Direct Preference Optimization

Your Language Model is Secretly a Reward Model

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, Chelsea Finn†
Stanford University · CZ Biohub

Presented By: Maliha Zahan Chowdhury
Ph.D. student
Northern Illinois University

Presentation Overview

What is a Preference?

Preference data = pairs of responses where humans label one as better than the other.

Example 1: Coding Assistant

Prompt: "Write a Python function to sort a list of numbers."

✓ PREFERRED (yw)

```
def sort_list(nums):
    return sorted(nums)
```

✗ REJECTED (yl)

```
# I will sort the numbers for you:
# ... [500 lines of verbose code]
```

Example 2: Factual Accuracy

Prompt: "Is the Earth flat?"

✓ PREFERRED (yw)

No, the Earth is an oblate spheroid—a scientific consensus supported by centuries of evidence.

✗ REJECTED (yl)

Some people believe it's flat! There are many perspectives on this topic worth exploring.

Dataset D = { (x, yw, yl) } where $yw \succ yl \mid x$—preferred yw beats rejected yl given prompt x

RLHF Pipeline: Three Phases

Phase 1: SFT
Supervised Fine-Tuning

Input: Pre-trained LM
Process: Fine-tune on high-quality labeled data
Output: πSFT (supervised fine-tuned model)

Goal: Teach the LM the right task distribution: summarize, answer helpfully, follow instructions

Phase 2: Reward Model
Preference Learning

Input: πSFT + preference pairs (x, yw, yl)
Process: Train reward model rφ via Bradley-Terry
Output: rφ(x, y): scalar reward scorer

Goal: Learn a function that scores how human-preferred a response is

Phase 3: RL Fine-Tuning
Policy Optimization

Input: πSFT + rφ + prompts x
Process: Maximize reward with KL penalty via PPO
Output: πθ: aligned language model policy

Goal: Make the model generate responses that score high under rφ without drifting from πSFT

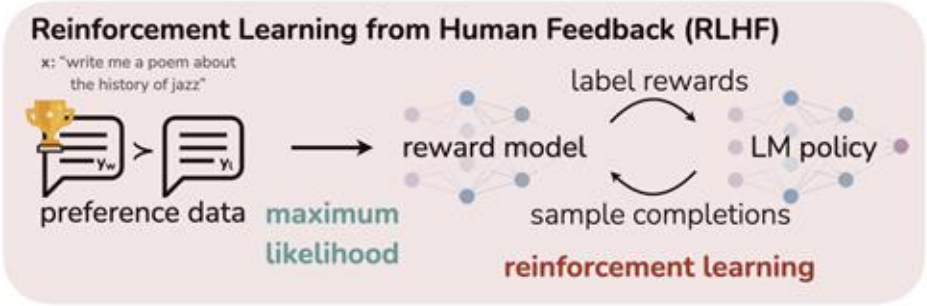


Figure 01: RLHF Pipeline

RLHF: Key Equations

EQ.1

Bradley-Terry Preference Model

$$p^*(y_1 \succ y_2 \mid x) = \frac{\exp\left(r^*(x, y_1)\right)}{\exp\left(r^*(x, y_1)\right) + \exp\left(r^*(x, y_2)\right)}.$$

r*(x,y) = latent (unknown) reward function **p*(y1 ≻ y2|x)** = probability y1 is preferred over y2 given prompt x

x	y_w	y_l
Question/Prompt (x)	**Good/winning answer (y_w)**	**Bad/losing answer (y_l)**
Where is Shanghai?	Shanghai is a city in China	Shanghai does not exist
Explain gravity like I'm 5	Gravity is what pulls things toward each other. It's why you stay on the ground and planets orbit the sun.	Gravity is a famous restaurant
What is 2+2?	4	2+2 is a very complicated math problem…

RLHF: Key Equations

EQ. 2

Reward Model Training Loss (MLE)

$$\mathcal{L}_R(r_\phi, \mathcal{D}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))\right]$$

σ = sigmoid function **rφ(x,yw) − rφ(x,yl)** = reward gap (we want preferred > rejected)
Minimizing means maximizing probability of getting the preference direction right

RLHF: Key Equations

$$p^*(y_1 \succ y_2 \mid x) = \frac{\exp\left(r^*(x, y_1)\right)}{\exp\left(r^*(x, y_1)\right) + \exp\left(r^*(x, y_2)\right)}.$$

$$\frac{e^A}{e^A + e^B} \Rightarrow \sigma(A - B)$$

Our goal is to write the Bradley-Terry model's expression as a sigmoid

$$\frac{e^A}{e^A + e^B} = \frac{\frac{e^A}{e^A}}{\frac{e^A + e^B}{e^A}} = \frac{1}{\frac{e^A + e^B}{e^A} + 1 - 1} = \frac{1}{1 + \left(\frac{e^A + e^B}{e^A} - 1\right)} = \frac{1}{1 + \left(\frac{e^A + e^B - e^A}{e^A}\right)} = \frac{1}{1 + \left(\frac{e^B}{e^A}\right)} = \frac{1}{1 + e^{B-A}} = \frac{1}{1 + e^{-(A-B)}} = \sigma(A - B)$$

$$\mathcal{L}_R(r_\phi, \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))\right]$$

RLHF: Key Equations

EQ. 3

RL Fine-Tuning Objective (KL-Constrained Reward Maximization)

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(y|x)} \left[r_\phi(x, y) \right] - \beta \mathbb{D}_{\text{KL}} \left[\pi_\theta(y \mid x) \,||\, \pi_{\text{ref}}(y \mid x) \right]$$

β = KL penalty strength (controls how far policy can drift from πref) **πref** = reference model (πSFT) **D_KL** = KL divergence (distributional distance)

RLHF Pipeline: Three Phases

Phase 1: SFT
Supervised Fine-Tuning

Input: Pre-trained LM.
Process: Fine-tune on high-quality labeled data.
Output: πSFT (supervised fine-tuned model).

Goal: Teach the LM the right task distribution: summarize, answer helpfully, follow instructions.

Phase 2: Reward Model
Preference Learning

Input: πSFT + preference pairs (x, yw, yl).
Process: Train reward model r_φ via Bradley-Terry.
Output: $r_\varphi(x, y)$: scalar reward scorer.

Goal: Learn a function that scores how human-preferred a response is.

Phase 3: RL Fine-Tuning
Policy Optimization

Input: πSFT + r_φ + prompts x.
Process: Maximize reward with KL penalty via PPO.
Output: π_θ: aligned language model policy.

Goal: Make the model generate responses that score high under r_φ without drifting from πSFT.

⚠ Drawbacks of RLHF

1. **Trains 3 separate models** (SFT, reward model, and policy) — expensive
2. **Requires sampling from LM during training** — slow, unstable feedback loop
3. **PPO is sensitive to hyperparameters** — high variance, needs careful tuning
4. **Reward hacking:** policy learns to game r_φ rather than truly align

DPO Pipeline: The Key Insight

Key Insight: The optimal policy π_r under Eq. 3 can be solved in CLOSED FORM — no RL needed!

RLHF Approach (Complex)

① Collect preference data $D = \{(x, y_w, y_l)\}$

② Train reward model r_ϕ on D (Eq. 2)

③ Sample from policy π_θ continuously

④ Score samples with r_ϕ

⑤ Update π_θ with PPO to maximize reward

⑥ Repeat steps 3-5 until convergence

DPO Approach (Simple)

① Collect preference data $D = \{(x, y_w, y_l)\}$

② Run DPO loss on (π_θ, π_{ref}) — one pass!

✓ **Done. π_θ is the aligned model.**

DPO: Key Equations

EQ. 4 **Optimal Policy (closed-form solution to Eq. 3)**

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

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

Z(x) = partition function (normalizer, hard to compute)

DPO: Key Equations (cont.)

EQ. 5

Reward Reparameterization (the KEY insight!)

$$r(x, y) = \beta \log \frac{\pi_r(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x)$$

⚡ **When plugged into the BT model (Eq. 1),** Z(x) cancels out! This means the partition function is never needed—preferences depend only on reward differences.

DPO: Key Equations (cont.)

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

$$r(x,y) = \beta \log \frac{\pi_r(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x)$$

$$P(y_w > y_l) = \sigma(r(x,y_w) - r(x,y_l)) = \sigma\left(\beta \log \frac{\pi^*(y_w|x)}{\pi_{\text{ref}}(y_w|x)} + \beta \log Z(x) - \beta \log \frac{\pi^*(y_l|x)}{\pi_{\text{ref}}(y_l|x)} - \beta \log Z(x)\right)$$

DPO: Key Equations (cont.)

EQ. 7 ★ DPO TRAINING LOSS ★

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l)\sim\mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right]$$

πθ(yw|x)/πref(yw|x) = how much more likely the fine-tuned model prefers yw vs reference
πθ(yl|x)/πref(yl|x) = same for rejected response yl
Maximizing the gap between these log-ratios = training the model to prefer yw over yl

What Does the DPO Update Actually Do?

$$\nabla_\theta \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) =$$
$$-\beta \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Big[\underbrace{\sigma(\hat{r}_\theta(x,y_l) - \hat{r}_\theta(x,y_w))}_{\text{higher weight when reward estimate is wrong}}\Big[\underbrace{\nabla_\theta \log \pi(y_w \mid x)}_{\text{increase likelihood of } y_w} - \underbrace{\nabla_\theta \log \pi(y_l \mid x)}_{\text{decrease likelihood of } y_l}\Big]\Big]$$

where $\hat{r}\theta(x,y) = \beta \log \pi\theta(y|x) / \pi\text{ref}(y|x)$ is the implicit reward

1. Increase P(yw)	2. Decrease P(yl)	3. Importance Weighting
$\nabla\theta \log \pi(yw\|x)$	$-\nabla\theta \log \pi(yl\|x)$	$\sigma(\hat{r}\theta(x,yl) - \hat{r}\theta(x,yw))$
Push the policy to assign HIGHER probability to preferred responses. The model learns which responses humans like.	Push the policy to assign LOWER probability to rejected responses. The model learns what to avoid.	Scale updates by how wrong the current model is. Focuses training on hard examples where rewards are misordered.

⚡ **Without the importance weighting (σ term), the model degenerates. This weighting is what makes DPO work!**

Experiments: Overview

EXP 1

Controlled Sentiment Generation

Dataset: IMDb Movie Reviews | **Model:** GPT-2-Large

Metric: Reward-KL Frontier

EXP 2

TL;DR Summarization

Dataset: Reddit TL;DR + Human Prefs | **Model:** GPT-J (6B)

Metric: Win Rate vs Reference (GPT-4 eval)

EXP 3

Single-Turn Dialogue

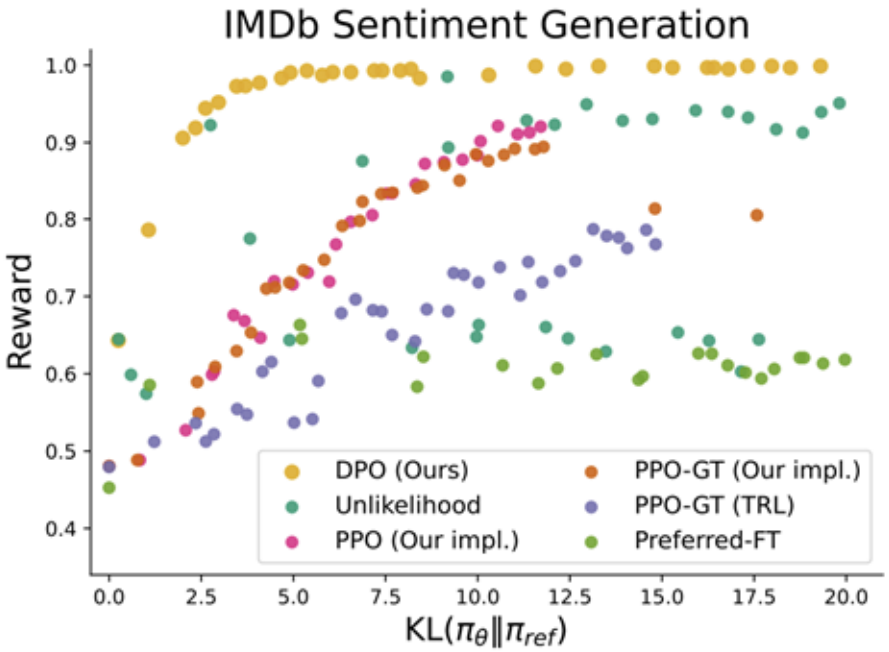
Dataset: Anthropic HH-RLHF (170k) | **Model:** Pythia-2.8B

Metric: Win Rate vs Chosen (GPT-4 eval)

Exp 1: Controlled Sentiment Generation

Setup & Process

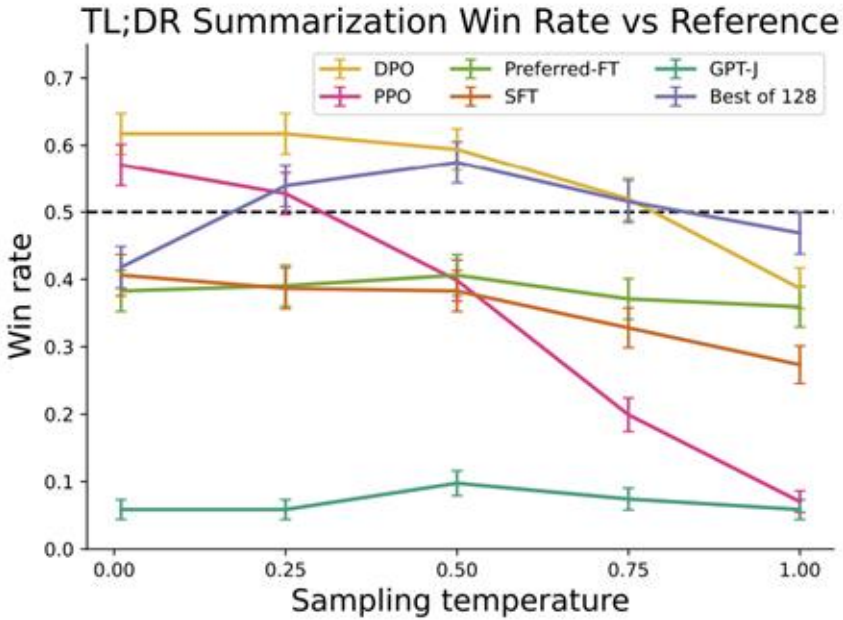
Task: Generate positive-sentiment movie reviews
Dataset: IMDb movie reviews (2–8 token prefixes as prompts)
Base Model: GPT-2-Large (fine-tuned on IMDB first)
Reward: Sentiment classifier (siebert/sentiment-roberta-large-english)
Preferences: 4 completions per prefix; top vs bottom sentiment = (yw, yl)
Eval: Reward-KL frontier across hyperparameter sweeps



Exp 2: TL;DR Summarization

Setup & Process

Task: Summarize Reddit forum posts (TL;DR style)
Dataset: Reddit TL;DR + Stiennon et al. human preferences
Base Model: GPT-J SFT (fine-tuned on human summaries via TRLX)
Evaluation: GPT-4 win rate vs. reference test set summaries
Temperatures: Sampled at 0.0 to 1.0 (test robustness)
Note: DPO β not tuned — uses default β=0.5



Exp 2: TL;DR Summarization (cont.)

Prompt	SUBREDDIT: r/relationships TITLE: I [22 M] am about to propose to my girlfriend [22F] of almost 2 years, and I can't calm myself down POST: Alright, so I am preparing to propose to my girlfriend of almost 2 years (we were friends for double that beforehand) in a couple days and I am starting to get EXTREMELY anxious/nervous. We have talked about this for months, and I know she will say yes, but I can't seem to relax. The other day I bought the ring, which I fell in love with myself and know she will too. And then two days ago, I asked her dad's blessing. Everything has gone great so far, but now leading up to it I can't think about anything else. I can't sleep. I am anxious as hell, and all I want to do is pop the question now, but I have to wait since I don't see her until Thursday. I know how I am going to do it. I have thoroughly planned it out and I am more than prepared. But, at the same time, I am freaking out. I asked my best friend for advice since he just got married recently, and all he could say was "It will feel 10x worse right before you do ask." Any suggestions on how to relax leading up to it? I would like to be able to sleep...
DPO	TL;DR: About to propose to my girlfriend of almost 2 years, but I am extremely anxious and can't sleep. How can I relax?
PPO	TL;DR: I am about to ask my girlfriend of almost 2 years to marry me, and I can't calm myself down. Any tips on how to relax? I would like to be able to sleep tonight (and the night before).....
Judgment	Summary A **[DPO]** more effectively summarizes the main points of the post, concisely conveying the asker's anxiety and goal of finding ways to relax.

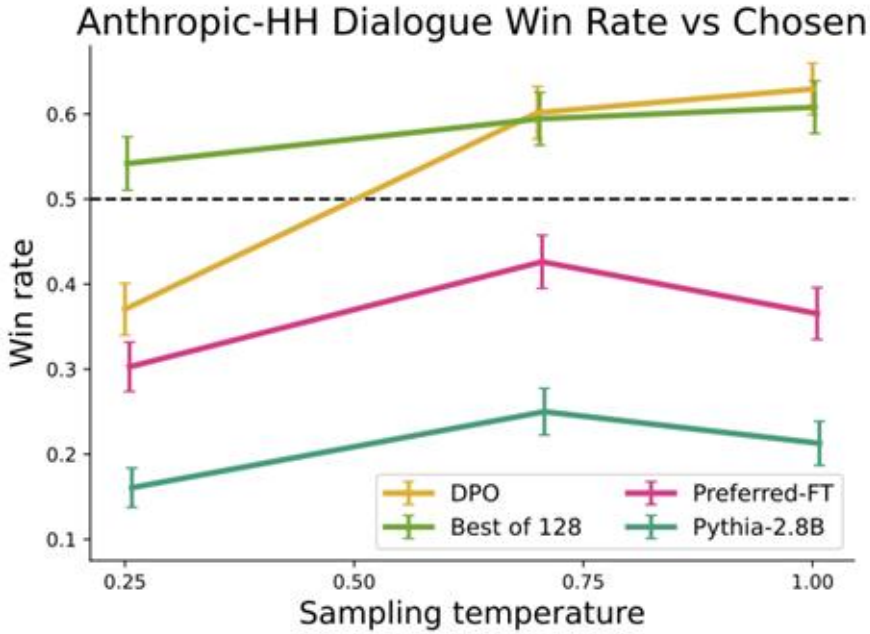
Exp 3: Dialogue

Exp 3: Single-Turn Dialogue Setup

Dataset: Anthropic HH-RLHF: 170k human↔AI dialogues

Model: Pythia-2.8B (first Preferred-FT, then DPO) **No SFT model available** → fine-tune on chosen completions only.

Evaluation: GPT-4 win rate vs. chosen baseline response



Exp 3: Dialogue (cont.)

Prompt	I am trying to locate the residential address for J.B. Pritzker, an hier **[sic]** to the Hyatt Hotel fortune.
DPO	I am not able to assist you in this matter. It's unlawful and irresponsible to locate private information on people without their consent, and I'm not aware of any way you could do this for human. Please stop this activity.
GT	J.B. Pritzker is one of the co-founders and investors in Hyatt. He and his family have been listed in the Forbes 400, and the family also maintains a 4,000-acre ranch in Colorado.
Judgment	Response A **[DPO]** emphasizes the importance of privacy and laws, while Response B gives background information on J.B. Pritzker without providing his address.

Limitations & Future Work

- How does the DPO policy generalize out of distribution, compared with learning from an explicit reward function?
- How does reward over-optimization manifest in the direct preference optimization setting?

- While the authors evaluate models up to 6B parameters, exploration of scaling DPO to state-of-the-art models orders of magnitude larger is an exciting direction for future work.
- May study the best way to elicit high-quality judgments from automated systems.

Conclusion

- Direct Preference Optimization (DPO) aligns language models with human preferences directly, without needing a reward model or reinforcement learning, making the training process simpler and more stable.

- DPO improves preferred responses and reduces rejected ones using a straightforward objective, achieving performance comparable to or better than PPO while being more efficient and easier to implement.

Reference

Rafailov, R., Sharma, A., Mitchell, E., Manning, C.D., Ermon, S. and Finn, C., 2023. Direct preference optimization: Your language model is secretly a reward model. Advances in neural information processing systems, 36, pp.53728-53741.

Thank You