Alignment of Language Models – Contrastive Learning

Advances in Large Language Models

ELL8299 · AIL861

Gaurav Pandey
Senior Research Scientist, IBM Research

PPO/GRPO for LLM alignment

- Collect human preferences (x, y_+, y_-)
- Learn a reward model

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

- Train the policy

$$\theta^* = \underset{\theta}{\text{argmax}}\, E_x[r_{\phi^*}(x, y) - \beta.KL(\pi_\theta(y|x)||\pi_{ref}(y|x))]$$

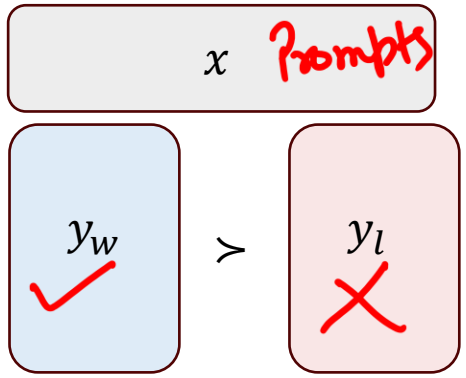
indirectly uses preferences

- Optionally
  - Also learn the value function
- **Question:** Why do we need this intermediate step of learning reward model?

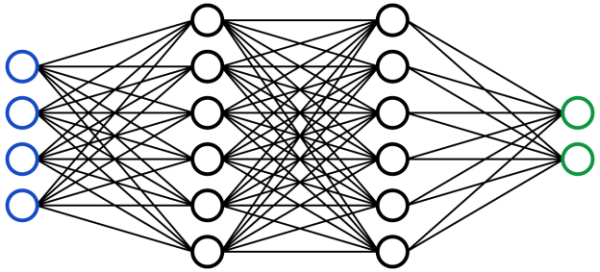
Why go beyond PPO/GRPO?

- Rollouts are expensive.
- Reward Models add bias (trained separately).
- Training loop = Complex RL pipeline.
- Desire: **direct optimization on preference pairs (chosen vs rejected responses).**

Policy learning without rewards

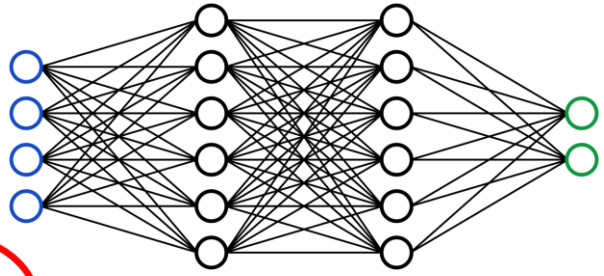


Maximum Likelihood

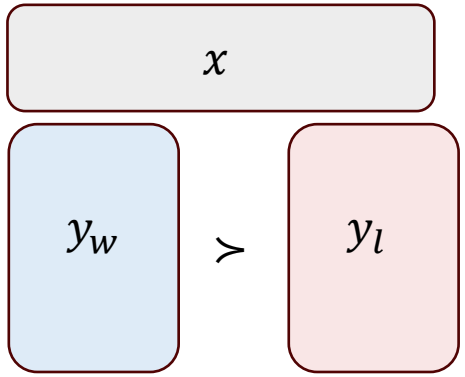


Reward Model

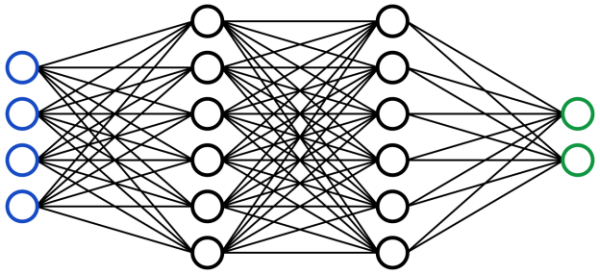
PPO/GRPO



LLM Policy



Maximum Likelihood



LLM Policy

DPO

Desirable scenario

- Given a batch of $(prompt, preferred, rejected) = (x, y_+, y_-)$
- Instead of sampling $\{y_1, \ldots, y_n\}$ and training via RL:
  - Train directly on human preferences (y_+, y_-)

- Optimize policy so

$$\pi_\theta(y_+|x) > \pi_\theta(y_-|x)$$

- Somehow incorporate the KL term

$$KL(\pi_\theta || \pi_{ref})$$

The two optimization problems

Assume that the policy & reward model can be arbitrary

- Learn a reward model

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

- Train the policy

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

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

Cutting out the middle man

- Express the optimium reward model r^* as a function of the optimum policy π^*
  - Derive Lagrangian for the policy objective.
  - Set the partial derivative with respect to the policy to 0.

- Plug it back in the reward model optimization objective.

Lagrangian for the policy objective

- For a fixed prompt x, optimize $\pi(.|x)$

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

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

$$\mathcal{L}(\pi, \lambda) = E_{\pi(y|x)} r^*(x,y) - \beta KL(\pi(\cdot|x) \,||\, \pi_{ref}(\cdot|x)) + \lambda \left(\sum_{y \in Y} \pi(y|x) - 1 \right)$$

The optimum reward (in terms of optimum policy)

- Setting $\partial L / \partial \pi(y|x) = 0$

$$\mathcal{L}(\pi, \lambda) = \sum_{y \in Y} \pi(y|x)\, r^*(x,y) - \beta \sum_{y \in Y} \pi(y|x) \log \frac{\pi(y|x)}{\pi_{ref}(y|x)} + \lambda \left(\sum_{y \in Y} \pi(y|x) - 1 \right)$$

$$\frac{\partial \mathcal{L}}{\partial \pi(y|x)} = r^*(x,y) - \beta \left(1 + \log \frac{\pi^*(y|x)}{\pi_{ref}(y|x)} \right) + \lambda = 0$$

$$r^*(x,y) = \beta \left(1 + \log \frac{\pi^*(y|x)}{\pi_{ref}(y|x)} \right) - \lambda$$

$$\boxed{r^*(x,y) = \beta \log \frac{\pi^*(y|x)}{\pi_{ref}(y|x)} + \beta - \lambda}$$

The optimum policy (in terms of optimum reward)

$$r^*(x,y) - (\beta - \lambda) = \beta \log \frac{\pi^*(y|x)}{\pi_{ref}(y|x)}$$

$$\frac{r^*(x,y)}{\beta} - \left(\frac{\beta - \lambda}{\beta}\right) = \log \frac{\pi^*(y|x)}{\pi_{ref}(y|x)}$$

$$e^{\frac{r^*(x,y)}{\beta}} \times e^{-\frac{(\beta-\lambda)}{\beta}} \times C = \frac{\pi^*(y|x)}{\pi_{ref}(y|x)}$$

$$\pi^*(y|x) = \pi_{ref}(y|x) \exp\left(\frac{r^*(x,y)}{\beta}\right) \times C$$

The optimum policy (in terms of optimum reward)

The optimum policy and reward (π^*, r^*)

$$\pi^*(y|x) \propto \pi_{ref}(y|x) \exp\left(\frac{r^*(x,y)}{\beta}\right)$$

$$r^*(x,y) = \beta \log \frac{\pi^*(y|x)}{\pi_{ref}(y|x)} + (\beta - \lambda) \quad \text{--- IV}$$

$$r_\theta(x,y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{ref}(y|x)} + C$$

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.

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

- 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

Implicit reward

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

- Reward of the negative output

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

- Training objective

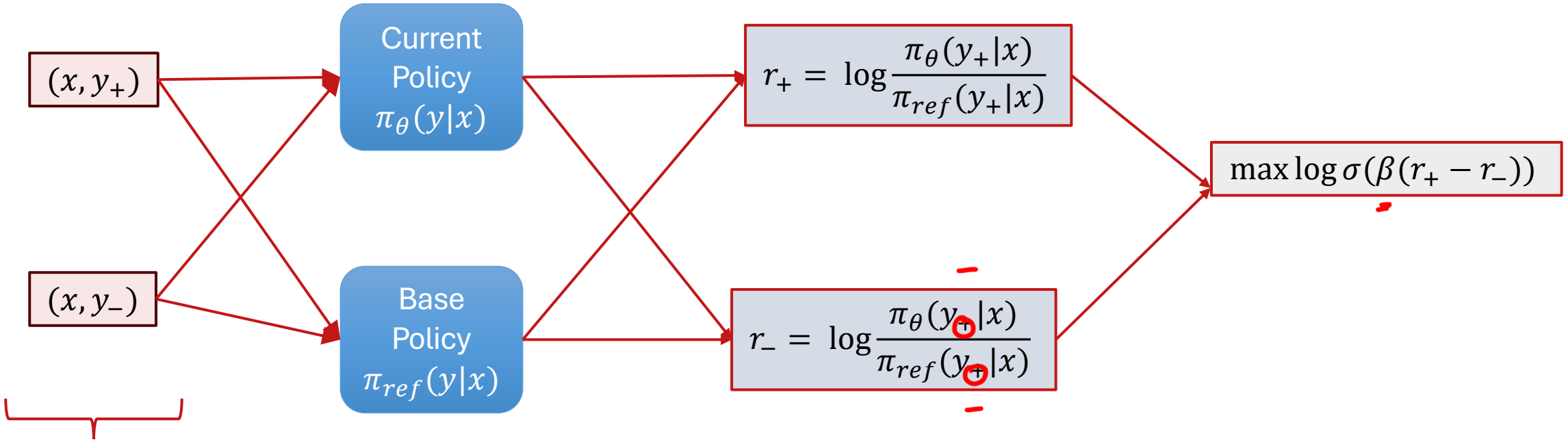
$$\underset{\theta}{\operatorname{argmax}} \sum_{(x,y_+,y_-)\in D} \log \sigma(r_\theta(x, y_+) - r_\theta(x, y_-))$$

The training objective

$$\arg\max_{\theta} \frac{1}{|D|} \sum_{(x, y_+, y_-) \in D} \log \sigma \left(\beta \log \frac{\pi_\theta(y^+|x)}{\pi_{ref}(y^+|x)} + \beta \log Z_x(\theta) - \beta \log \frac{\pi_\theta(y^-|x)}{\pi_{ref}(y^-|x)} - \beta \log Z_x(\theta) \right)$$

$$= \arg\max_{\theta} \log \sigma \left(\log \frac{\pi_\theta(y^+|x)}{\pi_{ref}(y^+|x)} - \log \frac{\pi_\theta(y^-|x)}{\pi_{ref}(y^-|x)} \right)$$

The DPO objective



Human Preferences

The implicit KL divergence

- 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 that $\pi_{ref}(y_+|x)$ (say, 0.11)
  - In absolute terms, it might still be low

$\left(\frac{0.9}{0.8}\right) \approx 1.1$

$0.8 \rightarrow 0.9$ ≈ 1.1

$0.1 \rightarrow 0.11$

Interpreting β in the RLHF objective

$$\theta^* = \underset{\theta}{\text{argmax}}\, E_x[r_{\phi^*}(x, y) - \beta . KL(\pi_\theta(y|x)||\pi_{ref}(y|x))]$$

- Higher the value of β, more the model stays close to the reference policy.

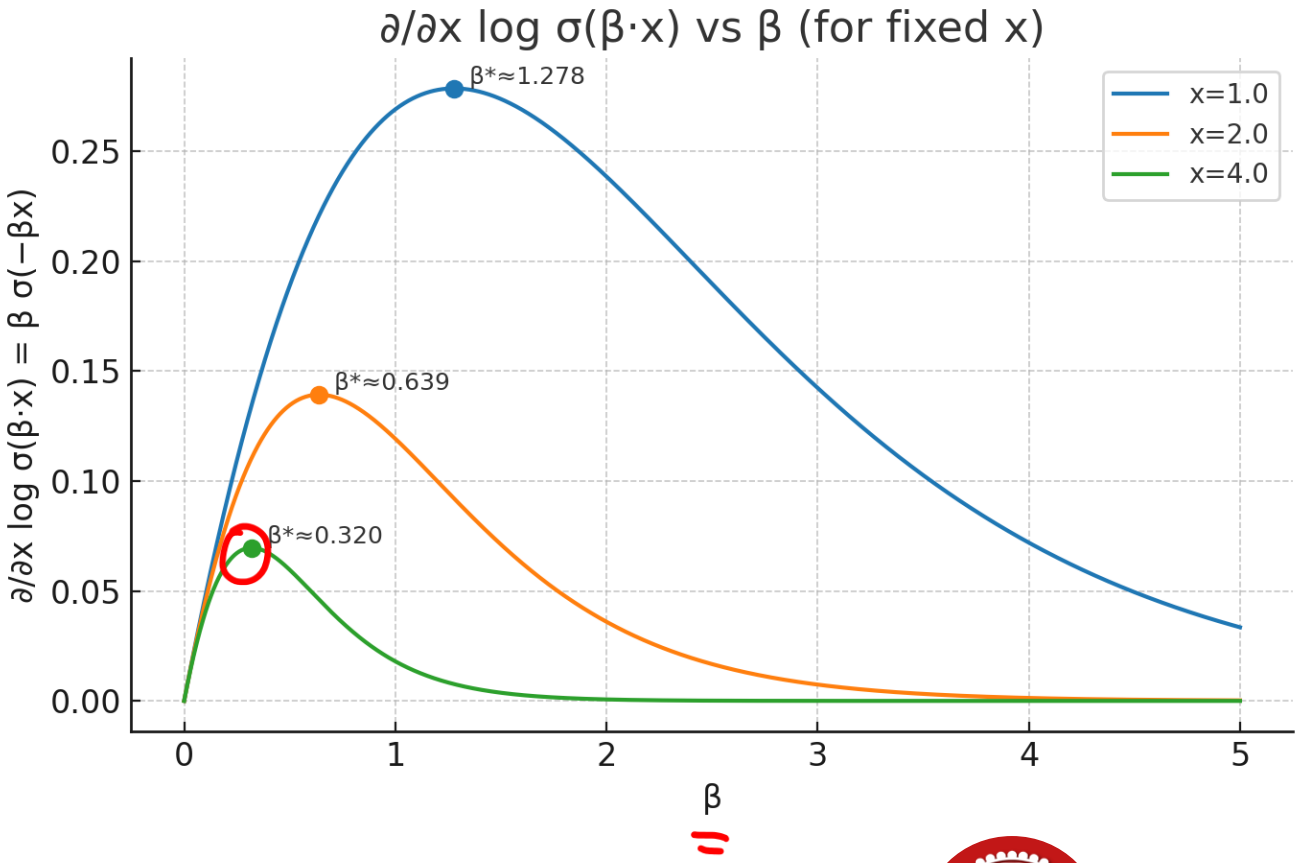
Interpreting β in the DPO objective

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

- It appears
  - Higher the value of β, more the model attempts to increase the gap between the reward of +ve and –ve outputs.
  - It is not true, though.

$\log \sigma(\beta x)$ wrt x

Does higher beta lead to a more contrastive policy?



- x is the difference in implicit reward.
- As β increases
  - Gradient for the implicit reward difference increases
  - Policy becomes more contrastive
- Until the saturation point.
- After that
  - Gradient for the implicit reward difference decreases.
  - Policy becomes less contrastive.

Practical tip

- Treat β as the KL-strength knob
  - Higher β → Lesser drift
  - Lower β → More drift

- Typical starting range $\beta = 0.1$
- Tune by watching $KL(\pi||\pi_{ref})$ and held-out pairwise accuracy
  - If KL runs high, increase β
  - If KL is low, decrease β

PPO vs DPO – Reward/Preference hacking

PPO

- Problem- Model finds adversarial shortcuts to inflate RM scores
- Solution
  - Ensembling reward models
  - Enforcing diversity in output space.

DPO

- Problem – Model captures unimportant differences between the preferences
- Solution
  - Curate hard, diverse preference pairs – not easy to hack
  - Add SFT loss on +ve preference

Why is DPO biased?

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

implicit reward of +ve

Why is DPO biased?

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

$\pi_{ref}(y_+|x)$: 0.5

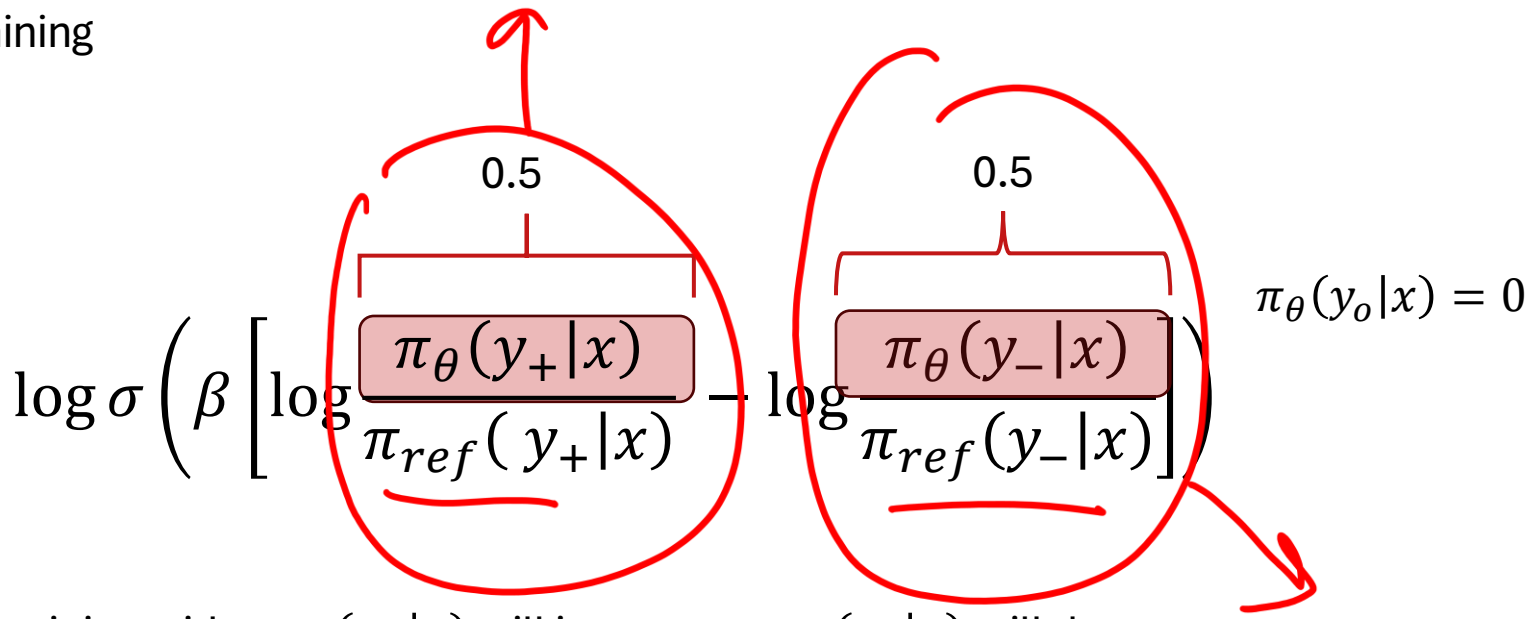
$\pi_{ref}(y_-|x)$: 0.5

$\pi_{ref}(y_o|x) = 0$

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

Why is DPO biased?

At the beginning of training



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

$$\log \sigma\left(\beta\left[\log \frac{\overbrace{\pi_\theta(y_+|x)}^{0.6}}{\pi_{ref}(y_+|x)} - \log \frac{\overbrace{\pi_\theta(y_-|x)}^{0.1}}{\pi_{ref}(y_-|x)}\right]\right)$$

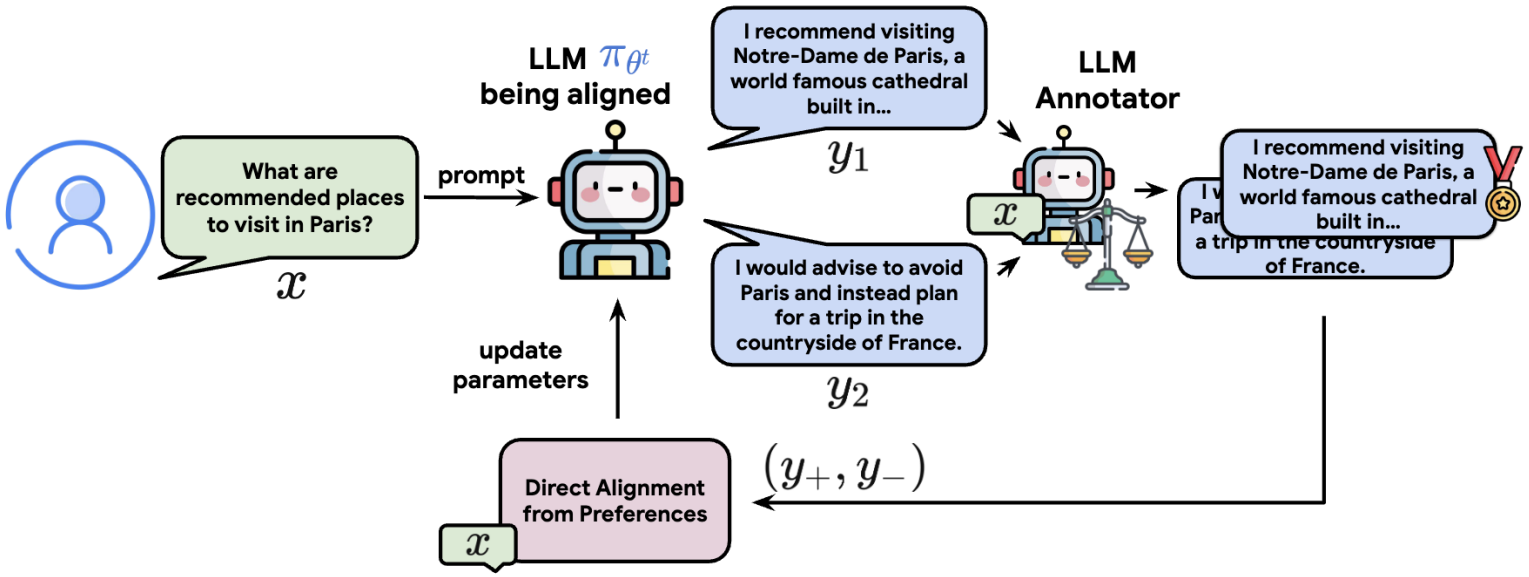
$\pi_\theta(y_o|x) = 0.3$

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

- 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%	3.46
		Helpfulness		
Online DPO	**58.60%**	21.20%	20.20%	**4.08**
Offline DPO	20.20%		58.60%	3.44
		Harmlessness		
Online DPO	**60.26%**	35.90%	3.84%	**4.41**
Offline DPO	3.84%		60.26%	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 capture unimportant differences between the preferences
- 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

Resources

- **Direct Preference Optimization: Your Language Model is Secretly a Reward Model**
- **Direct Preference Optimization Explained In-depth - https://www.tylerromero.com/posts/2024-04-dpo**
- **Direct Language Model Alignment from Online AI Feedback**
- **https://huggingface.co/docs/trl/main/en/online_dpo_trainer**
- **DPO from scratch - https://github.com/0xallam/Direct-Preference-Optimization**