



# Direct Preference Optimization(DPO): Your Language Model is Secretly a Reward Model

-미니 프로젝트 2 논문 발제-

2025.05.15

Team 4: 삼권분립: 고헤정, 신진섭, 홍준기

# Contents



- INTRODUCTION
- BACKGROUND & RELATED WORKS
- PRELIMINARIES
- DIRECT PREFERENCE OPTIMIZATION(DPO)
- UNDERSTANDING DPO
- THEORETICAL ANALYSIS OF DPO
- EXPERIMENTS
- CONCLUSION

## 1. Motivation

- LLM들은 여러가지 목표, 우선순위, skillset 을 가지고 있는 사람들에게 의해서 만들어진 데이터로부터 훈련됨
  - 몇몇 데이터는 훈련되기에 적합하지 않을 수 있음 (Garbage-in-Garbage-out)
- 요구되는 응답과 행동을 만들어내기 위해, 정제되고, 선호되는 조절 가능한 AI system을 만드는 것이 중요함
  - 도메인 특화 AI Chatbot, Human-in-the-loop관점에서의 바람직한 output 생성
- LLM 자체는 넓은 지식을 습득하지만 행동 제어는 어려움

## 2. Previous Method

- SFT(Supervised Fine Tuning) 방식:
  - 훈련될 때 개인 별 전문가의 writing능력에 따른 annotating 문제-> quality문제
  - 그냥 좋은 예시를 따라하는 방식-> 어떠한 reward도 없음
- Reinforcement Learning by Human Feedback(RLHF) 방식:
  - 답변의 Golden label을 제시하는 것보다 “어떤 것이 상대적으로 Preference된다” 라고 Supervision signal 방식이 Demonstration 방식보다는 일관성 있는 annotation을 이끌어낼 수 있다는 데에서 착안됨
  - 복잡성 문제: Reward Model의 학습 + PPO 기반 LM policy 업데이트 + Policy 샘플링 + 하이퍼파라미터 민감 문제

## 3. Summary of DPO

Reinforcement Learning 없이 Preference를 학습하여, Reward 모델 없이도 학습 안정성과 성능을 모두 얻음

## 1. Instruction-Tuning & Human Preference 기반 Fine-Tuning 흐름

Self-supervised LLMs → Instruction-tuning → Human Preference

- 다양한 태스크에서 **zero-shot, few-shot**으로도 놀라운 성능을 보임
- 하지만, 유저가 원하는 방식으로 **정밀하게 행동** 제어하기는 어려움

- "이런 질문에는 이렇게 답해야 해!"
- 명시적으로 시킨 **instruction** 데이터셋으로 **fine-tuning**
- LLM이 훨씬 유용해지고 **alignment**가 증가

- 좋은 응답 vs 나쁜 응답 → **비교 기반 선호 데이터 수집**
- Reward 모델 학습 (Bradley-Terry 모델 기반)
- PPO 등 강화학습으로 **fine-tuning**

**Anthropic Claude**  
**InstructGPT** (OpenAI, 2022)  
**Google Sparrow**  
(DeepMind)

## 2. DPO's State vs Related Works

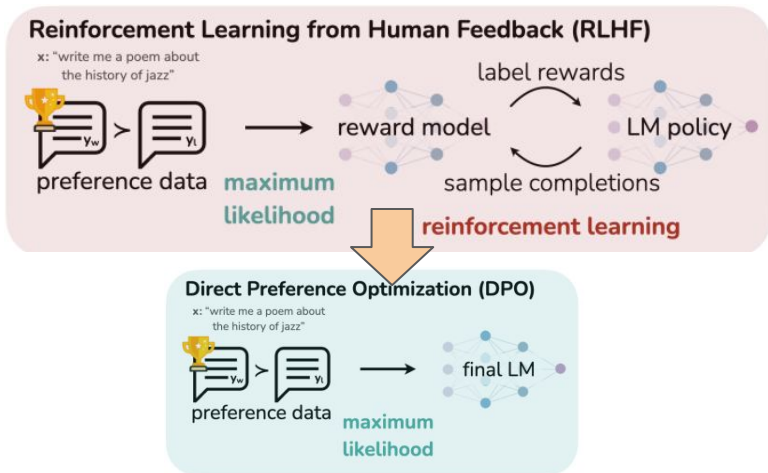
| 분류                 | 기존 방식                                  | 특징                         |
|--------------------|--|----------------------------|
| Instruction Tuning | $(x, y) \rightarrow$ 지도 학습             | 단일 예시 학습, 비교 학습 아님         |
| RLHF (PPO)         | 선호 $\rightarrow$ 보상모델 $\rightarrow$ RL | pipeline 복잡, 학습 비용 많음      |
| <b>DPO</b>         | $(x, y_w, y_l)$ 만으로 바로 최적 policy 학습    | <b>reward</b> 모델, RL 모두 없음 |

# Background



## 1. Basic Terms to understand PPO & DPO

- 강화학습: 총 보상(return)을 최대화하는 정책 학습
  - RLHF에서의 적용:
    - 상태: 입력(prompt), 행동: 응답 (token sequence), 보상: 사람이 더 선호한 응답 → 높은 보상
    - 정책: 언어 모델  $\pi(y/x)$
- Bradley-Terry Model**: 두 응답의 상대 보상 차이로 선호 확률을 모델링하는 확률 모델
- Reward Model**: 사람의 선호(preference)를 학습하여 각 응답에 대해 점수(reward)를 예측하는 모델
  - RLHF에서는 이 reward model이 PPO의 보상 함수 역할을 함
- PPO**: 기존 정책에서 너무 멀어지지 않게 안정적으로 강화학습하는 방식



| Consideration | PPO                        | DPO                        |
|---------------|----------------------------|----------------------------|
| KL 제약         | 명시적으로 필요                   | log ratio 기반 →<br>내재적으로 존재 |
| learning rate | gradient scale<br>커서 튜닝 중요 | BCE 기반 상대적으로<br>안정적        |
| clip range    | 필수적으로 넣어야<br>정책 변화 안정적     | 없음                         |
| gradient 폭주   | 있음                         | 낮음                         |

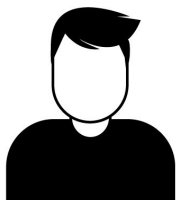
1. **SFT (Supervised Fine-Tuning)**
2. **Reward Modeling**
3. **RL Fine-Tuning**

1. **SFT (Supervised Fine-Tuning)** ← 지도학습을 통한 파인 튜닝

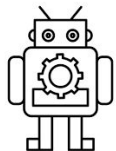


1. SFT (Supervised Fine-Tuning)

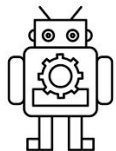
2. **Reward Modeling** ←



→ 모델아 농담 해봐



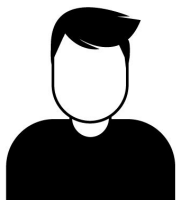
→ 죄송합니다. 농담은 제공할 수 없습니다.



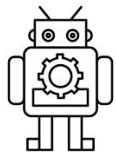
→ 우유가 넘어지면 아야~! 깔깔

1. SFT (Supervised Fine-Tuning)

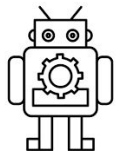
2. **Reward Modeling** ←



→ 모델아 농담 해봐



→ 죄송합니다. 농담은 제공할 수 없습니다.



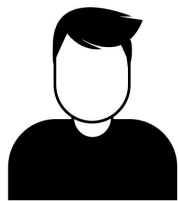
→ 우유가 넘어지면 아야~! 깔깔

## 1. SFT (Supervised Fine-Tuning)

## 2. **Reward Modeling** ←

보상 함수

$$r_{\phi}(x, y)$$

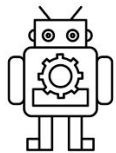


→ 모델아 농담 해봐

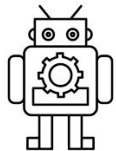
Bradley-Terry 모델

$$p(y_1 \succ y_2 | x) = \sigma(r_{\phi}(x, y_1) - r_{\phi}(x, y_2))$$

→ 두 응답의 보상값 차이를 시그모이드 함수에 넣어 선호확률을 모델링 하는 방식



→ 죄송합니다. 농담은 제공할 수 없습니다.



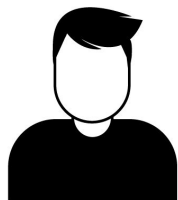
→ 우유가 넘어지면 아야~! 깔깔

## 1. SFT (Supervised Fine-Tuning)

## 2. **Reward Modeling** ←

보상 함수

$$r_{\phi}(x, y)$$

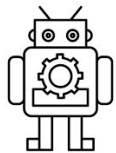


→ 모델아 농담 해봐

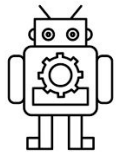
Bradley-Terry 모델

$$p(y_1 \succ y_2 | x) = \sigma(r_{\phi}(x, y_1) - r_{\phi}(x, y_2))$$

→ 두 응답의 보상값 차이를 시그모이드 함수에 넣어 선호확률을 모델링 하는 방식



→ 죄송합니다. 농담은 제공할 수 없습니다.



→ 우유가 넘어지면 아야~! 깔깔

보상 함수가 무엇인지를 배우는 과정

1. SFT (Supervised Fine-Tuning)
2. Reward Modeling
3. **RL Fine-Tuning** ←

$$\max_{\pi_{\theta}} \mathbb{E}_{x, y \sim \pi_{\theta}} [r_{\phi}(x, y)] - \beta D_{\text{KL}} [\pi_{\theta}(y|x) \parallel \pi_{\text{ref}}(y|x)]$$

↑  
보상이 높은 응답을 생성하도록 학습

↑  
기존 모델과 너무 멀어지지 않게 제한

# Direct Preference Optimization (DPO)



## DPO의 목적

→ 좋은 응답의 확률을 높이고 나쁜 응답의 확률을 낮추자

### 기존의 강화학습

질문 → 응답 → 보상 → 강화학습

### DPO

질문 + 선호 응답쌍 → 직접 학습



$(x, y_w, y_l)$

사람이 선택한 좋은 응답과 나쁜 응답을 한 쌍으로 만든 것

# Direct Preference Optimization (DPO)

## DPO의 목적

→ 좋은 응답의 확률을 높이고 나쁜 응답의 확률을 낮추자

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

$$\quad \quad \quad \searrow \quad z = \beta \left( \log \frac{\pi_{\theta}(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \log \frac{\pi_{\theta}(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right) = \beta \log \left( \frac{\pi_{\theta}(y_w|x)/\pi_{\text{ref}}(y_w|x)}{\pi_{\theta}(y_l|x)/\pi_{\text{ref}}(y_l|x)} \right)$$

1.  $y_w$ : 사람이 더 선호한 응답

2.  $y_l$ : 사람이 덜 선호한 응답

3.  $\pi_{\theta}$ : 현재 학습 중인 모델의 확률 분포

$\pi_{\text{ref}}$ : 기준이 되는 SFT 모델의 확률 분포

⇒ 좋은 답변의 상대적인 log 확률 비율이 나쁜 응답보다 더 크도록 유도

이걸 시그모이드 함수에 넣어서 **이진 분류 문제**처럼 학습함

# Direct Preference Optimization (DPO)

## DPO의 목적

→ 좋은 응답의 확률을 높이고 나쁜 응답의 확률을 낮추자

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

$$\downarrow$$

$$z = \beta \left( \log \frac{\pi_{\theta}(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \log \frac{\pi_{\theta}(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right) = \beta \log \left( \frac{\pi_{\theta}(y_w|x)/\pi_{\text{ref}}(y_w|x)}{\pi_{\theta}(y_l|x)/\pi_{\text{ref}}(y_l|x)} \right)$$

1.  $y_w$ : 사람이 더 선호한 응답

2.  $y_l$ : 사람이 덜 선호한 응답

3.  $\pi_{\theta}$ : 현재 학습 중인 모델의 확률 분포

$\pi_{\text{ref}}$ : 기준이 되는 SFT 모델의 확률 분포

⇒ 좋은 답변의 상대적인 log 확률 비율이 나쁜 응답보다 더 크도록 유도

이걸 시그모이드 함수에 넣어서 **이진 분류 문제**처럼 학습함

**log-sigmoid를 취해서 binary cross-entropy loss 형태로 최적화**



# Understanding DPO



Q. 왜 실제로도 잘 작동하는지 + 어떻게 기존 RLHF의 학습 원리를 내포하고 있는지

$$\mathcal{L}_{\text{DPO}} = -\log \sigma \left( \beta \cdot \log \frac{\pi_{\theta}(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \cdot \log \frac{\pi_{\theta}(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right)$$

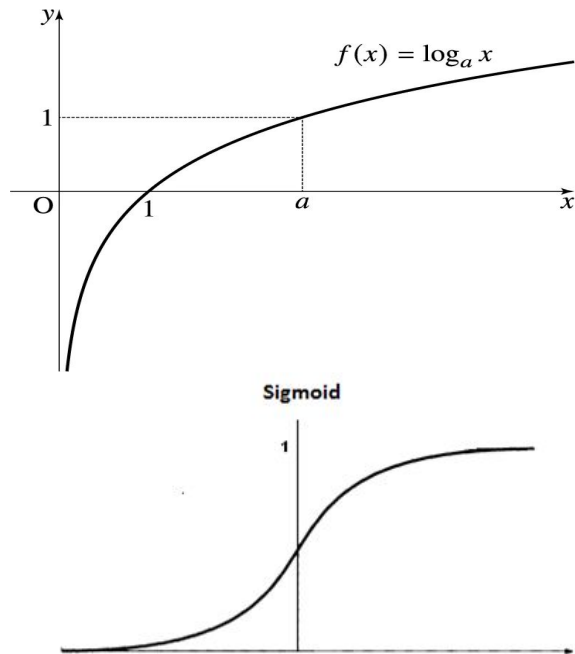
$\pi_{\theta}(y|x)$  : 지금 학습 중인 내 모델이 응답  $y$ 를 선택할 확률

$\pi_{\text{ref}}(y|x)$  : 기준이 되는 SFT 모델이 응답  $y$ 를 선택할 확률

좋은 응답은 SFT 모델보다 더 높은 확률  
나쁜 응답은 SFT 모델보다 더 낮은 확률

⇒ 좋은 응답의 log 비율은 올리고, 나쁜 응답은 빼는 형태

⇒ 시그모이드 함수에 넣어 그 확률이 1에 가까워지도록 학습



# Understanding DPO



Q. 왜 실제로도 잘 작동하는지 + 어떻게 기존 RLHF의 학습 원리를 내포하고 있는지

$$\mathcal{L}_{\text{DPO}} = -\log \sigma \left( \beta \cdot \log \frac{\pi_{\theta}(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \cdot \log \frac{\pi_{\theta}(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right)$$

$\pi_{\theta}(y|x)$  : 지금 학습 중인 내 모델이 응답  $y$ 를 선택할 확률

$\pi_{\text{ref}}(y|x)$  : 기준이 되는 SFT 모델이 응답  $y$ 를 선택할 확률

좋은 응답은 SFT 모델보다 더 높은 확률

나쁜 응답은 SFT 모델보다 더 낮은 확률

⇒ 좋은 응답의 log 비율은 올리고, 나쁜 응답은 빼는 형태

⇒ 시그모이드 함수에 넣어 그 확률이 1에 가까워지도록 학습

$$\max_{\pi_{\theta}} \mathbb{E}_{x,y \sim \pi_{\theta}} [r_{\phi}(x,y)] - \beta \cdot D_{\text{KL}} [\pi_{\theta} \parallel \pi_{\text{ref}}]$$

**KL-divergence 항이 없는데 ???**

DPO는?  $\log \frac{\pi_{\theta}(y|x)}{\pi_{\text{ref}}(y|x)}$  가 있다

⇒ KL-divergence의 미분 형태와 유사

$$\nabla_{\theta} D_{\text{KL}} [\pi_{\theta} \parallel \pi_{\text{ref}}] = \mathbb{E}_{y \sim \pi_{\theta}} [\nabla \log \pi_{\theta}(y|x) - \nabla \log \pi_{\text{ref}}(y|x)]$$

# Understanding DPO



Q. 왜 실제로도 잘 작동하는지 + 어떻게 기존 RLHF의 학습 원리를 내포하고 있는지

$$\mathcal{L}_{\text{DPO}} = -\log \sigma \left( \beta \cdot \log \frac{\pi_{\theta}(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \cdot \log \frac{\pi_{\theta}(y_l|x)}{\pi_{\text{ref}}(y_l|x)} \right)$$

$$\max_{\pi_{\theta}} \mathbb{E}_{x,y \sim \pi_{\theta}} [r_{\phi}(x, y)] - \beta \cdot D_{\text{KL}} [\pi_{\theta} \parallel \pi_{\text{ref}}]$$

$\pi_{\theta}(y|x)$  : 지금 학습 중인 내 모델이 응답  $y$ 를 선택할 확률

$\pi_{\text{ref}}(y|x)$  : 기준이 되는 SFT 모델이 응답  $y$ 를 선택할 확률

좋은 응답은 SFT 모델보다 더 높은 확률  
나쁜 응답은 SFT 모델보다 더 낮은 확률

⇒ 좋은 응답의 log 비율은 올리고, 나쁜 응답은 빼는 형태

⇒ 시그모이드 함수에 넣어 그 확률이 1에 가까워지도록 학습

**KL-divergence 항이 없는데 ???**

DPO는?  $\log \frac{\pi_{\theta}(y|x)}{\pi_{\text{ref}}(y|x)}$  가 있다

⇒ KL-divergence의 미분 형태와 유사

$$\nabla_{\theta} D_{\text{KL}} [\pi_{\theta} \parallel \pi_{\text{ref}}] = \mathbb{E}_{y \sim \pi_{\theta}} [\nabla \log \pi_{\theta}(y|x) - \nabla \log \pi_{\text{ref}}(y|x)]$$

**Gradient 구조**

$$\nabla_{\theta} \mathcal{L}_{\text{DPO}} \propto \sigma'(z) \cdot [\nabla_{\theta} \log \pi_{\theta}(y_l|x) - \nabla_{\theta} \log \pi_{\theta}(y_w|x)]$$

# Method: DPO

$\pi$  (LLM foundation)

SFT (Supervised fine-tuning)

$\leftarrow \{(X_{SFT}, Y_{SFT}), \dots\}$  high-quality prompt &



$\pi^{SFT}$

## Preference sampling

**Response pair generation**  $(y_1, y_2) \sim \pi^{SFT}(y | x) \leftarrow x$ : 임의의 prompt

**Human preference annotation**  $\mathcal{D} = \{x^{(i)}, y_w^{(i)}, y_l^{(i)}\}_{i=1}^N \sim p^* \leftarrow p^*$ : 사람들의 preference distribution (즉,  $y_w \succ y_l | x \sim r^*(y, x)$ )

## Reward learning

**Reward modeling** (ft. Bradley-Terry (BT) model)

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

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

$\leftarrow$  binary classification loss.

$r_\phi(x, y)$ 는 주로  $\pi^{SFT}$  위에 linear layer 를 있어서 초기화.

## Reinforcement-learning (RL) optimization

**Objective function** (constrained reward maximization) Reference 용도 ( $\pi^{SFT}$ )

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

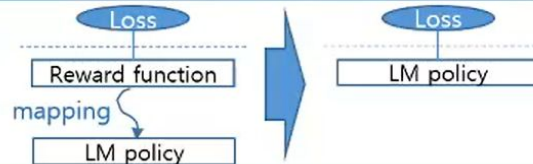
reward model  
따르도록.

초심(?)을 잃지 않게.  
(너무 reward 쪽으로 치우치면  
diversity 가 훼손될 수 있어서)

**RL optimization** (ft. PPO algorithm, ...)

## Idea

objective function 에서  
reward function 을 없애므로써  
reward learning 과정을  
없애고, RL objective  
대신 binary classification  
loss 로써 LM policy 학습!



## DPO formulation

**Objective function**

$$\pi_r(y | x) = \frac{1}{Z(x)} \pi_{ref}(y | x) \exp\left(\frac{1}{\beta} r(x, y)\right) \leftarrow \text{Optimal solution, where } Z(x) = \sum_y \pi_{ref}(y | x) \exp\left(\frac{1}{\beta} r(x, y)\right)$$

$$r(x, y) = \beta \log \frac{\pi_r(y | x)}{\pi_{ref}(y | x)} + \beta \log Z(x) \leftarrow \text{Optimal solution 에 대한 reward}$$

**Binary classification loss**

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

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

# Theoretical Analysis of DPO

Definition 1.

두 보상 함수  $r(x, y)$ 와  $r'(x, y)$ 가 어떤 함수  $f(x)$ 에 대해

$$r(x, y) - r'(x, y) = f(x)$$

가 성립하면, 우리는  $r$ 과  $r'$ 을 동치(equivalent)라고 말합니다.

Lemma 1.

Plackett–Luce, 특히 Bradley–Terry 선호(preference) 프레임워크에서, 같은 동치 클래스에 속하는 두 보상 함수는 동일한 선호 분포(preference distribution)를 유도합니다.

Lemma 2.

동일한 동치 클래스에 속하는 두 보상 함수는 제약된 강화학습 문제(constrained RL problem) 하에서 동일한 최적 정책(restricted optimal policy)을 유도합니다.

# Theoretical Analysis of DPO



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

# Theoretical Analysis of DPO



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



**Theorem 1.** *Under mild assumptions, all reward classes consistent with the Plackett-Luce (and Bradley-Terry in particular) models can be represented with the reparameterization  $r(x, y) = \beta \log \frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)}$  for some model  $\pi(y | x)$  and a given reference model  $\pi_{\text{ref}}(y | x)$ .*

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

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

$$r(x, y) = \beta \log \frac{\pi_r(y | x)}{\pi_{\text{ref}}(y | x)} + \beta \log Z(x) \qquad Z(x) = \sum_y \pi_{\text{ref}}(y | x) \exp \left( \frac{1}{\beta} r(x, y) \right)$$

$$r'(x, y) = f(r, \pi_{\text{ref}}, \beta)(x, y) = r(x, y) - \beta \log Z(x)$$

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



## 5.1 Your Language Model Is Secretly a Reward Model


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

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

DPO는 명시적인 보상 함수를 학습하고 강화학습을 수행하여 정책을 익히는 과정을, 단일한 maximum likelihood objective만으로 대체할 수 있다.

## 5.2 Instability of Actor-Critic Algorithms

DPO can be used to diagnose instabilities of actor-critic algorithms. Connecting the PPO objective to the DPO optimal policy leads to:


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

# Theoretical Analysis of DPO



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

첫 번째 괄호 안의 항은 보상 함수  $r_{\text{phi}}$ 에 대한 DPO 등가 보상으로,

그 아래 첨자  $f(r_{\text{phi}}, \pi_{\text{ref}}, \beta)$ 는 기준 정책  $\pi_{\text{ref}}$ 의 "소프트 값 함수(soft value function)"에 해당하는 정규화 항을 나타낸다.

정규화 항은 최적 해에는 영향을 주지 않지만, 이를 생략하면 분산이 커져 학습이 불안정

# Theoretical Analysis of DPO



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

$$\nabla_{\theta} J(\theta) = \mathbb{E} \left[ (R - V^{\pi}(x)) \nabla_{\theta} \log \pi_{\theta}(y | x) \right] \quad V^{\pi}(x) \text{가 baseline 역할}$$

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

어떤 baseline도 직접 설계하거나 추정할 필요 없이, 보상 함수 자체에 정규화 항이 내장

# Experiments



| Task                              | (x, y)  | Dataset   | Human preference dataset   | SFT   | Evaluation   |
|-----------------------------------|---|---|--|---|--|
| "Controlled" Sentiment generation | <b>x</b> : movie review prefix<br><b>y</b> : completion text (positive) | IMDb dataset (movie review)                     | LM이 생성한 2개의 텍스트에 대해 학습 완료된 Sentiment classifier 'S'로 preference label 생성 | GPT-2-large (training set으로 fine-tuning)          | 'S'로 reward 측정 가능<br>·Reward vs. KL constraint 간 frontier plot                       |
| Summarization                     | <b>x</b> : forum post<br><b>y</b> : summary text                        | Reddit TL;DR summarization dataset              | Stiennon et al. 의 라벨 사용 (LM이 생성한 2개의 텍스트에 대해 human-annotated label 구축)   | GPT-J (human-written summary로 fine-tuning)        | GPT-4 judge로 summarization quality 기준 win-rate 측정<br>기준: test set의 reference summary |
| Single-turn dialog                | <b>x</b> : user query<br><b>y</b> : response                            | Anthropic Helpful and Harmless dialogue dataset | LM이 생성한 2개의 텍스트에 대해 human-annotated label 제공                             | Pythia-2.8B (preferred completions으로 fine-tuning) | GPT-4 judge로 Helpfulness 기준 win-rate 측정<br>기준: test set의 preferred response          |

# Experiments

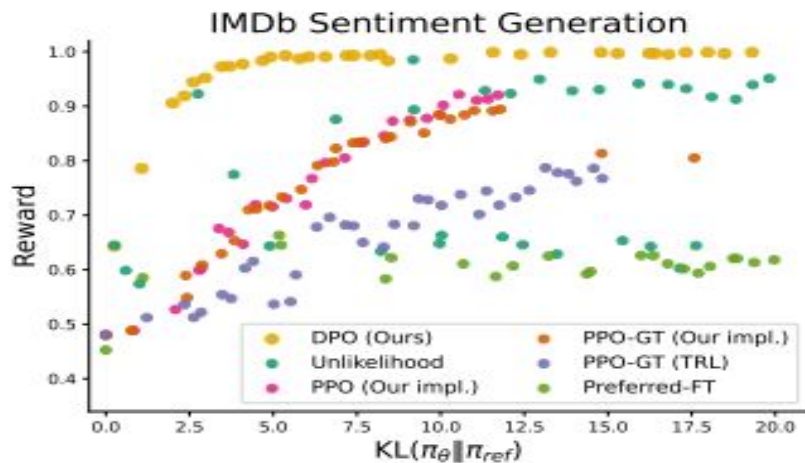


| 분류           | 기법                 | 설명  |
|--------------|--------------------|---|
| 제로샷/소수샷 프롬프트 | GPT-J 0-shot       | Summarization task에 사용  |
|              | Pythia-2.8B 2-shot | Dialog task에 사용   |
| 지도학습 기반 모델   | SFT                | 일반적인 최대우도추정 방식의 fine-tuning   |
|              | Preferred-FT       | 올은 응답 $y_w$ 에 대해서만 학습   |
|              | Unlikelihood       | $y_w$ 확률을 최대화하고, $y_l$ 확률을 최소화하도록 negative-sampling loss를 추가한 fine-tuning     |
| 강화학습 기반 모델   | PPO                | Proximal Policy Optimization  |
|              | PPO-GT             | Oracle로서 ground-truth reward function을 직접 학습 controlled task에서만 가             |
| 추론 시 선택 기법   | Best of N          | SFT 모델로 N개 text 생성 후, preference data로 학습된 reward 함수로 점수를 매겨 최고점 응답을 output으로 |

# Experiments



| Task                                    | (x, y)   | Dataset                        | Human preference dataset  | SFT   | Evaluation   |
|---|--|--------------------------------|---|---|--|
| "Controlled"<br>Sentiment<br>generation | x: movie review prefix<br>y: completion text<br>(positive) | IMDb dataset<br>(movie review) | LM이 생성한 2개의 text에<br>대해, 학습 완료된<br>Sentiment classifier 'S'로<br>preference label 생성 | GPT-2-large (fine-tuned<br>with training set) | 'S'가 있으니 reward 생성 &<br>KL constraint가 frontier plot |



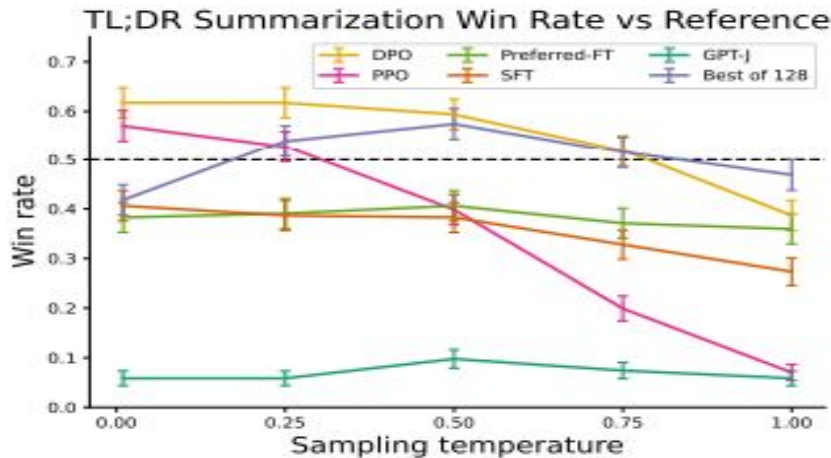
1. DPO와 PPO는 동일한 목적을 최적화하지만, DPO가 훨씬 더 효율적이다.
2. PPO가 실제(reward-GT)에 접근할 수 있는 경우(PPO-GT)에도 불구하고, DPO는 PPO보다 더 나은 프런티어를 달성한다.



# Experiments



| Task          | (x, y)                           | Dataset                            | Human preference dataset   | SFT   | Evaluation   |
|---------------|----------------------------------|------------------------------------|--|---|--|
| Summarization | x: forum post<br>y: summary text | Reddit TL;DR summarization dataset | Stiennon et al. 외 label 사용 (LM이 생성한 2개의 text에 대해 human-annotated label 구축) | GPT-J (fine-tuned with human-written summary) | Judge(GPT-4)를 통한 summarization quality 기준으로 win rate (baseline: test set의 reference summary) |

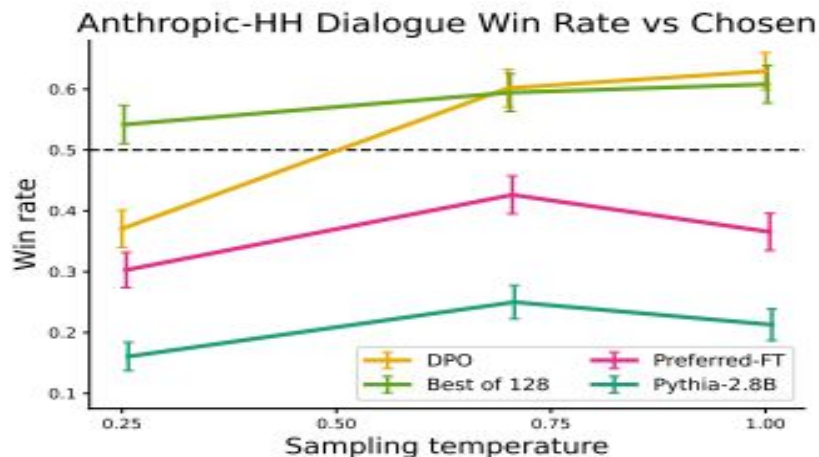




# Experiments



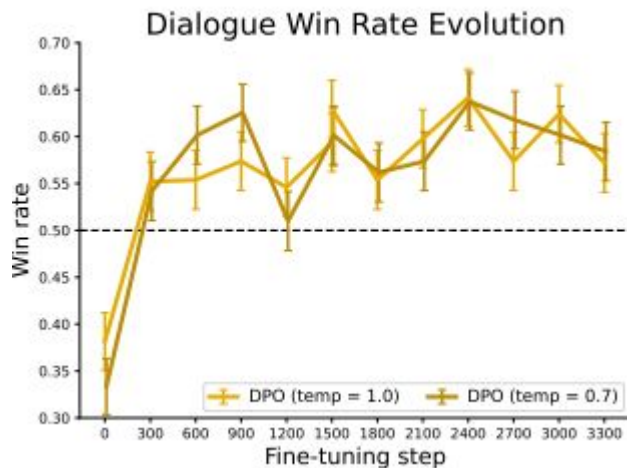
| Task               | (x, y)                       | Dataset   | Human preference dataset                       | SFT  | Evaluation  |
|--------------------|------------------------------|---|--|--|---|
| Single-turn dialog | x: user query<br>y: response | Anthropic Helpful and Harmless dialogue dataset | LM이 생성한 2개의 text에 대해 human-annotated label 제공됨 | Pythia-2.8B (fine-tuned with preference completions) | Judge(GPT-4)를 통한 Helpfulness 기준으로 win rate (baseline: test set의 preferred response) |



# Experiments



DPO's improvement over the dataset labels is fairly stable



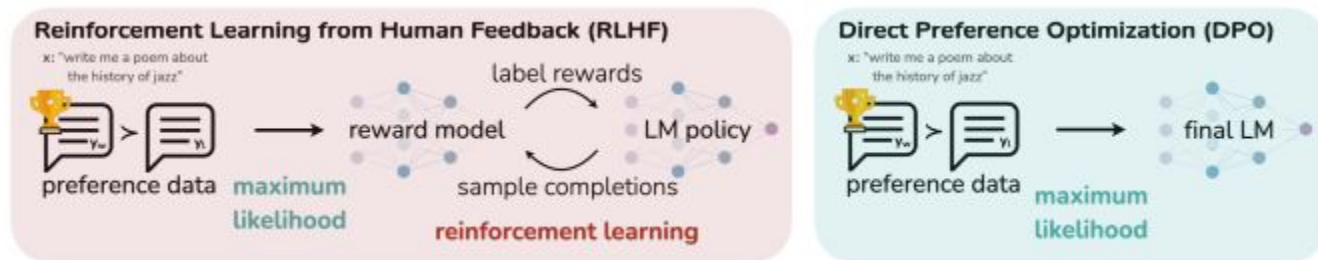
Generalization to a new input distribution

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

Validating GPT-4 judgments with human judgments

|                   | 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    |

# Conclusion



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

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

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

# Thank You

---

들어주셔서 감사합니다 :)