Your Language Model is Secretly a Reward Model

Direct Preference Optimization

Rafael Rafailov[*†] **Archit Sharma**[*†] **Eric Mitchell**[*†]

Stefano Ermon[†‡] **Christopher D. Manning**[†] **Chelsea Finn**[†]

[†]Stanford University [‡]CZ Biohub
`{rafailov,architsh,eric.mitchell}@cs.stanford.edu`

HUMANE Lab 김건수

2025.03.21 랩세미나

Introduction

- Large-scale Language Models (LLMs) acquire knowledge but lack precise co ntrollability.
- RLHF (Reinforcement Learning from Human Feedback) is widely used but ha s challenges.
- This research proposes Direct Preference Optimization (DPO) to simplify alig nment.

RLHF Overview

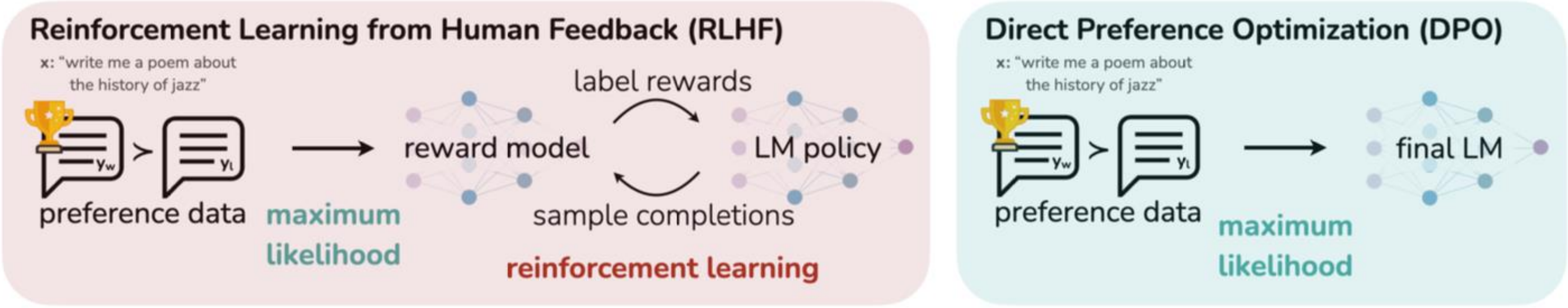
- RLHF is a method to fine-tune large language models using human feedback.
- It contains three main steps:
  - Supervised Fine-Tuning(SFT): The model is pre-trained on high-quality human response s.
  - Reward Model Training: A reward model is trained to predict human preferences.
  - Reinforcement Learning: The model is fine-tuned using the reward model and Proximal Policy Optimization(PPO)

Challenges in RLHF

- Complexity: Requires multiple models (LLM, reward model, RL).
- Instability: RLHF can lead to unpredictable results.
- High Cost: Sampling-based reinforcement learning is computationally expensive.
- Difficult Hyperparameter Tuning: PPO needs careful KL control.

What is DPO?

- DPO (Direct Preference Optimization) eliminates RL from the preference optimization pipeline.
- Optimizes policy directly using a **classification loss**.
- More stable, computationally efficient, and easier to train.



DPO Methodology

1. Collect human preference data.
2. Apply the Bradley-Terry model for preference ordering.
3. Optimize using a simple cross-entropy loss.
4. No explicit reward modeling or reinforcement learning needed.

Key Equations in DPO

1. Optimal policy under KL-Constrained reward maximization

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

2. Reward function reparameterization

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

3. Optimal policy in Bradley-Terry framework

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

4. DPO loss function

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

Theoretical Analysis of DPO

- Your language model is secretely a Reward Model
  - DPO eliminates explicit reward modeling by reparameterizing rewards in terms of policy ratios:

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

  - This formulation allows DPO to learn the optimal policy without reinforcement learning.
  - The model itself inherently encodes the rewards, making explicit reward modeling unne cessary.

Equivalence of Reward Functions

- Definition: Two reward functions: $r(x, y)$ and $r'(x, y)$ are equivalent if:
- $r(x,y) - r'(x,y) = f(x)$
- For some function $f(x)$ that depends only on x.
- Lemma1: Two equivalent reward functions induce the same preference distribution.
- Lemma2: Equivalent reward functions yield the same optimal policy under KL-constrained RL.

- > DPO only needs to recover any reward functions within the optimal equivalence class

Theorem 1: Generality of DPO Reparameterization

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

- DPO reparameterization is general and preserves the full class of reward models.
- The transformation normalizes rewards using:

$$f(r; \pi_{\text{ref}}, \beta)(x, y) = r(x, y) - \beta \log \sum_{y} \pi_{\text{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right)$$

- -> DPO learns a valid reward function without reinforcement learning.

Why RLHF with PPO is Unstable?

- PPO Objective:

$$\max_{\pi_\theta} \mathbb{E}_{\pi_\theta(y|x)} \left[\underbrace{r_\phi(x,y) - \beta \log \sum_y \pi_{\text{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r_\phi(x,y)\right)}_{f(r_\phi, \pi_{\text{ref}}, \beta)} - \underbrace{\beta \log \frac{\pi_\theta(y \mid x)}{\pi_{\text{ref}}(y \mid x)}}_{\text{KL}} \right]$$

- High variance in policy gradients leads to instability.
- KL divergence tuning is difficult, leading to mode collapse.
- Normalization issues make value function estimation unreliable.

- -> DPO avoids issues by directly optimizing the policy using preference dat a.

Experiments & Results

- Tested on Sentiment Generation, Summarization, and Dialogue Tasks.
- DPO outperforms PPO in reward efficiency and stability.
- DPO achieves better human-aligned responses with lower computational co st.

Comparison: DPO vs PPO

- - Higher reward for lower KL divergence.
- - Better stability in sampling temperature variations.
- - No need for an explicit reward model.

Limitations of DPO

- Cannot leverage explicit reward signals
- Limited to pairwise preference learning
- Inefficient for long Chain-of-Thought(CoT) responses
- Poor generalization to unseen data

Conclusion

- DPO simplifies preference learning without RL.
- It is computationally efficient and easy to implement.
- Future work: Scaling to larger models and applying to other domains.

Appendix: Why Use DAPO

- Maintains RL advantages while improving stability
- Optimized for long responses(CoT learning)
- Better generalization to new tasks
- Can leverage reward models if needed

Scenario	Best Choice	Reason
Simple preference fine-tuning	DPO	No RL required, faster training
Long response (CoT) training	DAPO	Token-level optimization
Generalization explicit reward sig nals	DAPO	RL-based learning improves adaptability
Low compute environments	DAPO	No reinforcement learning needed
Large-scale LLM training	DPO	Optimized RLHF for better performance
Using explicit reward signals	DAPO	Can optimize with reward models

Thank You!

- Q&A