Variants of Direct Preference Optimization

발표자: 김민혁

Korea University

Natural Language Processing
& Artificial Intelligence

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
{rafailov,architsh,eric.mitchell}@cs.stanford.edu

NeurIPS 2023

Direct Preference Optimization: Your Language Model is Secretly a Reward Model

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

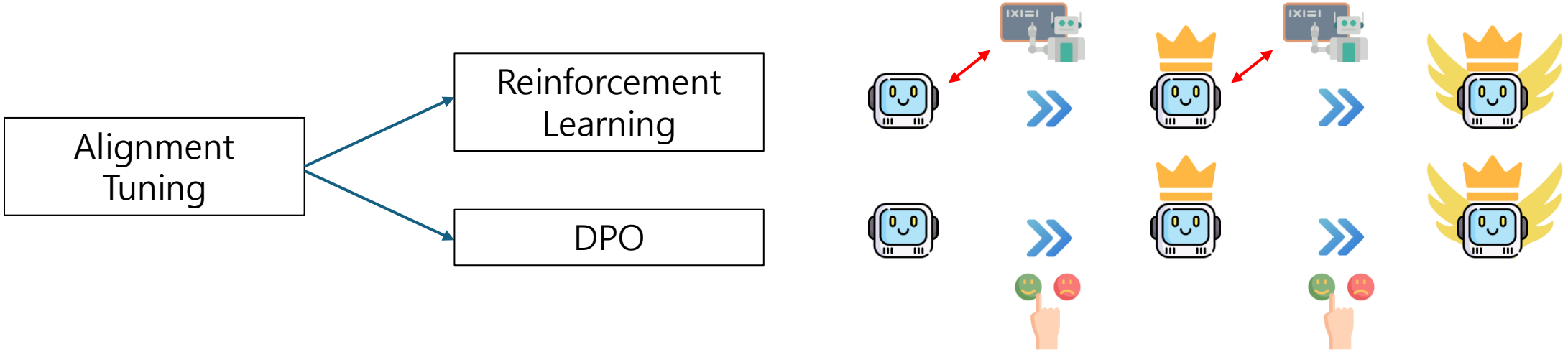
Chosen answer 확률 Rejected answer 확률

이 차이(Margin)을 벌리자!

Loss

- 인간 선호 쌍 데이터(query x, chosen answer y_w, rejected answer y_l)를 사용, chosen answer의 확률은 높이고 rejected answer의 확률은 낮춤
- 별도의 보상 모델 학습 X, 학습 정책 π_θ와 참조 정책 π_ref의 로그비를 암묵적인 보상으로 활용
- 분할 함수 Z(x)는 Bradley-Terry 식에 의해 소거 (Chosen answer 확률 – Rejected anwer 확률 각 항에서 동일한 Z(x)를 사용하기 때문)

DPO는 강화학습이 아닙니다.



using preferences directly. Unlike prior RLHF methods, which learn a reward and then optimize it via RL, our approach leverages a particular choice of reward model parameterization that enables extraction of its optimal policy in closed form, without an RL training loop. As we will describe

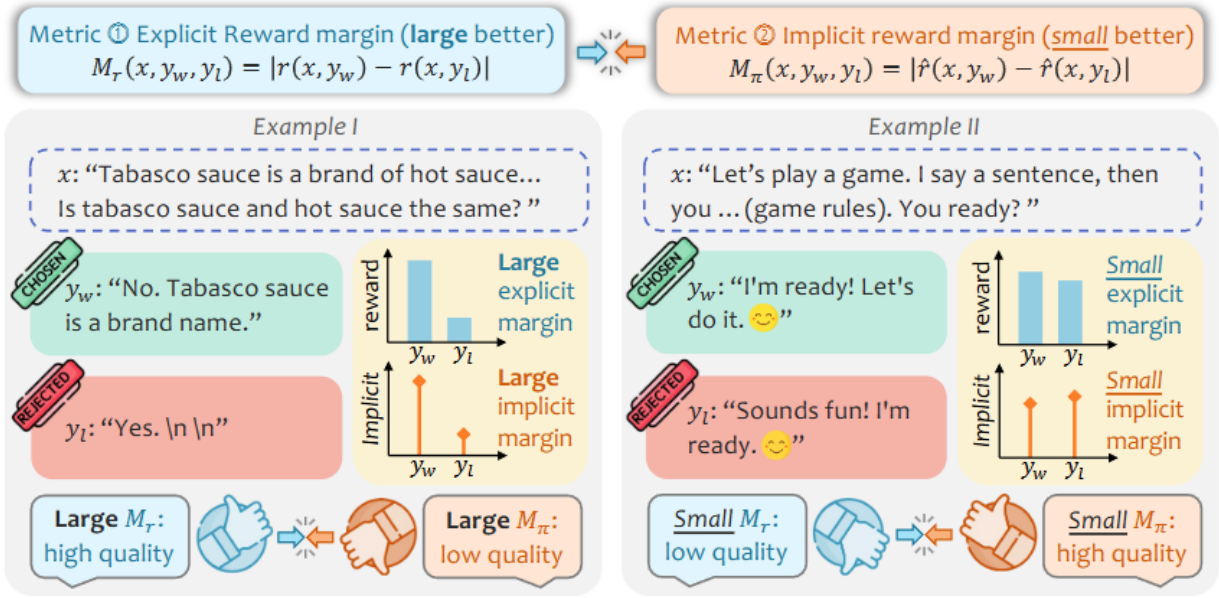
- 강화학습: 에이전트가 환경과 상호작용하며 보상을 최대화하는 정책 학습 (학습된 보상 모델을 환경으로 사용)
- DPO: 선호도 데이터를 직접 사용하여 모델의 정책을 직접 최적화 (별도의 보상 모델 사용 X)

Larger or Smaller Reward Margins to Select Preferences for Alignment?

Kexin Huang [†1] Junkang Wu [†1] Ziqian Chen [2] Xue Wang [2] Jinyang Gao [2] Bolin Ding [2] Jiancan Wu [1]
Xiangnan He [1] Xiang Wang [1]

ICML 2025

Larger or Smaller Reward Margins to Select Preferences for Alignment?



DPO에 적합한 데이터는 무엇인가?

- M_r: 외부 reward 모델을 사용하여 부여한 점수를 바탕으로 측정한 margin
- M_π: 학습 대상이 되는 모델의 logit을 바탕으로 측정한 margin
- 일반적으로 M_r이 높을 수록, M_π가 낮을 수록 DPO에 적합한 데이터

  → M_r이 높다는 건 Chosen과 Rejected의 구분선이 뚜렷하다는 뜻

  → M_ π가 낮다는 건 학습 대상이 되는 모델에게 어려운 데이터라는 뜻

Larger or Smaller Reward Margins to Select Preferences for Alignment?

$M_r \downarrow$ / $M_\pi \downarrow$	$M_r \uparrow$ / $M_\pi \downarrow$
외부 reward 모델이 구분하기 어렵고, 학습 대상이 되는 모델도 어려워함 **(Too hard)**	외부 reward 모델이 구분하기 쉽고, 학습 대상이 되는 모델은 어려워함 **(Best)**
$M_r \downarrow$ / $M_\pi \uparrow$	$M_r \uparrow$ / $M_\pi \uparrow$
외부 reward 모델은 구분이 어려우나, 학습 대상이 되는 모델은 이미 잘 함 **(Worst)**	외부 reward 모델이 구분하기 쉽고, 학습 대상이 되는 모델 역시 이미 잘 함 **(Use-less)**

M_r vs. M_π

- M_r: 외부 reward 모델을 사용하여 부여한 점수를 바탕으로 측정한 margin
- M_π: 학습 대상이 되는 모델의 logit을 바탕으로 측정한 margin

좋은 데이터는 $M_r \uparrow$ / $M_\pi \downarrow$

Metric Formulation - Preliminary

$$r^*(x, y_w) - r^*(x, y_l) \equiv \hat{r}^*_\theta(x, y_w) - \hat{r}^*_\theta(x, y_l).$$

Human Preference

↓

외부 Reward Model

Target Model
(학습 O)

학습이 완료된 모델의 Margin은 Human Preference와 일치해야 한다!

- 실제로는 Human Preference를 바탕으로 Margin을 구할 수 없기 때문에 외부 Reward Model로 갈음

Larger or Smaller Reward Margins to Select Preferences for Alignment?

Metric Formulation - M_1

$$
\begin{aligned}
&M_1(x, y_w, y_l; \theta, r^*) \\
:=&|(\hat{r}^*_\theta(x, y_w) - \hat{r}^*_\theta(x, y_l)) - (\hat{r}_\theta(x, y_w) - \hat{r}_\theta(x, y_l))| \\
=&|\underbrace{(r^*(x, y_w) - r^*(x, y_l))}_{\text{given by aligned optimum } \pi^*_\theta} - \underbrace{(\hat{r}_\theta(x, y_w) - \hat{r}_\theta(x, y_l))}_{\text{given by current model } \pi_\theta}|
\end{aligned}
$$

외부 Reward Model

Target Model
(학습 X)

좋은 데이터가 M_r ↑ / M_π ↓일 때 데이터의 품질을 파악하는 M_1

- M_r과 M_π의 차가 크면 M_1 ↑

Larger or Smaller Reward Margins to Select Preferences for Alignment?

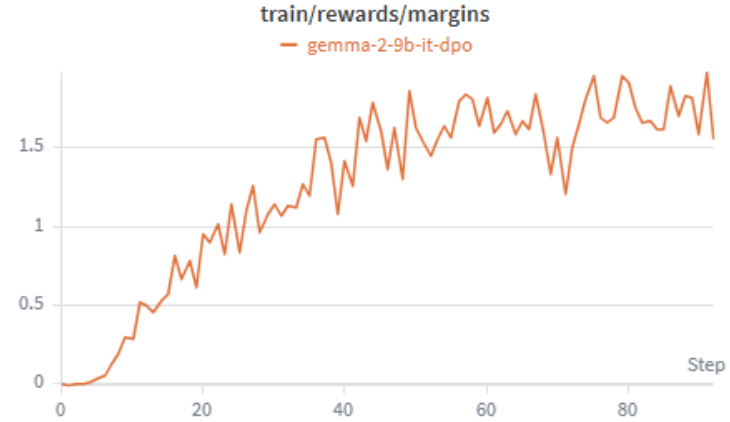
Metric Formulation - M+

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

$$\begin{aligned} &M_1(x, y_w, y_l; \theta, r^*) \\ :=&|(\hat{r}^*_\theta(x, y_w) - \hat{r}^*_\theta(x, y_l)) - (\hat{r}_\theta(x, y_w) - \hat{r}_\theta(x, y_l))| \\ =&|\underbrace{(r^*(x, y_w) - r^*(x, y_l))}_{\text{given by aligned optimum } \pi^*_\theta} - \underbrace{(\hat{r}_\theta(x, y_w) - \hat{r}_\theta(x, y_l))}_{\text{given by current model } \pi_\theta}| \end{aligned}$$

↓

$$\begin{aligned} &M_+(x, y_w, y_l; \theta, r^*) \qquad (10) \\ :=&(r^*(x, y_w) - r^*(x, y_l)) - (\hat{r}_\theta(x, y_w) - \hat{r}_\theta(x, y_l)). \end{aligned}$$



DPO 학습 경과에 따른 실제 Margin 변화

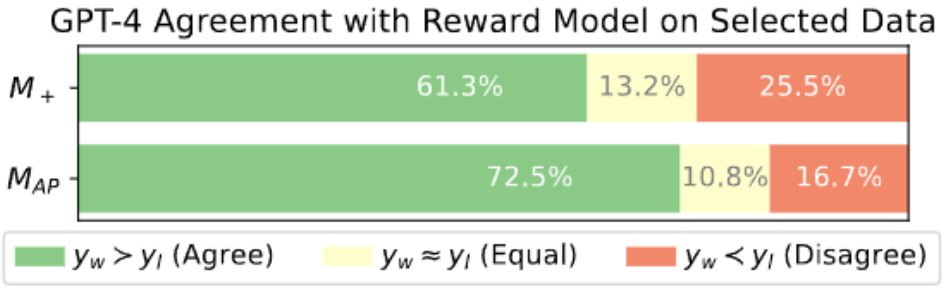
(Margin을 증가시키는 단방향 – Chosen > Rejected)

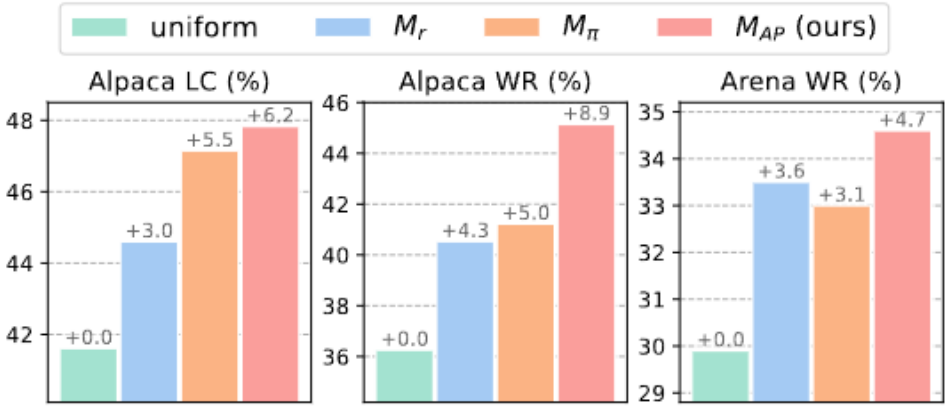
M_1을 보정한 M+ (절대값 제거)

- M_r과 M_π의 차가 크면 M_1 ↑
- 하지만 M_1은 M_r과 M_ π 차에 절대값을 씌움
- 이상적인 보상 마진이 양수이든 음수이든 상관없이, 현재 LLM의 보상 마진과의 차이가 얼마나 큰지만을 측정
- 하지만 실제 LLM 정렬 목표는 이상적인 보상 마진의 방향을 따라가는 것
  - 방향을 무시하고 거리만 보기 때문에 학습이 잘못된 방향으로 갈 수 있음 (Chosen < Rejected 인 경우에도 M_1은 높아질 수 있음)

Metric Formulation - MAP

$$M_{AP}(x, y_w, y_l) = \underbrace{|r(x, y_w) - r(x, y_l)|}_{\text{target explicit reward margin}} - \underbrace{|\hat{r}_\theta(x, y_w) - \hat{r}_\theta(x, y_l)|}_{\text{current implicit reward margin}}.$$





보상 모델의 노이즈를 잡는 MAP

- M+는 보상 모델이 판단하는 선호도와 현재 LLM이 판단하는 선호도 간의 차이를 측정
- 보상 모델이 잘못된 판단을 하여 M_r이 높아져버리면 M+ 수식으로는 이를 학습 잠재력이 높은 고품질 데이터로 판단
- **절대값 다시 도입. 단, M_r과 M+ 각각에 절대값을 취함** (기존: M_r – M+에 절대값을 취함)

Larger or Smaller Reward Margins to Select Preferences for Alignment?

EVA Framework (Original)

Select

외부 reward 모델을 사용한 margin 부여 후 상위 k개 선택

→

Evolve

Query를 재생성

Framework via MAP

Evolve

Query를 재생성

→

Select

외부 reward 모델을 사용한 margin 부여 후 상위 k개 선택

High Quality Data Generation

- 기존 self-play 방식의 EVA Framework 를 MAP 수식을 통해 개선
- EVA Framework 에서의 좋은 데이터 선별 후 재생성의 과정이 높은 퀄리티의 데이터를 보장하지 못하기 때문에 재생성 후 선별의 과정으로 변경

Larger or Smaller Reward Margins to Select Preferences for Alignment?

Metric Formulation - MAP

Method	Llama-3-8b-instruct			Gemma-2-9b-it		
	AH	AE 2.0		AH	AE 2.0	
	WR	LC	WR	WR	LC	WR
SFT	21.4	23.25	23.50	40.7	48.75	36.49
$DPO_{uf\text{-}10k}$	23.6	29.80	28.32	46.0	54.00	46.11
+uf-10k	35.0	42.67	41.67	56.4	62.56	62.17
+eva-10k	31.4	39.97	40.74	56.0	63.32	62.10
+ours-10k	**35.4**	**42.82**	**45.00**	**56.6**	**63.88**	**64.52**
$SimPO_{uf\text{-}10k}$	24.2	28.45	26.23	45.2	56.90	43.80
+uf-10k	28.5	35.75	32.82	53.5	66.43	64.55
+eva-10k	32.6	41.12	39.94	**61.0**	66.78	66.86
+ours-10k	**34.6**	**43.76**	**42.00**	60.9	**66.85**	**68.34**

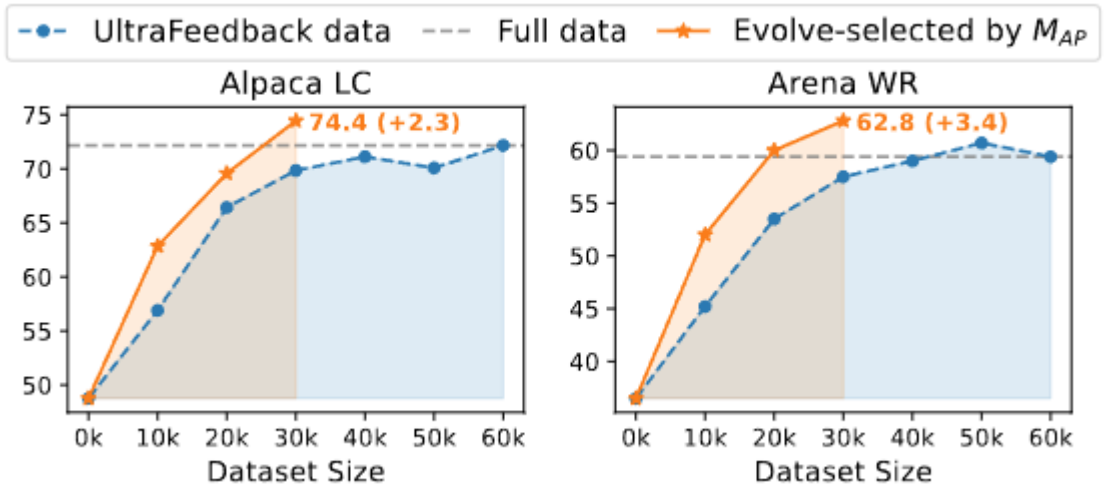
Result

- Uf-10K: Ultrafeedback 데이터셋 10K 샘플링
- Eva-10K, Ours-10K 모두 select-then-evolve 방식으로 수행 (Appendix에 evolve-then-select 방식으로 진행한 결과 값이 있음: 순서를 바꾼 것은 일관된 개선을 가져오지는 못함)

Larger or Smaller Reward Margins to Select Preferences for Alignment?

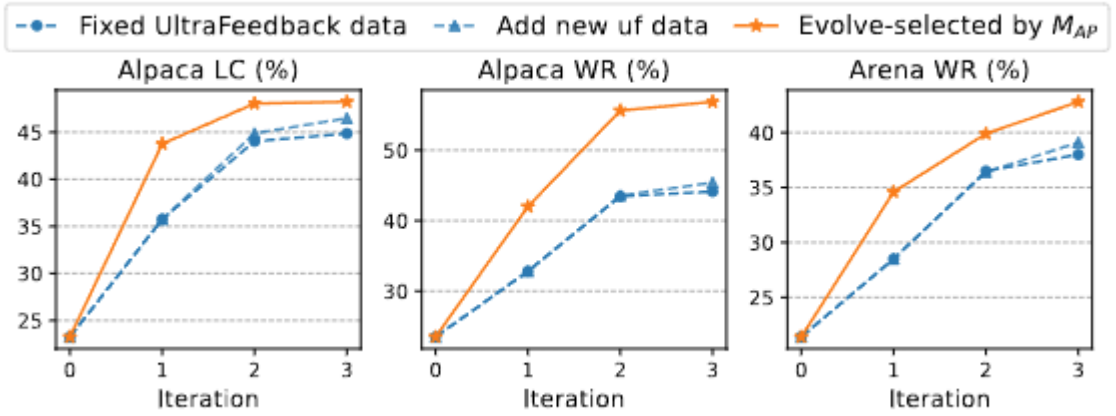
Scaling with Dataset Size

- 같은 사이즈의 데이터셋을 사용하였음에도 MAP 기반 데이터셋 구축이 더 높은 성능 향상을 가져옴
- 2배 더 많은 양의 데이터셋을 학습시킨 것 보다 더 높은 성능을 기록함



Scaling with Training Iteration

- MAP 기반으로 구축된 데이터는 Iteration 반복 시에도 기존 대비 꾸준히 더 높은 성능을 기록함
- 데이터를 추가로 샘플링하여 다음 iteration 때 적용하여 추가로 학습하여도 MAP 기반 데이터셋이 더 높은 성능 향상 폭을 불러옴

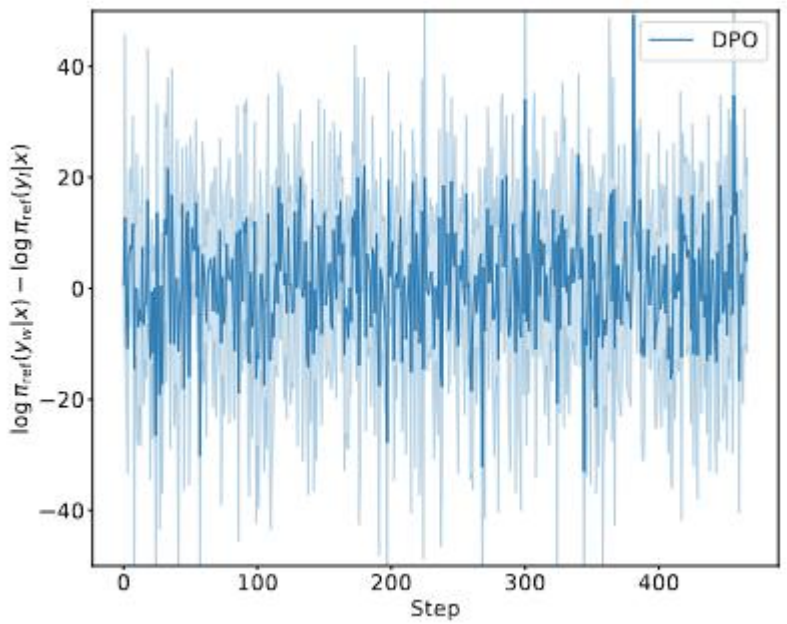


AlphaDPO: Adaptive Reward Margin for Direct Preference Optimization

Junkang Wu [1] **Xue Wang** [2] **Zhengyi Yang** [1] **Jiancan Wu** [1] **Jinyang Gao** [2] **Bolin Ding** [2] **Xiang Wang** [1 *]
Xiangnan He [1 *]

ICML 2025

AlphaDPO: Adaptive Reward Margin for Direct Preference Optimization



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

Limitations of DPO & SimPO

- DPO는 학습 진행에 따라 성능이 저하될 수 있는 정적인 참조 모델(초기 모델)에 의존
- SimPO는 모든 데이터 샘플에 대해 동일, 균일한 목표 보상 margin을 가정하여 각 쌍의 선호도 강도를 무시함

AlphaDPO: Adaptive Reward Margin for Direct Preference Optimization

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

SimPO: Alpha = 0
DPO: Alpha = 1

① 균등 분포

$$\hat{\pi}_{\text{ref}}(y|x) \propto U(y|x)\left(\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right)^{\alpha}$$

$$\log\frac{\hat{\pi}_{\text{ref}}(y_w|x)}{\hat{\pi}_{\text{ref}}(y_l|x)} = \log\left(\frac{U(y_w|x)(\pi_\theta(y_w|x)/\pi_{\text{ref}}(y_w|x))^\alpha}{U(y_l|x)(\pi_\theta(y_l|x)/\pi_{\text{ref}}(y_l|x))^\alpha}\right)$$
$$= \log\frac{U(y_w|x)}{U(y_l|x)} + \alpha\left(\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)$$

전개 과정

AlphaDPO

② $$L_{\text{AlphaDPO}}(\pi_\theta, \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)} - \beta\log\frac{\hat{\pi}_{\text{ref}}(y_w|x)}{\hat{\pi}_{\text{ref}}(y_l|x)}\right)\right]$$

- 참조 모델은 선호 응답과 덜 선호되는 응답을 구별하는 데 기여해야 함
- 각 데이터 샘플별 난이도 및 모델의 현재 실력에 따라 다른 Margin을 적용해야 함

→ 각 응답 쌍의 특성에 맞게 보상 마진을 동적으로 조정하는 implicit한 참조 모델을 도입 ①

→ 이를 기존 DPO 식에 활용하여 전개하면 ②

AlphaDPO: Adaptive Reward Margin for Direct Preference Optimization

②

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

$$M(x,y_w,y_l) = \beta\left[\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right]$$ 이라 하면

$$= -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}$$

$$\left[\log\sigma\left(\beta\left(\log\frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)}\right) - [\gamma + \alpha M(x,y_w,y_l)]\right)\right]$$ ③

AlphaDPO

- 앞서 구한 2번 식을 각 데이터 샘플별 Margain M으로 하여 전개하면 ③
- γ: SimPO에서 사용된 고정된 보상 마진과 유사한 역할을 하는 상수항

AlphaDPO: Adaptive Reward Margin for Direct Preference Optimization

$$= -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log\sigma\left(\beta\left(\log\frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)}\right) - [\gamma + \alpha M(x,y_w,y_l)]\right)\right] \quad ③ \longrightarrow$$

④

$$\mathcal{L}_{\text{AlphaDPO}}(\pi_\theta,\pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} [\log\sigma\left(u(x,y_w,y_l) - \text{sg}\left[\gamma + \alpha M^*(x,y_w,y_l)\right]\right)]$$

AlphaDPO

- Sg (Stop Gradient): π_ref는 학습 과정에서 고정해야 하므로 Stop-gradient를 적용하여 역전파 방지 (학습 안정성 보장)
- Normalization of M (x, yw, yl): 정책 모델과 참조 모델 간의 차이를 측정하는 항으로, 학습 데이터셋 내에서 값의 범위(scale)가 다양할 수 있고, 이러한 값의 변동이 손실 함수를 지배하여 학습을 불안정하게 만드는 것을 방지하기 위해 Z-score 정규화를 사용
- Length-normalized Reward Formulation: 보상 값을 응답 시퀀스의 길이에 맞춰 조정

AlphaDPO: Adaptive Reward Margin for Direct Preference Optimization

Method	Llama3-Instruct (8B)					Mistral-Instruct (7B)				
	AlpacaEval 2		Arena-Hard			AlpacaEval 2		Arena-Hard		
	LC (%)	WR (%)	SC (%)	LC (%)	WR (%)	LC (%)	WR (%)	SC (%)	LC (%)	WR (%)
SFT	24.0	23.6	22.1	22.2	22.4	19.0	15.4	18.3	13.2	12.9
DPO	40.2	**38.1**	31.9	32.0	31.2	20.3	17.9	18.9	13.7	13.4
IPO	35.9	34.4	29.2	29.9	30.2	22.3	18.6	22.4	16.6	16.2
CPO	29.6	34.4	26.3	28.1	29.4	26.2	31.7	26.6	21.4	**23.8**
KTO	38.3	34.1	30.3	30.6	30.3	19.4	20.3	21.5	16.0	16.8
ORPO	31.6	29.8	26.6	26.6	26.3	24.0	23.0	24.4	18.5	18.6
R-DPO	40.3	37.3	33.1	32.9	32.9	21.4	22.2	18.7	14.0	13.8
SimPO	43.8	38.0	33.5	33.5	32.6	30.2	32.1	25.6	19.8	20.1
AlphaDPO	**46.6**	**38.1**	**34.1**	**34.2**	**33.3**	**32.3**	**32.6**	**27.2**	**21.5**	21.5
Method	Llama3-Instruct v0.2 (8B)					Gemma2-Instruct (9B)				
	AlpacaEval 2		Arena-Hard			AlpacaEval 2		Arena-Hard		
	LC (%)	WR (%)	SC (%)	LC (%)	WR (%)	LC (%)	WR (%)	SC (%)	LC (%)	WR (%)
SFT	24.0	23.6	22.1	22.2	22.4	48.7	36.5	32.0	42.2	42.1
DPO	51.9	50.8	26.1	31.5	33.9	70.4	**66.9**	43.9	55.6	58.8
IPO	40.6	39.6	**31.1**	34.2	34.9	62.6	58.4	41.1	51.9	53.5
CPO	36.5	40.8	29.4	32.8	34.2	56.4	53.4	42.4	53.3	55.2
KTO	41.4	36.4	27.1	29.5	28.9	61.7	55.5	41.7	52.3	53.8
ORPO	36.5	33.1	28.8	30.8	30.4	56.2	46.7	35.1	45.3	46.2
R-DPO	51.6	50.7	29.2	34.3	35.0	68.3	**66.9**	45.1	55.9	57.9
SimPO	55.6	49.6	28.5	34.0	33.6	72.4	65.0	45.0	56.1	57.8
AlphaDPO	**58.7**	**51.1**	30.8	**36.3**	**35.7**	**73.4**	66.1	**48.6**	**59.3**	**60.8**

Main Result

- 대부분의 경우 AlphaDPO는 다른 방법론보다 우수한 성능을 보임
- 길이 제어(LC) 지표는 모델의 실제 성능을 더 잘 반영하는 것으로 나타남. 응답 길이의 편향을 줄여 모델의 품질을 더 정확하게 평가할 수 있기 때문

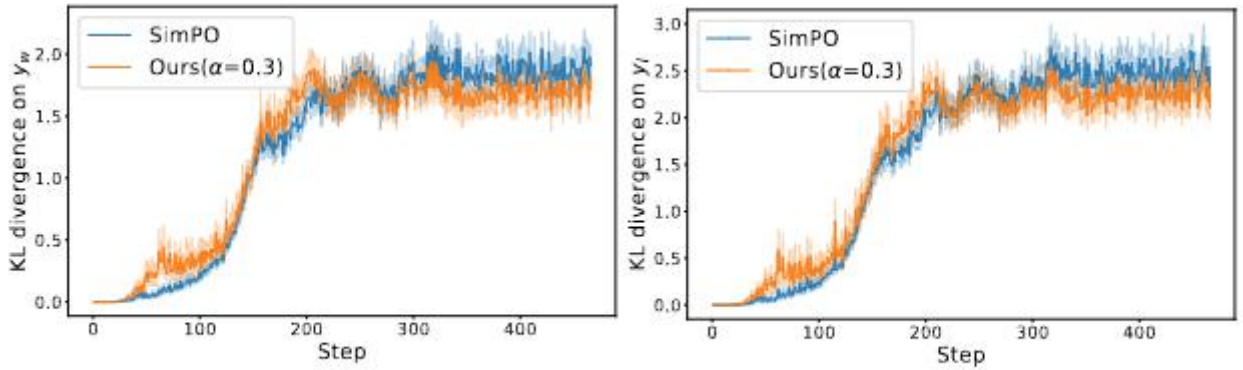
AlphaDPO: Adaptive Reward Margin for Direct Preference Optimization

Method	Llama3-Instruct v0.2 (8B)					Mistral-Instruct (7B)				
	AlpacaEval 2		Arena-Hard			AlpacaEval 2		Arena-Hard		
	LC (%)	WR (%)	SC (%)	LC (%)	WR (%)	LC (%)	WR (%)	SC (%)	LC (%)	WR (%)
$U(\cdot\|x)$	55.6	49.6	28.5	34.0	33.6	30.2	32.1	25.6	19.8	20.1
$U(\cdot\|x)\,(\pi_\theta(\cdot\|x)/\pi_{\text{ref}}(\cdot\|x))^\alpha$	58.7	51.1	30.8	36.3	35.7	32.3	32.6	27.2	21.5	21.5
w/o Normalization	56.5	49.7	23.1	28.4	27.7	32.1	33.1	25.2	19.7	19.6
w/o sg	2.7	3.7	7.7	5.4	6.3	27.2	27.7	25.8	20.3	20.7
$\gamma = 0$	51.2	44.9	30.0	34.5	33.3	31.9	31.3	24.2	19.6	19.3

Ablation Study

- U(·|x): α = 0인 케이스로, SimPO에 해당
- (U(·|x) (πθ(·|x)/πref(·|x))^α): AlphaDPO
- U(·|x): α = 0인 케이스로, SimPO에 해당

AlphaDPO: Adaptive Reward Margin for Direct Preference Optimization

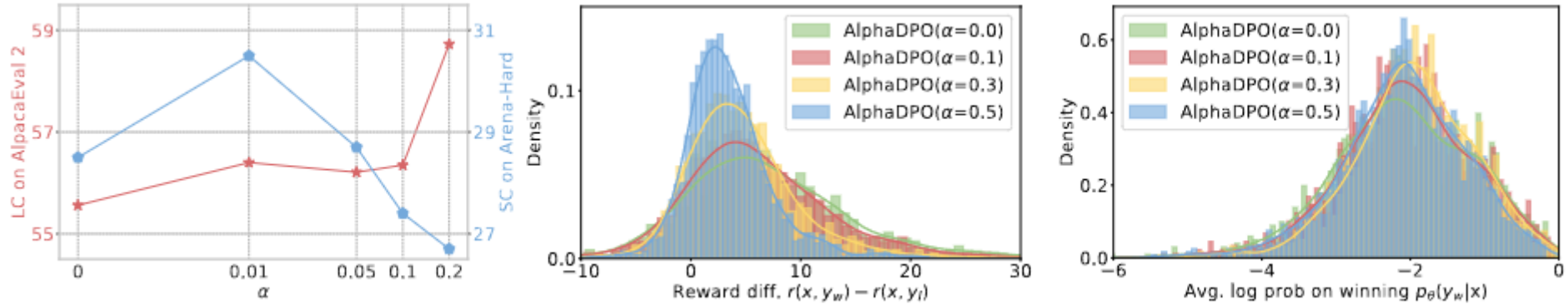


(a) KL divergence of y_w over steps. (b) KL divergence of y_l over steps.

KL divergence

- (a) 선호 응답에 대해 AlphaDPO가 SimPO보다 학습 초기 단계에서 약간 더 높은 KL 발산을 보이지만, 이후에는 비슷한 수준으로 수렴
- (b) 선호하지 않는 응답에 대해 AlphaDPO는 학습 초기에는 y_lyly_lyl에 대한 KL 발산이 매우 낮지만, 점차 증가하여 SimPO와 비슷한 수준으로 수렴함. 거절 응답에 대한 KL 발산이 선호 응답 보다 더 높은 수준에서 수렴

AlphaDPO: Adaptive Reward Margin for Direct Preference Optimization



AlphaDPO

- 왼쪽 그래프: α 값에 따른 AlpacaEval의 길이 제어(LC) 정확도와 Arena-Hard의 스타일 제어(SC) 정확도의 변화
- 중앙 그래프: 다른 α 값에 대한 보상 차이(Margin)의 분포
- 오른쪽 그래프: 다른 α 값에 대한 선호 응답의 평균 로그 확률분포

Thank you

Q&A