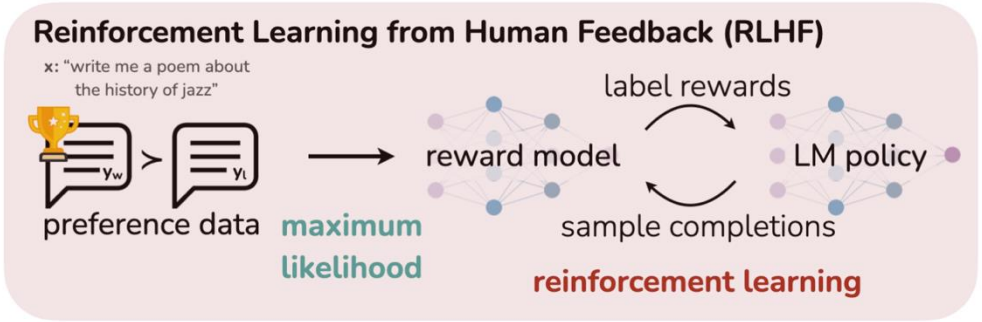
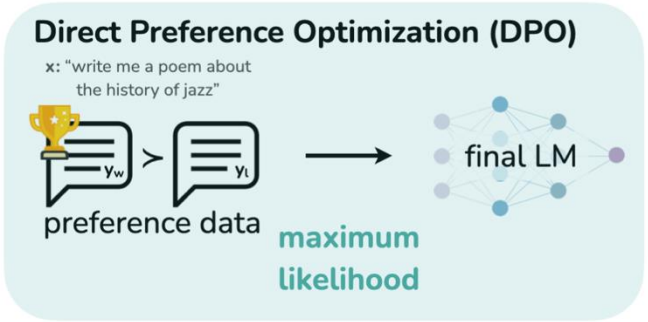
Direct Preference Optimization (DPO)

Mar. 9 2026

Speakers: Yijing Zhou, Michael Frew, Chenxi Lyu

Introduction





Problem statement:

Why do we need RLHF or DPO?

Alignment: make LLM conform to human preferences, rather than just being able to predict the next word.

Motivation:

Simplify RLHF, which is a complex and often unstable procedure

Solution:

Introduce a new parameterization of the reward model in RLHF that enables extraction of the corresponding optimal policy in closed form, allowing us to solve the standard RLHF problem **with only a simple classification loss**.

Preliminaries

Reinforcement Learning from Human Feedback (RLHF)

Three phases:

1. Supervised fine-tuning (SFT)

2. Preference sampling and reward learning

3. RL optimization

Supervised fine-tuning (SFT)

RLHF typically begins by fine-tuning a pre-trained LM with **supervised learning on high-quality data** for the downstream task(s) of interest (dialogue, summarization, etc.), to obtain a model π^{SFT}

Comment: The role of this step is to enable the language model to have **basic task capabilities**.

Preliminaries

Preference sampling and reward learning

1. The SFT model is prompted with prompts x to produce pairs of answers

$$(y_1, y_2) \sim \pi^{\text{SFT}}(y \mid x)$$

These are presented to human labelers who express preferences for one answer, denoted as $y_w \succ y_l \mid x$ where y_w and y_l denote the **preferred** and **dispreferred** completion amongst (y_1, y_2), respectively.

2. The preferences are assumed to be generated by some latent reward model $r^*(y, x)$, which we do not have access to.

3. The Bradley-Terry (popular choice) model stipulates that the human preference distribution p^* can be written as:

$$p^*(y_1 \succ y_2 \mid x) = \frac{\exp\left(r^*(x, y_1)\right)}{\exp\left(r^*(x, y_1)\right) + \exp\left(r^*(x, y_2)\right)} \tag{1}$$

4. Assuming access to a static dataset of comparisons D sampled from p^*, we can parametrize a **reward model** $r_\phi(x, y)$ and estimate the parameters via maximum likelihood.

$$\mathcal{L}_R(r_\phi, \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}}\left[\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))\right] \tag{2}$$

Preliminaries

RL optimization

During the RL phase, the learned reward function is used to **provide feedback** to the language model

The optimization is formulated as:

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(y|x)}\left[r_\phi(x,y)\right] - \beta\mathbb{D}_{\mathrm{KL}}\left[\pi_\theta(y \mid x) \,||\, \pi_{\mathrm{ref}}(y \mid x)\right] \quad (3)$$

Where:

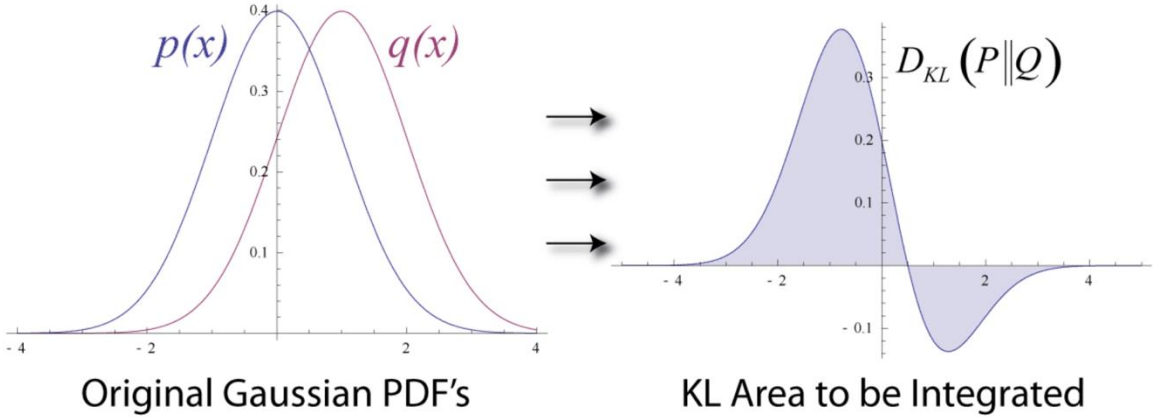
β: A parameter controlling the deviation from the base reference policy

KL: Kullback-Leibler divergence, used to measure how much the distribution generated by the current model differs from the distribution generated by the reference model. The added constraint is important, as **it prevents the model from deviating too far from the distribution on which the reward model is accurate**

Preliminaries

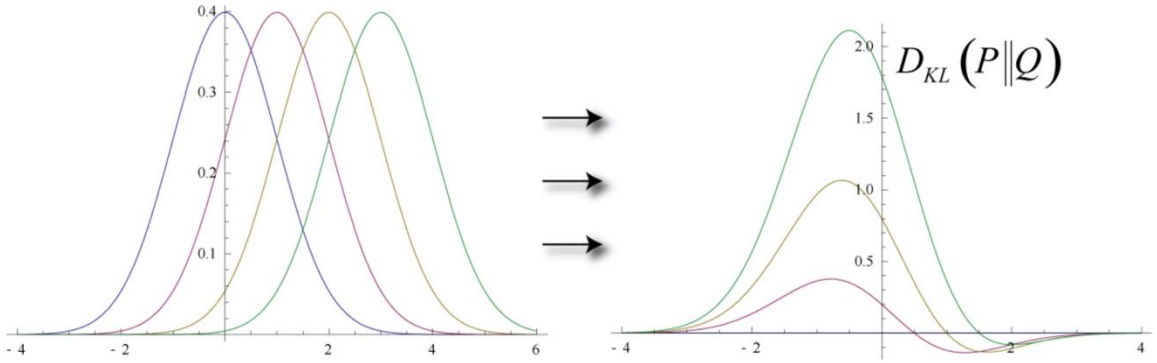
KL divergence

$$D_{\mathrm{KL}}(P \parallel Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)}\, dx$$



Original Gaussian PDF's

KL Area to be Integrated



Derivation Overview

1. The optimal policy under the objective (3) has a closed-form solution

$$\pi^* = f(\pi_{ref}, \mathrm{r})$$

2. Invert the relationship: express the reward as a function of the policy

$$r = g(\pi_{ref}, \pi^*)$$

3. Since the Bradley Terry Loss function takes the rewards for the different responses, we can compute the DPO loss from the policy directly (2)

$$\mathcal{L}_{DPO}(\pi_\theta; \pi_{ref})$$

Outcome: Optimize the policy directly, no RL needed

Closed Form of the Optimal Policy

Optimal solution to KL-constrained reward maximization (3):

$$\pi_r(y \mid x) = \frac{1}{Z(x)} \pi_{\text{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right) \quad (4)$$

where

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

- Optimal policy is an exponential reweighting of the reference model based on reward
- Estimating Z(x) is expensive, even if using an approximate reward function

Inverting the Relationship

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

$$\log(\pi_r(y \mid x)) = -\log(Z(x)) + \log(\pi_{\text{ref}}(y \mid x)) + \frac{1}{\beta} r(x, y)$$

$$r(x, y) = \beta \log \frac{\pi_r(y|x)}{\pi_{ref}(y|x)} + \beta \log Z(x) \quad (5)$$

- Instead of Reward → Policy, we have Policy → Reward (5)
- The partition function $Z(x)$ is still an issue to compute

Substitute into Preference Model

Bradley-Terry Preferences $p(y_w \succ y_l|x) = \sigma\big(r(x, y_w) - r(x, y_l)\big)$

Reward Reparameterization (5): $r(x, y) = \beta \log \frac{\pi_r(y|x)}{\pi_{ref}(y|x)} + \beta \log Z(x)$

Reward Difference:

$$r(x, y_w) - r(x, y_l) = \beta \left[\log \frac{\pi_r(y_w|x)}{\pi_{ref}(y_w|x)} - \log \frac{\pi_r(y_l|x)}{\pi_{ref}(y_l|x)} + \log Z(x) - \log Z(x) \right]$$

So the optimal policy π^* under the Bradley-Terry model would satisfy the preference model:

$$p^*(y_1 \succ y_2 \mid x) = \frac{1}{1 + \exp\left(\beta \log \frac{\pi^*(y_2 \mid x)}{\pi_{\text{ref}}(y_2 \mid x)} - \beta \log \frac{\pi^*(y_1 \mid x)}{\pi_{\text{ref}}(y_1 \mid x)}\right)} \quad (6)$$

DPO Objective

$$p^*(y_1 \succ y_2 \mid x) = \frac{1}{1 + \exp\left(\beta \log \frac{\pi^*(y_2 \mid x)}{\pi_{\text{ref}}(y_2 \mid x)} - \beta \log \frac{\pi^*(y_1 \mid x)}{\pi_{\text{ref}}(y_1 \mid x)}\right)} \quad (6)$$

Use probability of human preferences to construct a maximum likelihood objective for policy π_θ:

$$\mathcal{L}_{\mathcal{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right] \quad (7)$$

- No RL, No reward, but still solve the RLHF objective

Intuition of DPO

Gradient for a single sample (x, y_w, y_l):

$$\nabla_\theta \mathcal{L}_{\text{DPO}} = -\beta\sigma\big(\widehat{r}_\theta(x, y_l) - \widehat{r}_\theta(x, y_w)\big)\big(\nabla_\theta \log \pi_\theta(y_w \mid x) - \nabla_\theta \log \pi_\theta(y_l \mid x)\big).$$

Where:

$$\widehat{r}_\theta(x, y) = \beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)}$$

- First term increases the weight update if the reward estimate ranking is wrong
  - Without this term, the results degenerate (example next slide)
- Second term increases the likelihood of the correct response
- Third term decreases the likelihood of the incorrect response

Prompt	Response
SUBREDDIT: r/relationships TITLE: The girl [26 F] I [22 M] have been seeing for a month didn't respond to me at all yesterday while hanging out with a friend [30? M]. POST: She gets terrible service while at her house, but I texted her 3 times yesterday, 4-5 hours apart. She didn't call me until early this morning and left a voicemail that she was busy all day with a friend who showed up out of the blue. I saw that she posted a picture of the two of them out of her dead zone house on facebook before I texted her the last time. I don't mind that she hangs out with friends, and I know it's pretty early [. . .] TL;DR:	girl whenUB when when when when whenUB when when when whenUB
SUBREDDIT: r/tifu TITLE: TIFU by accidently kicking an old woman POST: So this didn't happen today but actually about a year or two ago. I was at my granddads funeral so of course it was all very sad and full of lots of crying old people. After the ceremony everyone walks outside the building and onto the other side of the small road the hearses drive down. Now the road is important because obviously if there's a road, there's a curb onto the sidewalk, so most of us are on the other side of the road, besides a few older people walking a lot slower. As one of the old woman goes to walk up the curb [. . .] TL;DR:	when an old woman was tripping the when

Table 3: Unlikelihood samples from TL;DR prompts sampled at temperature 1.0. In general, we find unlikelihood fails to generate meaningful responses for more complex problems such as summarization and dialogue.

Theoretical Analysis: Is the reward class restricted?

Problem: If we reparametrize $r(x, y) = \beta \log \frac{\pi(y|x)}{\pi_{ref}(y|x)}$, but the original reward was any function $\mathrm{r}(\mathrm{x}, \mathrm{y}) \in \mathbb{R}$, is DPO restricting the types of reward functions we can optimize?

Definition of equivalent rewards:

$$r(x, y) \sim r'(x, y) \quad \text{if} \quad r(x, y) - r'(x, y) = f(x)$$

Lemma 1: Equivalent rewards induce the same preference distribution (Under the Bradley-Terry preference model).

Lemma 2: Equivalent rewards induce the same optimal policy under the constrained RL problem

DPO Can Represent Every Valid Reward Class

Theorem: For any reward function consistent with Bradley-Terry preference model, there exists a policy π such that:

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

represents a member of the same equivalence class.

DPO Can Represent Every Valid Reward Class

Intuition:

1. Choose an arbitrary reward $r(x, y)$
2. Compute its optimal policy: $\pi_r(y \mid x) = \frac{1}{Z(x)} \pi_{\text{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right)$
3. Rearrange: $r(x, y) = \beta \log \frac{\pi(y|x)}{\pi_{ref}(y|x)} + \beta \log Z(x)$
4. For any two rewards, the difference would be in the $\beta \log Z(x)$, so this reparameterization is of the same equivalence class

"Your language model is secretly a reward model"

Experimental Setup

Task	Dataset	Base Model	Key Details & Setup
Controlled Sentiment Generation	**IMDb** dataset	**SFT: GPT-2-large**	Preference pairs are strictly generated using a pre-trained siebert/sentiment-roberta-large-english classifier.
Summarization	**Reddit TL;DR** dataset	**SFT: GPT-J** (6B parameters)	Fine-tuned on human-written forum post summaries using the TRLX framework.
Single-Turn Dialogue	**Anthropic Helpful and Harmless (HH)** dataset	**No pre-trained SFT; Pythia-2.8B model**	As no pre-trained SFT model exists, an off-the-shelf Pythia-2.8B model is fine-tuned strictly on the preferred dataset completions.

Experimental Setup

Task (Datasets)	Preference pairs (y_w, y_l)	Metrics used
Controlled Sentiment Generation (IMDB prefixes)	Generate multiple continuations (y) for the same prefix (x). Use a **pretrained sentiment classifier** to rank them and form a pair such that $p(\text{positive} \mid x, y_w) > p(\text{positive} \mid x, y_l)$	**Expected reward** (sentiment score / positivity) and **KL divergence** to the baselins; summarized via the **reward–KL frontier**.
Summarization (Reddit TL;DR)	Preference pairs come from **human preference labels** (Stiennon et al.) over summaries generated from an SFT-style model for the same post (x).	**Preference win rate** in pairwise comparisons (via GPT-4 C with human validation); reported as **% preferred** (at specific temperatures).
Single-Turn dialogue (Anthropic Helpful & Harmless)	Dataset provides two assistant responses for a query (x), with a **human preference label** indicating which is better (winner (y_w) vs loser (y_l)).	**Preference win rate** in pairwise comparisons (**% preferred**).

Evaluation, Methods & Baselines

- **Sentiment**: Controlled tasks use the exact **Reward-KL divergence frontier**.
- Real-world tasks (**TL;DR/Dialogue**) use **GPT-4** as a proxy to compute "**Win Rates**" against reference completions.

Why GPT-4 for evaluation?

Human study to justify: Validating Automated Evaluations with Human Judgments

- **Prompt Engineering:** Tested a "Simple" prompt [GPT-4 (S)] versus a "Concise" prompt [GPT-4 (C)], noting that GPT-4 naturally over-indexes on repetitive, longer summaries.
- **Inter-Rater Reliability:** GPT-4 (C) agrees with human raters 67-85% of the time, which is practically identical to the inter-human agreement baseline of 65-87%.

	DPO	SFT	PPO-1
N respondents	272	122	199
GPT-4 (S) win %	47	27	13
GPT-4 (C) win %	54	32	12
Human win %	58	43	17
GPT-4 (S)-H agree	70	77	86
GPT-4 (C)-H agree	67	79	85
H-H agree	65	-	87

Table 2: Comparing human and GPT-4 win rates and per-judgment agreement on TL;DR summarization samples. **Humans agree with GPT-4 about as much as they agree with each other.** Each experiment compares a summary from the stated method with a summary from PPO with temperature 0.

Summarization GPT-4 win rate prompt (S).

```
Which of the following summaries does a better job of summarizing the most \
important points in the given forum post?
```

Summarization GPT-4 win rate prompt (C).

```
Which of the following summaries does a better job of summarizing the most \
important points in the given forum post, without including unimportant or \
irrelevant details? A good summary is both precise and concise.

Post:
<post>

Summary A:
<Summary A>

Summary B:
<Summary B>

FIRST provide a one-sentence comparison of the two summaries, explaining which \
you prefer and why. SECOND, on a new line, state only "A" or "B" to indicate your \
choice. Your response should use the format:
Comparison: <one-sentence comparison and explanation>
Preferred: <"A" or "B">
```

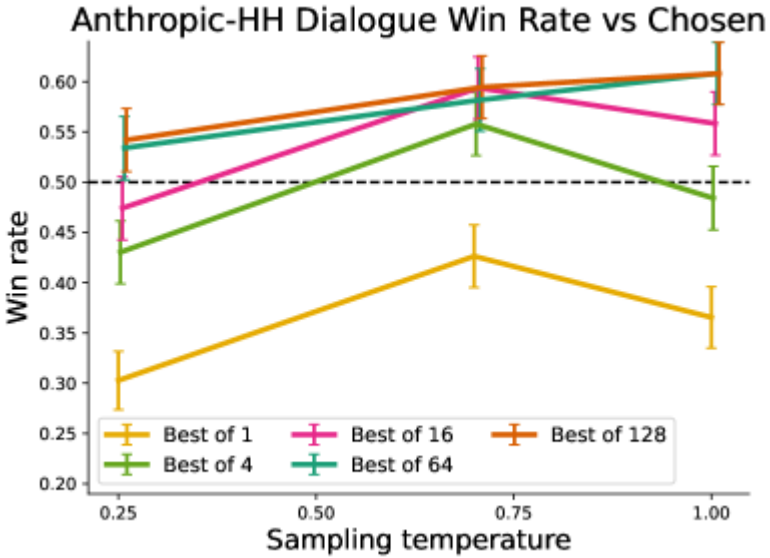
Approach	Core idea / objective	Training signal / data	Where used (as described)	Key notes / trade-offs
Preferred-FT	Supervised FT on chosen completion (y_w)	Preference data mapped to chosen outputs	Controlled Sentiment + Summarization: (y_w) from SFT; Dialogue: (y_w) from generic LM	Uses only the "winner," ignores explicit contrast with rejected
Unlikelihood	Maximize p(y_w), minimize p(y_l)	Preference pairs (y_w, y_l)	Evaluated as baseline	Optional weight α in [0,1]) controls strength of unlikelihood term
PPO (learned reward)	RL optimization of policy	Reward model learned from preference data	Evaluated as baseline	PPO stability/hyperparams matter
PPO-GT (oracle reward)	PPO with ground-truth reward	Ground-truth reward function	Controlled sentiment setting	1) off-the-shelf version 2) improved (reward normalization + tuned hyperparams)
Best-of-(N)	Sample (N) outputs and select best by reward	Learned reward model for scoring	Sample from SFT (or Preferred-FT in dialogue)	Decouples reward model from PPO training, **computationally heavy** (N samples per query at test time)

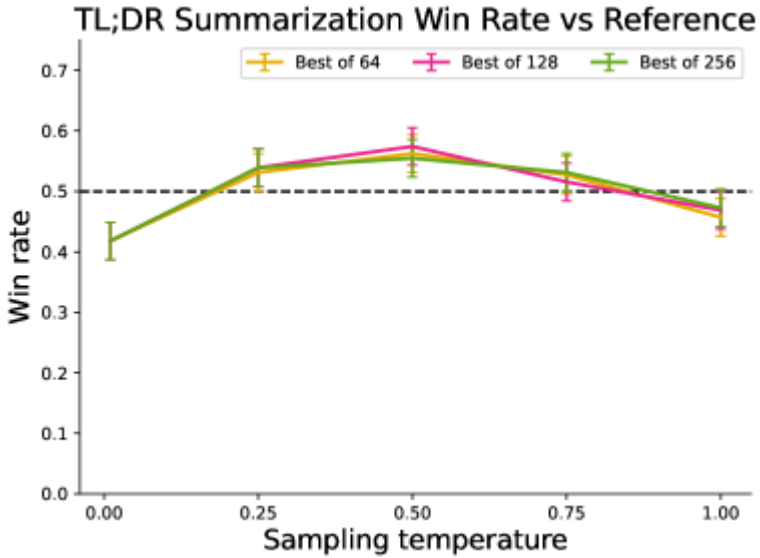
RLHF Baselines: Standard PPO with learned rewards, and an oracle PPO-GT (Ground Truth) which learns directly from the underlying sentiment classifier.

Supervised & Search Baselines: Preferred-FT (supervised fine-tuning on chosen completions), Unlikelihood training (with α∈), and computationally heavy "Best of N" sampling under a learned reward function.

Best-of-N: sample N candidates from SFT (or Preferred-FT in dialogue) and pick the highest-reward one using the learned reward model; strong but expensive because it needs Nsamples per query at test time.

We find that the Best of N baseline is a strong (although computationally expensive, requiring sampling many times) baseline in our experiments.



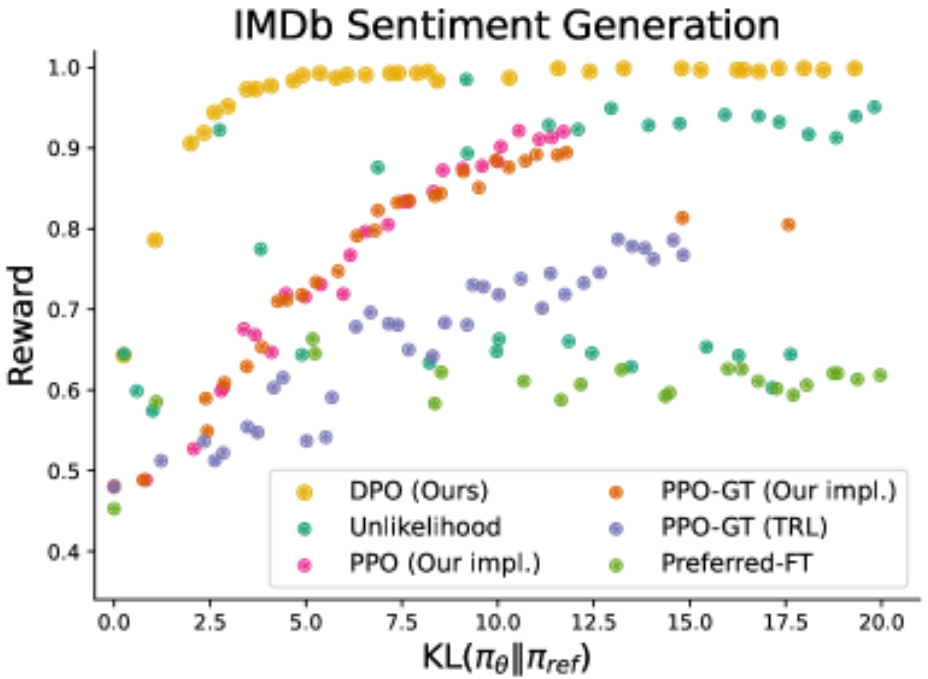


Results

DPO Dominates the Reward-KL Frontier

Experimental Hyperparameters: Evaluated across 22 runs sweeping policy conservativeness (Target KL $\in\{3,6,9,12\}$ for PPO; $\beta\in\{0.05,0.1,1,5\}$ for DPO; $\alpha\in\{0.05,0.1,0.5,1\}$ for unlikelihood, random seeds for preferred-FT). 22 runs in total. Models evaluated every 100 training steps.

- **Dominating Efficiency:** DPO provides the most efficient optimization frontier, achieving the maximum expected reward while maintaining a strictly lower sequence-level KL divergence than PPO.
- **Beating the Oracle:** DPO's reward/KL tradeoff not only strictly dominates standard PPO but also outperforms PPO-GT (the oracle with access to ground-truth rewards).

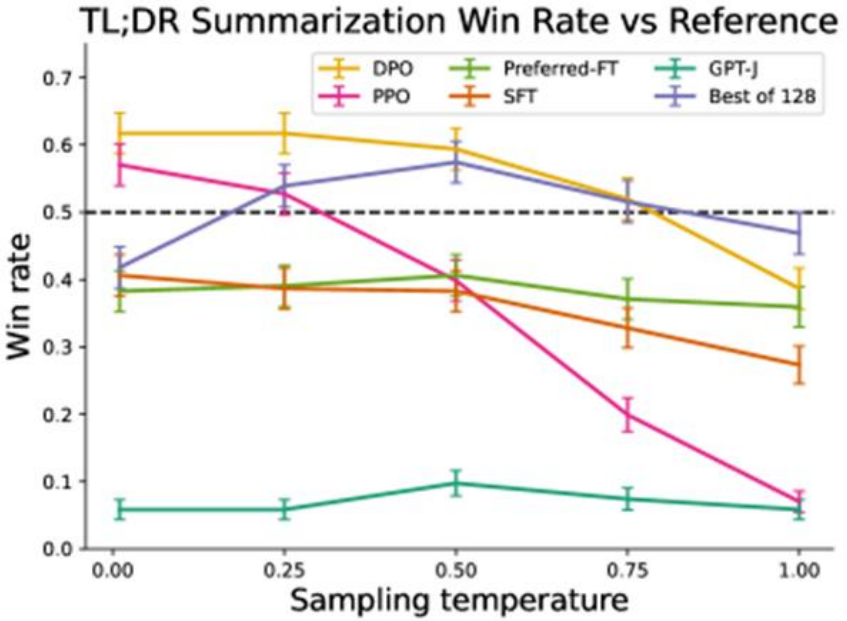


Scaling to Real Preference Datasets

Real-World Performance on Summarization

Summarization (TL;DR): DPO achieves a ≈61% win rate at temperature T=0.0, strictly exceeding PPO's peak performance of 57%. DPO maintains high win rates even at T=1.0, whereas PPO severely degrades.

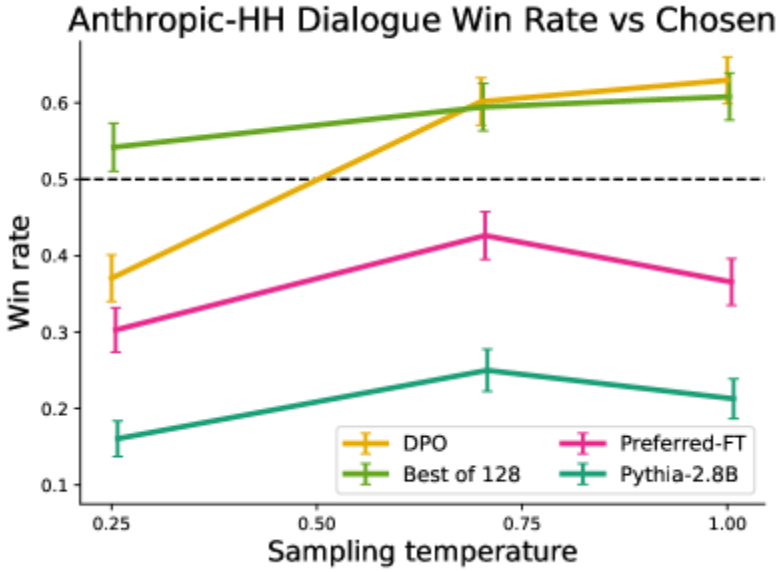
- DPO, PPO and Preferred-FT all fine-tune the same GPT-J SFT model
- DPO's β hyperparameter is not carefully fine-tuned, so results may underestimate DPO's potential
- DPO (T=0.25) was preferred **58%** of the time over PPO (T=0) in human evaluations

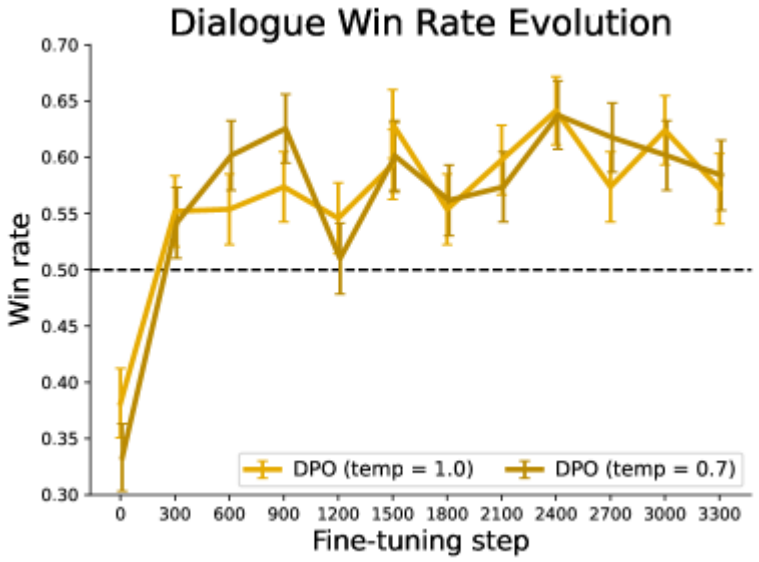


Scaling to Real Preference Datasets

Real-World Performance on HH Dialogue

- **HH Dialogue:** DPO is the only computationally efficient method that improves over the preferred dataset completions.
- DPO on Pythia-2.8B performs similarly or better than the highly expensive "Best of 128" sampling baseline. Base PPO failed to out-perform the Pythia-2.8B base model entirely.
- DPO removes the necessity for generation during training, resulting in lower VRAM consumption and **faster wall-clock convergence** times per step.





Out-of-Distribution Generalization

Generalization to New Input Distributions (Zero-Shot)

- **Task Setup:** Policies trained on Reddit posts (TL;DR) but evaluated **zero-shot** on news articles from the CNN/DailyMail, which is a **distribution shift**.

- **Implication:** DPO policies generalize out-of-distribution (OOD) similarly to, or better than, PPO, despite DPO lacking the unlabelled prompt exploration phase used by PPO.

- **Caveat:** Win rates are overall low(<0.4) and the gap is modest, so this result should be read as a directional trend rather than a definitive advantage.

	Win rate vs. ground truth	
Alg.	Temp 0	Temp 0.25
DPO	0.36	0.31
PPO	0.26	0.23

Table 1: GPT-4 win rates vs. ground truth summaries for out-of-distribution CNN/DailyMail input articles.

Conclusion

This paper triggered a paradigm shift in LLM alignment.

Before DPO: Training well-aligned LLMs with RLHF was an engineering nightmare even for big tech: unstable optimization, hard-to-tune training dynamics, and heavy compute/memory requirements.

DPO provides **a stable, high-performing, and computationally lightweight alternative to RLHF**. It dramatically lowers the barrier to alignment, enabling academics and open-source teams to match ChatGPT-style preferences using a standard supervised learning pipeline.

Limitations & Future Work

- **OOD generalization**: How well does DPO generalize out-of-distribution compared with explicit reward-based RL methods? Early results suggest similarity to PPO, but more study is needed.
- **Self-labeling / unlabeled prompts**: Can DPO leverage unlabeled prompts via self-labeling from the current policy effectively?
- **Scaling up**: They only test up to **~6B** parameters; scaling DPO to much larger SOTA models is an important next step.
- **Evaluation sensitivity**: GPT-4 win rates depend on the **prompt**; future work should improve how automated judges are elicited and validated.
- **Beyond LMs**: DPO may apply beyond LMs, including generative models in other modalities.

Take-home Messages

Theoretical Paradigm Shift: Bypassing Actor-Critic

- **Key shift**: DPO trains directly on **pairwise preferences (winner vs loser)** instead of learning a unary reward score and running RL (PPO).
- **Implicit Reward Modeling:** A language model inherently contains a value function if parameter bounds are properly mathematically constrained.
- **Loss Function Innovation:** DPO proves that maximizing the relative log-probabilities of y_{chosen} over $y_{rejected}$ scaled by the reference model $\log \frac{\pi(y|x)}{\pi_{ref}(y|x)}$ is mathematically equivalent to optimizing a complex **Actor-Critic** RL environment.
- **Gradient Dynamics:** DPO's gradient naturally scales the update magnitude by how incorrectly the model currently estimates the preference, acting as a self-correcting dynamic learning rate.

Take-home Messages

Practical Implications: Democratizing LLM Alignment

- **Practical benefit**: **No reward-model-to-policy RL loop**, fewer sensitive hyperparameters, typically more stable and compute-friendly alignment.
- **Hardware Democratization:** Removes the need to load **4** distinct models (**Policy, Value, Reference, Reward**) into VRAM. DPO requires only **2** (**Policy, Reference**), cutting hardware requirements drastically.
- **Pipeline Simplification:** No generation/sampling required during training step -> Massive speedups in wall-clock training time.
- **Future Directions:** Extending DPO to Plackett-Luce models (handling >2 ranked responses), integrating explicit length-penalties, and adapting DPO for non-language domains like diffusion models.

Thanks!

Q & A

Derivation: DPO Objective

It is straightforward to derive the DPO objective under the Bradley-Terry preference model as we have

$$p^*(y_1 \succ y_2|x) = \frac{\exp\left(r^*(x, y_1)\right)}{\exp\left(r^*(x, y_1)\right) + \exp\left(r^*(x, y_2)\right)} \tag{16}$$

In Section 4 we showed that we can express the (unavailable) ground-truth reward through its corresponding optimal policy:

$$r^*(x, y) = \beta \log \frac{\pi^*(y|x)}{\pi_{\text{ref}}(y|x)} + \beta \log Z(x) \tag{17}$$

Substituting Eq. 17 into Eq. 16 we obtain:

$$\begin{aligned} p^*(y_1 \succ y_2|x) &= \frac{\exp\left(\beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)} + \beta \log Z(x)\right)}{\exp\left(\beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)} + \beta \log Z(x)\right) + \exp\left(\beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)} + \beta \log Z(x)\right)} \\ &= \frac{1}{1 + \exp\left(\beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)} - \beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)}\right)} \\ &= \sigma\left(\beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)} - \beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)}\right). \end{aligned}$$

Lemma 1 Proof

Lemma 1 Restated. *Under the Plackett-Luce preference framework, and in particular the Bradley-Terry framework, two reward functions from the same equivalence class induce the same preference distribution.*

Proof. We say that two reward functions $r(x, y)$ and $r'(x, y)$ are from the same equivalence class if $r'(x, y) = r(x, y) + f(x)$ for some function f. We consider the general Plackett-Luce (with the Bradley-Terry model a special case for $K = 2$) and denote the probability distribution over rankings induced by a particular reward function $r(x, y)$ as p_r. For any prompt x, answers $y_1, \ldots, y_K$ and ranking τ we have:

$$
\begin{aligned}
p_{r'}(\tau|y_1, \ldots, y_K, x) &= \prod_{k=1}^{K} \frac{\exp(r'(x, y_{\tau(k)}))}{\sum_{j=k}^{K} \exp(r'(x, y_{\tau(j)}))} \\
&= \prod_{k=1}^{K} \frac{\exp(r(x, y_{\tau(k)}) + f(x))}{\sum_{j=k}^{K} \exp(r(x, y_{\tau(j)}) + f(x))} \\
&= \prod_{k=1}^{K} \frac{\exp(f(x)) \exp(r(x, y_{\tau(k)}))}{\exp(f(x)) \sum_{j=k}^{K} \exp(r(x, y_{\tau(j)}))} \\
&= \prod_{k=1}^{K} \frac{\exp(r(x, y_{\tau(k)}))}{\sum_{j=k}^{K} \exp(r(x, y_{\tau(j)}))} \\
&= p_r(\tau|y_1, \ldots, y_K, x),
\end{aligned}
$$

which completes the proof. □

Lemma 2 Proof

Lemma 2 Restated. *Two reward functions from the same equivalence class induce the same optimal policy under the constrained RL problem.*

Proof. Let us consider two reward functions from the same class, such that $r'(x, y) = r(x, y) + f(x)$ and, let us denote as π_r and $\pi_{r'}$ the corresponding optimal policies. By Eq. 4, for all x, y we have

$$
\begin{aligned}
\pi_{r'}(y|x) &= \frac{1}{\sum_y \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r'(x, y)\right)} \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r'(x, y)\right) \\
&= \frac{1}{\sum_y \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta}(r(x, y) + f(x))\right)} \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta}(r(x, y) + f(x))\right) \\
&= \frac{1}{\exp\left(\frac{1}{\beta} f(x)\right) \sum_y \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x, y)\right)} \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x, y)\right) \exp\left(\frac{1}{\beta} f(x)\right) \\
&= \frac{1}{\sum_y \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x, y)\right)} \pi_{\text{ref}}(y|x) \exp\left(\frac{1}{\beta} r(x, y)\right) \\
&= \pi_r(y|x),
\end{aligned}
$$

which completes the proof. □