

Alignment of Language Models – Contrastive Learning

Large Language Models: Introduction and Recent Advances

ELL881 · AIL821



Gaurav Pandey
Research Scientist, IBM Research

Policy Gradient/PPO for LLM alignment

- Collect human preferences (x, y_+, y_-)

- Learn a reward model

$$\phi^* = \operatorname{argmax}_{\phi} \sum_{(x, y_+, y_-) \in D} \log \sigma(r_{\phi}(x, y_+) - r_{\phi}(x, y_-))$$

- Train the policy

$$\theta^* = \operatorname{argmax}_{\theta} E_{\pi_{\theta}(y|x)} r_{\phi^*}(x, y) - \beta \cdot KL(\pi_{\theta}(y|x) || \pi_{ref}(y|x))$$

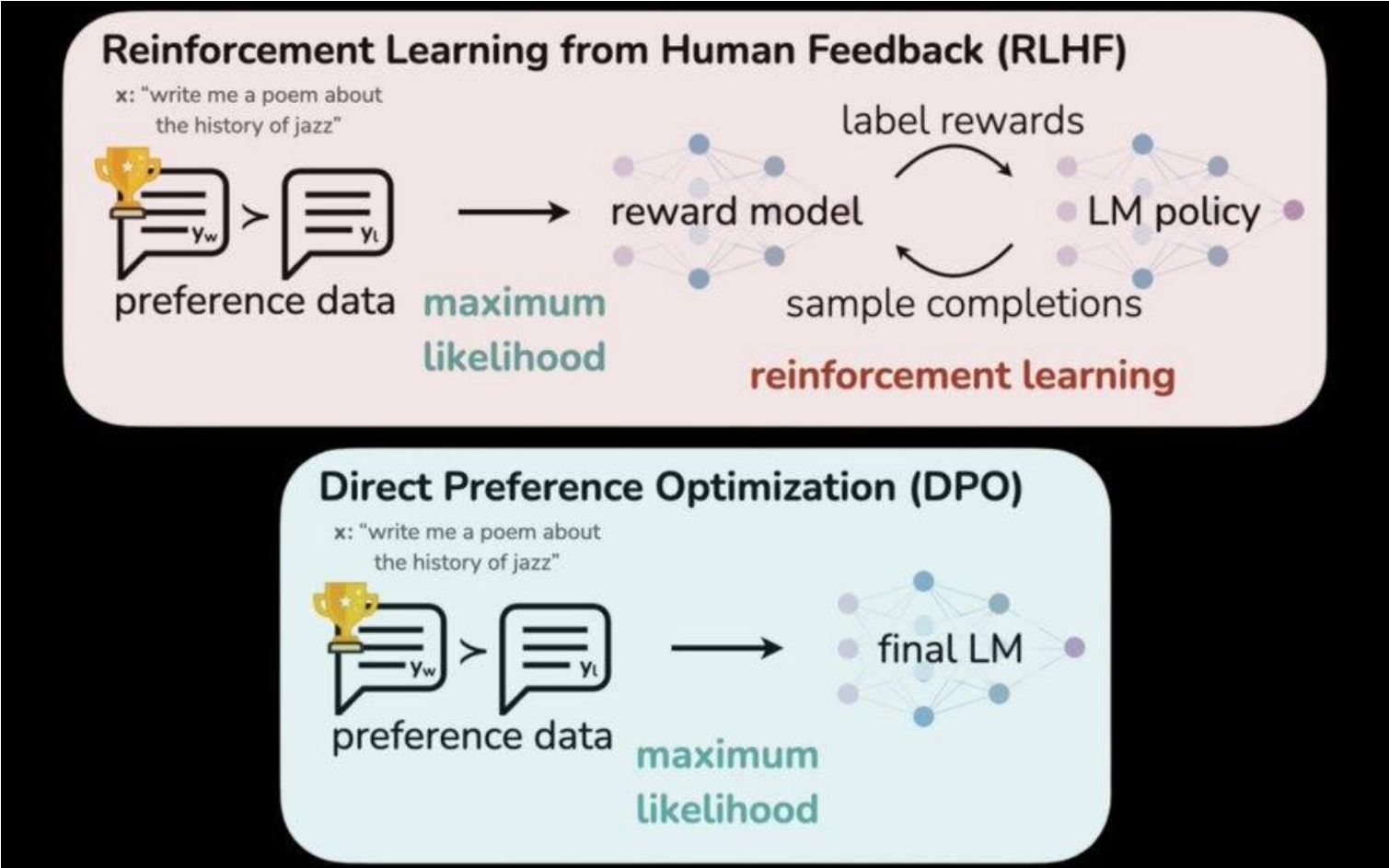
- Optionally

- Also learn the value function

- **Question:** Why do we need this intermediate step of learning reward model?



Direct Preference Optimization on preferences



Credit: <https://arxiv.org/pdf/2305.18290>



The non-parametric case

Assume that the policy & reward model can be arbitrary

- Learn a reward model

$$r^* = \operatorname{argmax}_r \sum_{(x, y_+, y_-) \in D} \log \sigma(r(x, y_+) - r(x, y_-))$$

- Train the policy

$$\pi^* = \operatorname{argmax}_{\pi} E_{\pi(y|x)} r^*(x, y) - \beta \cdot KL(\pi(y|x) || \pi_{ref}(y|x))$$

Primary idea of DPO: Cut out the middle-man r^*



The optimal policy & reward (π^*, r^*)

- Question: What does the optimal policy look like?

$$\pi^* = \underset{\pi}{\operatorname{argmax}} E_{\pi(y|x)} r^*(x, y) - \beta \cdot KL(\pi(y|x) || \pi_{ref}(y|x))$$

subject to $\sum_{y \in Y} \pi(y|x) = 1$



The optimal policy & reward (π^*, r^*)



The parametric policy & reward (π_θ, r_θ)

- In reality, the policy will be parametrized as a language model π_θ
- Idea: Let's parameterize the reward function in terms of the policy parameters.

$$r_\theta(x, y) = \beta \cdot \log \frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)} - \log Z_x(\theta)$$

- Next, train these parameterized reward function directly on human-preferences.



Training the reward function

Given a pair of human preferences (x, y_+, y_-)

- Reward of the positive output

$$r_{\theta}(x, y_+) = \beta \cdot \log \frac{\pi_{\theta}(y_+ | x)}{\pi_{ref}(y_+ | x)} - \log Z_x(\theta)$$

- Reward of the negative output

$$r_{\theta}(x, y_-) = \beta \cdot \log \frac{\pi_{\theta}(y_- | x)}{\pi_{ref}(y_- | x)} - \log Z_x(\theta)$$

- Training objective

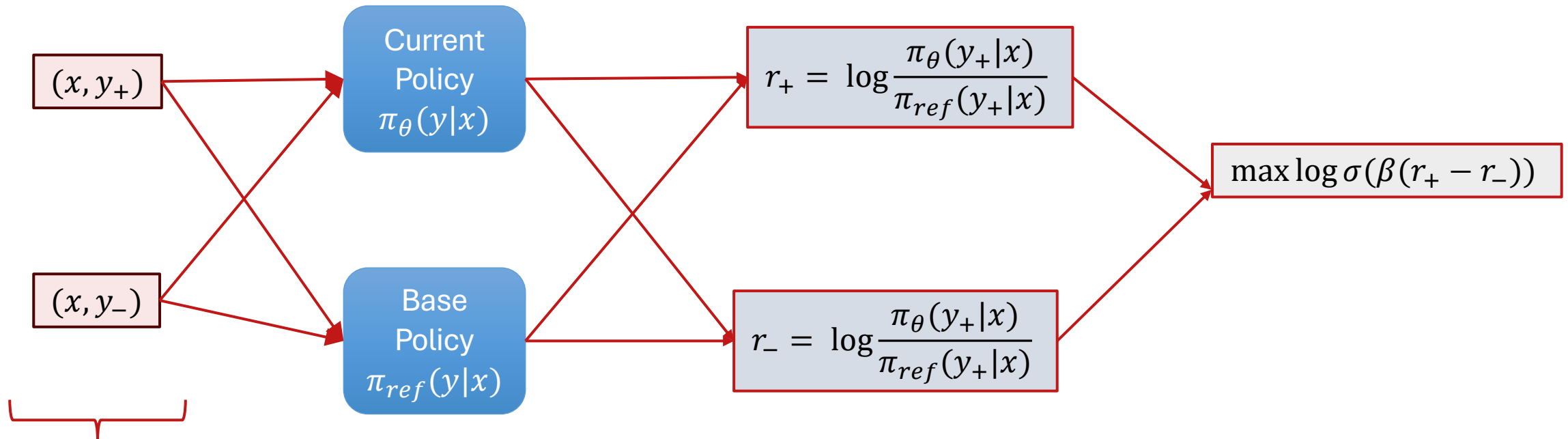
$$\operatorname{argmax}_{\theta} \sum_{(x, y_+, y_-) \in D} \log \sigma(r_{\theta}(x, y_+) - r_{\theta}(x, y_-))$$



The training objective



The DPO objective



Human Preferences



Interpreting the objective

- For a positive output, $\frac{\pi_{\theta}(y_+|x)}{\pi_{ref}(y_+|x)}$ should be high
- If the reference model already assigned high probability to y_+ (say, 0.8)
 - $\pi_{\theta}(y_+|x)$ will have to be relatively higher (say 0.9)
- If the reference model assigned low probability to y_+ (say, 0.1)
 - $\pi_{\theta}(y_+|x)$ will be relatively higher than $\pi_{ref}(y_+|x)$ (say, 0.11)
 - In absolute terms, it might still be low



Interpreting β

$$\log \sigma \left(\beta \left[\log \frac{\pi_{\theta}(y_+ | x)}{\pi_{ref}(y_+ | x)} - \log \frac{\pi_{\theta}(y_- | x)}{\pi_{ref}(y_- | x)} \right] \right)$$

- Higher the value of β , more the model attempts to increase the gap between the reward of +ve and -ve outputs.



PPO vs DPO

- Ongoing debate about the efficacy of the two algorithms
- DPO is simpler – no reward function or value functions are required
- DPO is prone to generating a biased-policy that favors out-of-distribution responses.
- PPO can capture spurious correlations in the reward function.
 - Many reward functions have a length bias – Higher length outputs have higher rewards.
 - PPO training with these reward functions results in longer outputs from the policy.



Why is DPO biased?

$$\log \sigma \left(\beta \left[\log \frac{\pi_{\theta}(y_+|x)}{\pi_{ref}(y_+|x)} - \log \frac{\pi_{\theta}(y_-|x)}{\pi_{ref}(y_-|x)} \right] \right)$$



Why is DPO biased?

$$\log \sigma \left(\beta \left[\log \frac{\pi_{\theta}(y_+ | x)}{\underbrace{\pi_{ref}(y_+ | x)}_{0.5}} - \log \frac{\pi_{\theta}(y_- | x)}{\underbrace{\pi_{ref}(y_- | x)}_{0.5}} \right] \right) \quad \pi_{ref}(y_0 | x) = 0$$

Say $y_0 = (the, the, the)$



Why is DPO biased?

At the beginning of training

$$\log \sigma \left(\beta \left[\log \frac{\pi_{\theta}(y_+ | x)}{\pi_{ref}(y_+ | x)} - \log \frac{\pi_{\theta}(y_- | x)}{\pi_{ref}(y_- | x)} \right] \right) \quad \pi_{\theta}(y_o | x) = 0$$

The equation shows the DPO loss function. The terms $\frac{\pi_{\theta}(y_+ | x)}{\pi_{ref}(y_+ | x)}$ and $\frac{\pi_{\theta}(y_- | x)}{\pi_{ref}(y_- | x)}$ are highlighted with red boxes. Red brackets above each box indicate a weight of 0.5. The overall expression is multiplied by β and passed through a sigmoid function $\log \sigma$. The reference distribution π_{ref} is assumed to be uniform, so $\pi_{ref}(y_o | x) = 0$.

After few steps of training, either $\pi_{\theta}(y_+ | x)$ will increase or $\pi_{\theta}(y_- | x)$ will decrease



Why is DPO biased?

- If $\pi_\theta(y_+|x)$ increases, there is no issue
- If $\pi_\theta(y_-|x)$ decreases, where does the probability go?
 - Ideally, it should go to y_+
 - Most often it goes to y_+ & others (y_o)
- After training, you might end up with

$$\log \sigma \left(\beta \left[\log \frac{\pi_\theta(y_+|x)}{\pi_{ref}(y_+|x)} - \log \frac{\pi_\theta(y_-|x)}{\pi_{ref}(y_-|x)} \right] \right)$$

0.6 0.1

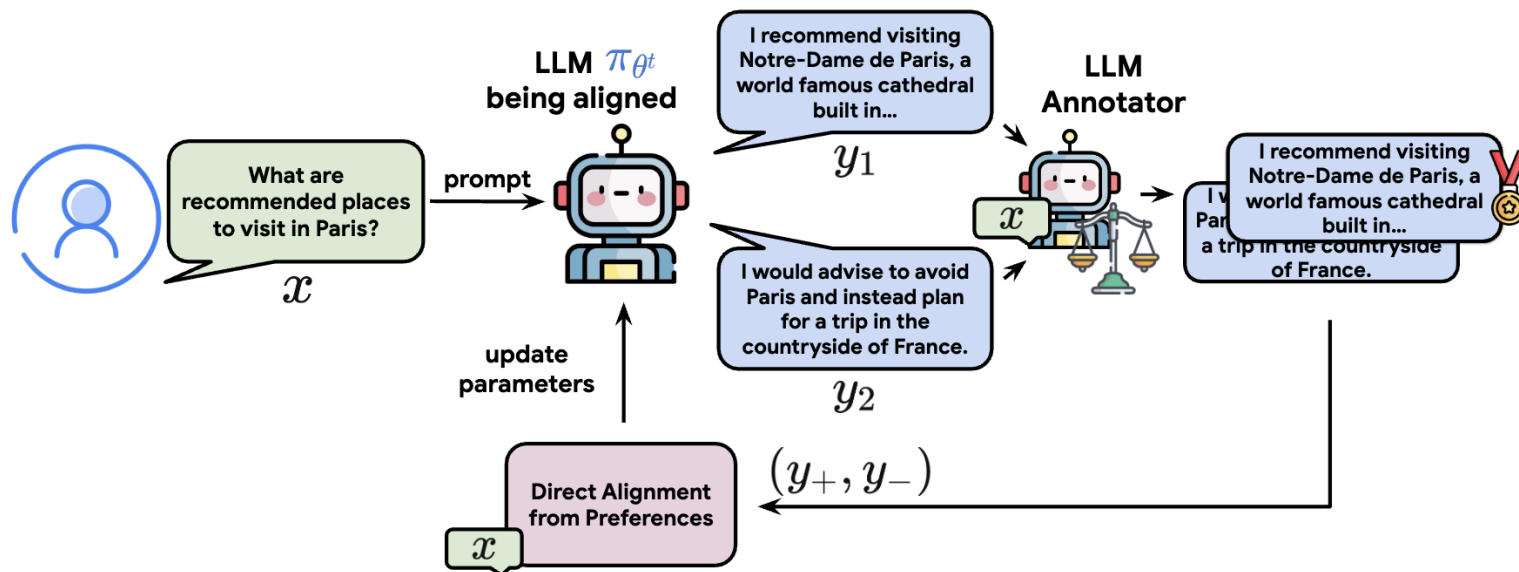
$\pi_\theta(y_o|x) = 0.3$
Say $y_o = (the, the, the)$

- Unfortunately, this is quite common



How to deal with out-of-distribution bias in DPO?

- Possible Solution: Online DPO



- If the probability of a certain OOD output increases
 - It gets sampled in online DPO
 - Gets a low reward
 - Its probability decreases
- Resampling should be done frequently to prevent OOD bias

- Open Problem: How to deal with out-of-distribution bias in offline DPO?

Credit: Direct Language Model Alignment from Online AI Feedback



Performance Comparison: Offline vs Online DPO

Method	Win	Tie	Loss	Quality
TL; DR				
Online DPO	63.74%	28.57%	7.69%	3.95
Offline DPO	7.69%	63.74%	28.57%	3.46
Helpfulness				
Online DPO	58.60%	21.20%	20.20%	4.08
Offline DPO	20.20%	58.60%	21.20%	3.44
Harmlessness				
Online DPO	60.26%	35.90%	3.84%	4.41
Offline DPO	3.84%	60.26%	35.90%	3.57

Table 2: Win/tie/loss rate of DPO with OAIF (online DPO) against vanilla DPO (offline DPO) on the TL; DR, Helpfulness, Harmlessness tasks, along with the quality score of their generations, judged by *human raters*.

Credit: Direct Language Model Alignment from Online AI Feedback



Main Takeaways

- DPO can learn the policy directly from human/AI preferences
 - No reward model or value function needed
- Can be biased towards OOD samples
- To prevent bias
 - A reward model can be trained
 - Outputs can be sampled frequently from the policy and ranked using the reward model

