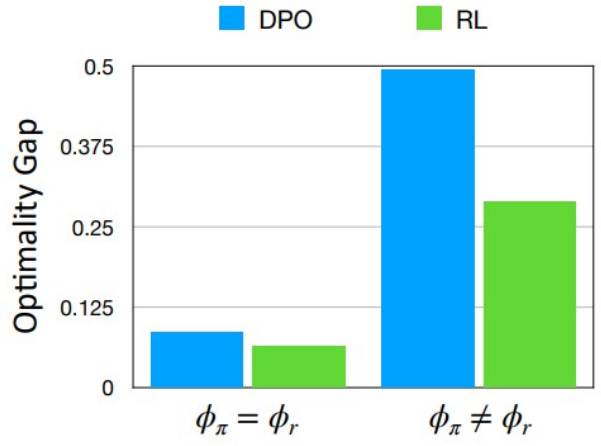


Figure 2: Alignment preference in the optimality gap (the smaller, the better).

When the hidden layer of policy ϕ_π and reward ϕ_r are different, DPO's accuracy significantly lags behind RL, and it also underperforms in terms of optimality gap.

Sufficient Online Update is Crucial for RL

Proposition (Error Bound of Learned Policy in RL)

We define: the reward evaluation error $\epsilon_r := \max_{(x,y)\in\mathcal{X}\times\mathcal{Y}} |r(x,y) - \hat{r}(x,y)|$; *the state distribution estimation error* $\epsilon_s := \sup_\pi |\mathbb{E}_{x\sim\rho(\cdot),a\sim\pi(\cdot|x)}[r(x,y)] - \mathbb{E}_{x\sim\hat{\rho}(\cdot),y\sim\pi(\cdot|x)}[r(x,y)]|$; *and the action distribution estimation error* $\epsilon_a := \sup_\pi ||\nabla \hat{V}_{\hat{r}}(\pi) - \hat{\nabla}\hat{V}_{\hat{r}}(\pi)||^2$. *Then we have that the optimality gap*

$$V_r(\pi^*) - \mathbb{E}[V_r(\bar{\pi})] \leq 2(\epsilon_r + \epsilon_s) + \sqrt{\frac{2(\epsilon_a + |\mathcal{Y}|R_{max}^2)}{T}},$$

where $R_{max} = \max_{(x,y)\in\mathcal{X}\times\mathcal{Y}} |\hat{r}(x,y)|$.

- Reduce ϵ_r by increasing the size of preference dataset;
- reduce ϵ_s by increasing the size of preference-free dataset $\mathcal{D}_{\text{pref-free}} = \{x_i\}$ for online-RL;
- reduce ϵ_a by increasing the number of responses sampled from the policy. This can be done by increasing the computational resources.
- But (offline) DPO can only reduce ϵ_r!

Online-RL

Sufficient non-annotated easy-to-obtain prompt and response data also helps reduce the estimation error and thus the optimality gap.

	RL (online)	**RL (prompts not suff.)**	**RL (responses not suff.)**
Optimality gap	0.2874	0.3078	0.2999

Table: Alignment performance in the optimality gap of RL-based algorithms.

Remark

- In the original paper, the error bound for the previous proposition is

$$V_r(\pi^*) - \mathbb{E}[V_r(\bar{\pi})] \leq 2(\epsilon_r + \epsilon_s) + \sqrt{\frac{2(\epsilon_a + |\mathcal{X}||\mathcal{Y}|^2 R_{\max}^2)}{T}},$$

  but one can easily derive a tighter bound with

$$V_r(\pi^*) - \mathbb{E}[V_r(\bar{\pi})] \leq 2(\epsilon_r + \epsilon_s) + \sqrt{\frac{2(\epsilon_a + |\mathcal{Y}| R_{\max}^2)}{T}}.$$

- The authors study linear bandit task with a finite action-state space. In the case where action-state space is countably infinite, the error bound may not be interesting.