Stat 9911
Various forms of preference optimization

Tao Wang and Sunay Joshi

July 3, 2026

Table of Contents

Recap of RLHF and DPO

RRHF

CPO

KTO

IPO

ORPO

SimPO

Experiments (from SimPO)

RLHF

- Standard workflow of reinforcement learning from human feedback
  - Supervised fine-tuning (SFT)
  - Reward model (RM) training
  - Proximal policy optimization (PPO)

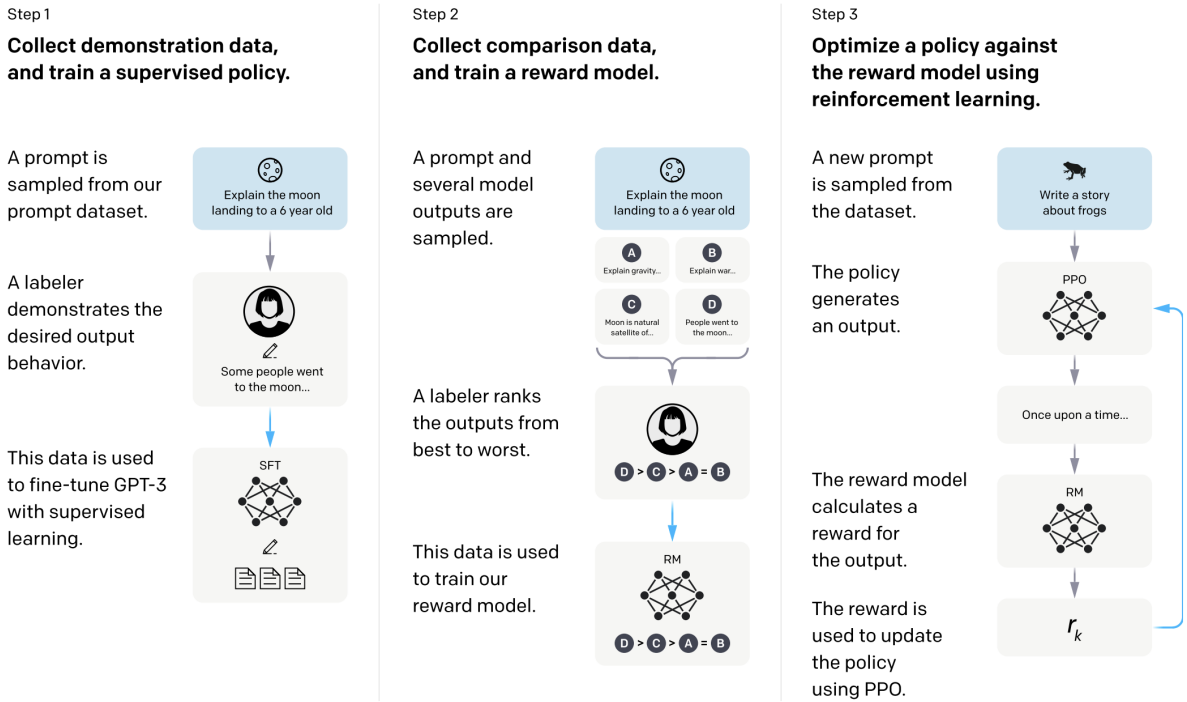


Figure: RLHF (Ouyang et al., 2022)

DPO

- RLHF objective:

$$\max_{\pi_\theta} \mathbb{E}_{X\sim\mathcal{D},Y\sim\pi_\theta}[r(X,Y)] - \beta \mathrm{KL}\left[\pi_\theta(Y \mid X) \middle|\middle| \pi_{\mathsf{ref}}(Y \mid X)\right]$$

DPO

- RLHF objective:

$$\max_{\pi_\theta} \mathbb{E}_{X\sim\mathcal{D}, Y\sim\pi_\theta}[r(X,Y)] - \beta \mathrm{KL}\left[\pi_\theta(Y \mid X) \middle|\middle| \pi_{\mathsf{ref}}(Y \mid X)\right]$$

- Optimality:

$$\pi^*(y \mid x) = \frac{1}{Z(x)} \pi_{\mathsf{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x,y)\right)$$

DPO

- RLHF objective:

$$\max_{\pi_\theta} \mathbb{E}_{X\sim\mathcal{D}, Y\sim\pi_\theta}[r(X,Y)] - \beta \mathrm{KL}\left[\pi_\theta(Y \mid X) \middle|\middle| \pi_{\text{ref}}(Y \mid X)\right]$$

- Optimality:

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

- Loss function for reward modeling:

$$\mathcal{L}_R(r, \mathcal{D}) = -\mathbb{E}_{(X,Y_w,Y_l)\sim\mathcal{D}}\left[\log \sigma\left(r\left(X, Y_w\right) - r\left(X, Y_l\right)\right)\right]$$

DPO

- RLHF objective:

$$\max_{\pi_\theta} \mathbb{E}_{X\sim\mathcal{D}, Y\sim\pi_\theta}[r(X,Y)] - \beta \mathrm{KL}\left[\pi_\theta(Y \mid X) \middle|\middle| \pi_{\text{ref}}(Y \mid X)\right]$$

- Optimality:

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

- Loss function for reward modeling:

$$\mathcal{L}_R(r, \mathcal{D}) = -\mathbb{E}_{(X,Y_w,Y_l)\sim\mathcal{D}}\left[\log \sigma\left(r\left(X, Y_w\right) - r\left(X, Y_l\right)\right)\right]$$

- Direct Preference Optimization(DPO) Loss:

$$-\mathbb{E}_{(X,Y_w,Y_l)\sim\mathcal{D}}\left[\log \sigma\left(\beta \log \frac{\pi_\theta\left(Y_w \mid X\right)}{\pi_{\text{ref}}\left(Y_w \mid X\right)} - \beta \log \frac{\pi_\theta\left(Y_l \mid X\right)}{\pi_{\text{ref}}\left(Y_l \mid X\right)}\right)\right]$$

Is the DPO optimum well-posed?

- The conditional distributions $\pi^*(y|x)$ obtained through DPO **might not** correspond to a valid probability distribution $\pi^*(z)$ over all strings z. But this is not an issue if we parametrize r appropriately:

Theorem (Edgar)

DPO yields a valid probability distribution π^* if and only if there is a probability distribution q, and a constant c, such that for all strings z,

$$r(z)/\beta = \log(q(z)/\pi_{\text{ref}}(z)) + c.$$

Alternatives

- RRHF: Rank Responses with Human Feedback (Yuan et al., 2023)
- CPO: Contrastive Preference Optimization (Xu et al., 2024)
- KTO: Kahneman-Tversky Optimization (Ethayarajh et al., 2024)
- ORPO: Odds Ratio Preference Optimization (Hong et al., 2024)
- IPO: Identity Preference Optimization (Azar et al., 2024)
- SimPO: Simple Preference Optimization (Meng et al., 2024)

Table of Contents

Recap of RLHF and DPO

RRHF

CPO

KTO

IPO

ORPO

SimPO

Experiments (from SimPO)

RRHF: Rank Responses to Align Language Models with Human Feedback without tears

Zheng Yuan[1]* **Hongyi Yuan**[12]* **Chuanqi Tan**[1] **Wei Wang**[1] **Songfang Huang**[1] **Fei Huang**[1]
[1]Alibaba DAMO Academy [2]Tsinghua University
{yuanzheng.yuanzhen,chuanqi.tcq}@alibaba-inc.com
yuanhy20@mails.tsinghua.edu.cn

Motivation: Drawbacks of PPO

- PPO requires complex hyperparameter tuning, reward design and advantage estimation.

Motivation: Drawbacks of PPO

- PPO requires complex hyperparameter tuning, reward design and advantage estimation.
- PPO needs to store multiple models:
  - policy model
  - value model/head
  - reward model
  - reference model (e.g., SFT)

Motivation: Drawbacks of PPO

- PPO requires complex hyperparameter tuning, reward design and advantage estimation.
- PPO needs to store multiple models:
  - policy model
  - value model/head
  - reward model
  - reference model (e.g., SFT)
- Goal: simplify the training pipeline.

RRHF

(1) Sampling responses y_i for x from policy ρ_i.

- ρ_i can be the initial model, the learned model π, other LLMs like GPT-4, or human expert responses.

RRHF

(1) Sampling responses y_i for x from policy ρ_i.

- ρ_i can be the initial model, the learned model π, other LLMs like GPT-4, or human expert responses.

(2) Scoring responses by a reward function and by the model π using length normalized log conditional probabilities:

$$r_i = R(x, y_i), \quad p_i = \frac{1}{\|y_i\|} \sum_t \log P_\pi \left(y_{i,t} \mid x, y_{i,<t} \right)$$

RRHF

(1) Sampling responses y_i for x from policy ρ_i.

- ρ_i can be the initial model, the learned model π, other LLMs like GPT-4, or human expert responses.

(2) Scoring responses by a reward function and by the model π using length normalized log conditional probabilities:

$$r_i = R(x, y_i), \quad p_i = \frac{1}{\|y_i\|} \sum_t \log P_\pi \left(y_{i,t} \mid x, y_{i,<t}\right)$$

(3) Ranking loss:

$$L_{\text{rank}} = \sum_{r_i < r_j} \max\left(0, p_i - p_j\right)$$

RRHF

(1) Sampling responses y_i for x from policy ρ_i.

- ρ_i can be the initial model, the learned model π, other LLMs like GPT-4, or human expert responses.

(2) Scoring responses by a reward function and by the model π using length normalized log conditional probabilities:

$$r_i = R(x, y_i), \quad p_i = \frac{1}{\|y_i\|} \sum_t \log P_\pi \left(y_{i,t} \mid x, y_{i,<t}\right)$$

(3) Ranking loss:

$$L_{\text{rank}} = \sum_{r_i < r_j} \max\left(0, p_i - p_j\right)$$

(4) Cross-entropy loss:

$$i' = \arg\max_i r_i, \quad L_{ft} = -\sum_t \log P_\pi \left(y_{i',t} \mid x, y_{i',<t}\right)$$

RRHF

(1) Sampling responses y_i for x from policy ρ_i.

- ρ_i can be the initial model, the learned model π, other LLMs like GPT-4, or human expert responses.

(2) Scoring responses by a reward function and by the model π using length normalized log conditional probabilities:

$$r_i = R(x, y_i), \quad p_i = \frac{1}{\|y_i\|} \sum_t \log P_\pi \left(y_{i,t} \mid x, y_{i,<t}\right)$$

(3) Ranking loss:

$$L_{\text{rank}} = \sum_{r_i < r_j} \max\left(0, p_i - p_j\right)$$

(4) Cross-entropy loss:

$$i' = \arg\max_i r_i, \quad L_{ft} = -\sum_t \log P_\pi \left(y_{i',t} \mid x, y_{i',<t}\right)$$

(5) Total Loss:

$$L = L_{rank} + L_{ft}$$

RRHF Workflow

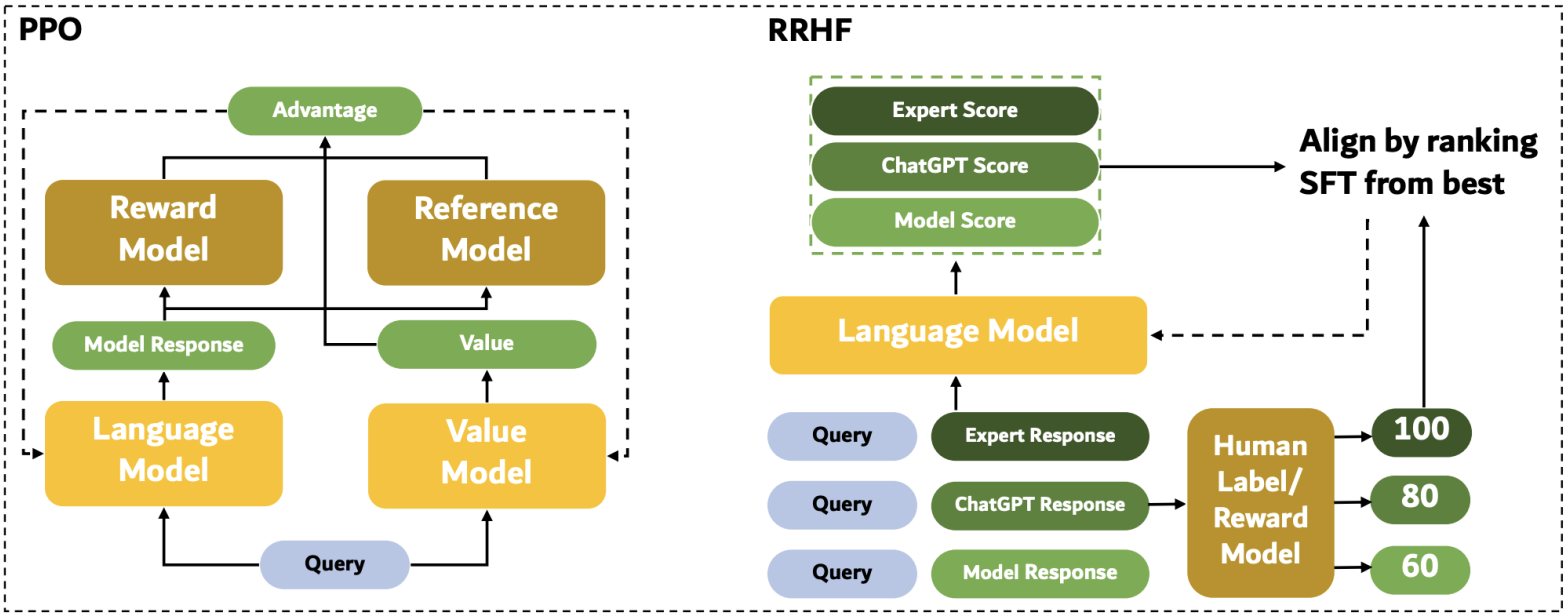


Figure: Workflow of RRHF compared with PPO

RRHF Experiments Setup

- Dataset: Anthropic's Helpfulness and Harmlessness dataset
- Proxy Reward Model: Dahoas/gptj-rm-static
- Models: Llama-7B, Alpaca-7B, Alpaca-7B SFT

RRHF Experiments Setup

- Dataset: Anthropic's Helpfulness and Harmlessness dataset
- Proxy Reward Model: Dahoas/gptj-rm-static
- Models: Llama-7B, Alpaca-7B, Alpaca-7B SFT
- Sampling Policy:

Table 1: Sampling policy used in our experiments. OP-k uses π for sampling (i.e. online sampling), we update π every k optimization steps. IP-n (Iterate update) uses updated policy ρ^* after training by IP-(n-1) and starts a new iteration. The dataset contains a good response and a bad response for each query which are used as ρ_5 and ρ_6, which are termed **P** (**P**rovided responses in datasets).

Setting	$\rho_1 \sim \rho_4$	ρ_5, ρ_6
BP	**B**eam search by ρ	**P**rovided responses
SP	top-p **S**ampling by ρ	**P**rovided responses
DP	**D**iverse beam search by ρ	**P**rovided responses
OP-k	**O**nline diverse beam by $\pi\dagger$	**P**rovided responses
IP-n	**I**terate diverse beam by ρ^*	**P**rovided responses
D	**D**iverse beam search by ρ	$\emptyset$
P	$\emptyset$	**P**rovided responses

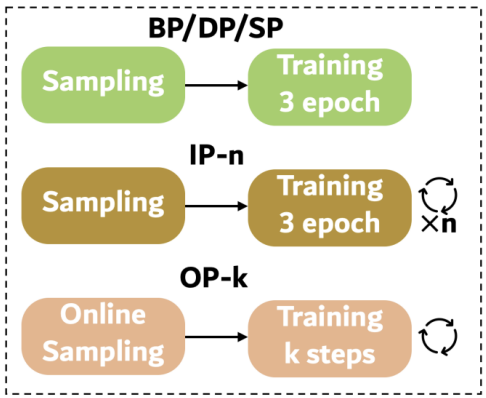


Figure 2: The workflow of sampling policy used in our experiments. IP-1 is equivalent to DP.

Auto Evaluation

- perplexity (gpt2-medium)
- average reward score (Dahoas/gptj-rm-static)

ρ	Setting	PPL	Reward
Good responses	$\emptyset$	21.46	-1.24
Bad responses	$\emptyset$	121.29	-1.48
LLaMA	$\emptyset$	20.78	-1.89
Alpaca	$\emptyset$	14.34	-1.18
Alpaca-sft	$\emptyset$	18.98	-1.46
Alpaca	Best-of-4	-	-0.97
LLaMA	PPO	42.53	-1.62
Alpaca	PPO	13.84	-1.03
Alpaca-sft	PPO	19.10	-1.25
LLaMA	RRHF_{DP}	67.12	-1.34
Alpaca-sft	RRHF_{DP}	18.10	-1.19
Alpaca	RRHF_{DP}	14.75	-1.03
Alpaca	RRHF_{SP}	14.41	**-0.96**

Figure: Automatic evaluation on HH dataset

More Results

- Human Evaluation:

Table 3: Human evaluation on HH dataset. All settings use ρ=Alpaca.

A	B	win	tie	lose
RRHF_{DP}	Good responses	59	30	11
RRHF_{DP}	PPO	27	48	25
RRHF_{DP}	$\text{RRHF}_{\text{IP-2}}$	0	90	10

More Results

- Human Evaluation:

Table 3: Human evaluation on HH dataset. All settings use ρ=Alpaca.

A	B	win	tie	lose
RRHF_{DP}	Good responses	59	30	11
RRHF_{DP}	PPO	27	48	25
RRHF_{DP}	$\text{RRHF}_{\text{IP-2}}$	0	90	10

- Accuracy as a Reward Model:

Table 5: Reward model accuracy evaluation.

Reward Model	Accuracy
`Dahoas/gptj-rm-static`	68.49%
LLaMA	45.09%
Alpaca	45.13%
Alpaca-PPO	46.03%
Alpaca-RRHF_{DP}	61.75%

Ablation

ρ	Setting	PPL	Reward	Mean	Std.	Max
LLaMA	DP	67.12	-1.34	-2.18	0.97	-1.27
Alpaca	DP	14.75	**-1.02**	-1.30	0.66	-0.95
Alpaca-sft	DP	18.10	-1.19	-1.49	0.79	-1.11
LLaMA	BP	17.03	-1.27	-2.26	0.96	-1.26
Alpaca	BP	14.37	-1.03	-1.31	0.67	-1.00
Alpaca-sft	BP	17.63	-1.14	-1.50	0.77	-1.15
LLaMA	P	18.49	-1.31	-1.50	0.79	-1.28
Alpaca	P	18.88	-1.31	-1.50	0.79	-1.28
Alpaca-sft	P	18.92	-1.31	-1.50	0.79	-1.28
Alpaca	D	13.66	-1.08	-1.21	0.65	-1.02
Alpaca	IP-1	14.75	-1.02	-1.30	0.66	-0.95
Alpaca	IP-2	14.31	-0.96	-1.13	0.57	-0.77
Alpaca	IP-3	14.51	-0.94	-1.05	0.56	-0.65
Alpaca	OP-32	63.78	0.34	-	-	-
Alpaca	OP-32+KL	19.76	**-0.86**	-	-	-

Ablation

ρ	Setting	PPL	Reward	Mean	Std.	Max
LLaMA	DP	67.12	-1.34	-2.18	0.97	-1.27
Alpaca	DP	14.75	**-1.02**	-1.30	0.66	-0.95
Alpaca-sft	DP	18.10	-1.19	-1.49	0.79	-1.11
LLaMA	BP	17.03	-1.27	-2.26	0.96	-1.26
Alpaca	BP	14.37	-1.03	-1.31	0.67	-1.00
Alpaca-sft	BP	17.63	-1.14	-1.50	0.77	-1.15
LLaMA	P	18.49	-1.31	-1.50	0.79	-1.28
Alpaca	P	18.88	-1.31	-1.50	0.79	-1.28
Alpaca-sft	P	18.92	-1.31	-1.50	0.79	-1.28
Alpaca	D	13.66	-1.08	-1.21	0.65	-1.02
Alpaca	IP-1	14.75	-1.02	-1.30	0.66	-0.95
Alpaca	IP-2	14.31	-0.96	-1.13	0.57	-0.77
Alpaca	IP-3	14.51	-0.94	-1.05	0.56	-0.65
Alpaca	OP-32	63.78	0.34	-	-	-
Alpaca	OP-32+KL	19.76	**-0.86**	-	-	-

Table 7: Ranking loss ablation.

ρ	Setting	PPL	Reward
Alpaca	BP	14.37	-1.03
Alpaca	BP - L_{rank}	14.74	-1.14

Conclusion

- Advantage:
  - Simplified training pipeline
  - Stability

Conclusion

- Advantage:
  - Simplified training pipeline
  - Stability
- Disadvantage:
  - Dependence on sampling qualities
  - Limited exploration of new responses

Table of Contents

Recap of RLHF and DPO

RRHF

CPO

KTO

IPO

ORPO

SimPO

Experiments (from SimPO)

CPO

Contrastive Preference Optimization: Pushing the Boundaries of LLM Performance in Machine Translation

Haoran Xu[♠] **Amr Sharaf**[♡] **Yunmo Chen**[♠] **Weiting Tan**[♠] **Lingfeng Shen**[♠] **Benjamin Van Durme**[♠]
Kenton Murray[*♠] **Young Jin Kim**[*♡]

Introduction

- Task: Machine Translation
  - Traditional SOTA models: Transformer Encoder-Decoder (e.g., NLLB-200, MT5)
  - Large-scale decoder-only LLMs (GPT-4, LLaMA)

Introduction

- Task: Machine Translation
  - Traditional SOTA models: Transformer Encoder-Decoder (e.g., NLLB-200, MT5)
  - Large-scale decoder-only LLMs (GPT-4, LLaMA)
- Problems:
  - Moderate-sized LLMs (7B, 13B) underperform compared to GPT-4 and WMT winners.
  - Supervised Fine-Tuning (SFT) on reference translation, e.g., ALMA (Xu et al., 2023)

Introduction

- Task: Machine Translation
  - Traditional SOTA models: Transformer Encoder-Decoder (e.g., NLLB-200, MT5)
  - Large-scale decoder-only LLMs (GPT-4, LLaMA)
- Problems:
  - Moderate-sized LLMs (7B, 13B) underperform compared to GPT-4 and WMT winners.
  - Supervised Fine-Tuning (SFT) on reference translation, e.g., ALMA (Xu et al., 2023)
- Goal: Introduce Contrastive Preference Optimization (CPO) to train moderate-sized models beyond SFT limitations and match the performances of GPT-4 and WMT winners.

Shortcomings of SFT

- SFT forces models to mimic reference translations, which could be suboptimal or flawed.

Source: 这是马特利 (Martelly) 四年来第五次入选海地临时选举委员会 (CEP)。

Reference: It is Martelly's fifth CEP in four years.

ALMA-13B-LoRA: This is Martelly's fifth time **being selected by the Provisional Electoral Council** (CEP) in four years.

GPT-4: This is the fifth time Martelly has been **selected for Haiti's Provisional Electoral Council** (CEP) in four years.

Figure: Example from FLORES-200 dataset

Shortcomings of SFT

- SFT forces models to mimic reference translations, which could be suboptimal or flawed.

Source: 这是马特利 (Martelly) 四年来第五次入选海地临时选举委员会 (CEP)。

Reference: It is Martelly's fifth CEP in four years.

ALMA-13B-LoRA: This is Martelly's fifth time **being selected by the Provisional Electoral Council** (CEP) in four years.

GPT-4: This is the fifth time Martelly has been **selected for Haiti's Provisional Electoral Council** (CEP) in four years.

Figure: Example from FLORES-200 dataset

- SFT lacks a mechanism to prevent the model from rejecting mistakes in translations.

CPO-Preference data construction

- Dataset: FLORES-200
- Construct preference translation data using GPT-4, ALMA, and human references

CPO-Preference data construction

- Dataset: FLORES-200
- Construct preference translation data using GPT-4, ALMA, and human references
- Score translations using reference-free evaluation models KIWI-XXL and XCOMET.

CPO-Preference data construction

- Dataset: FLORES-200
- Construct preference translation data using GPT-4, ALMA, and human references
- Score translations using reference-free evaluation models KIWI-XXL and XCOMET.
- Keep highest-scoring and lowest-scoring translations

CPO Objective

- DPO Loss:

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

- Memory-inefficient and speed-inefficient (two models)

CPO Objective

- DPO Loss:

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

- Memory-inefficient and speed-inefficient (two models)
- Set π_{ref} as a uniform prior U:

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

CPO objective

Theorem (Xu et al. (2024))

When $\pi_{\text{ref}} = \pi_w$, an ideal policy that precisely aligns with the true data distribution of the preferred data, i.e. $\pi_w(y_w|x) = 1$, then the DPO loss $\mathcal{L}(\pi_\theta; \pi_w) + C$ is upper bounded by $\mathcal{L}(\pi_\theta; U)$, where C is a constant.

- The approximation in $\mathcal{L}(\pi_\theta; U)$ is effective because it minimizes the upper bound on the DPO loss.

Proof

$$
\begin{aligned}
&\mathcal{L}\left(\pi_\theta; \pi_w\right) \\
&= -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta\left(y_w \mid x\right)}{\pi_w\left(y_w \mid x\right)} - \beta\log\frac{\pi_\theta\left(y_l \mid x\right)}{\pi_w\left(y_l \mid x\right)}\right)\right] \\
&= -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\pi_\theta\left(y_w \mid x\right) - \beta\log\pi_\theta\left(y_l \mid x\right)\right.\right. \\
&\quad \left.\left. + \beta\log\pi_w\left(y_l \mid x\right)\right)\right] \\
&= -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\left(\frac{1}{1+\frac{\pi_\theta(y_l|x)^\beta}{\pi_\theta(y_w|x)^\beta\cdot\pi_w(y_l|x)^\beta}}\right)\right] \\
&= -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\pi_\theta\left(y_w \mid x\right)^\beta + \log\pi_w\left(y_l \mid x\right)^\beta\right. \\
&\quad \left. - \log\left(\pi_\theta\left(y_w \mid x\right)^\beta\cdot\pi_w\left(y_l \mid x\right)^\beta + \pi_\theta\left(y_l \mid x\right)^\beta\right)\right].
\end{aligned}
$$

Proof

$$
\begin{aligned}
&\mathcal{L}'\left(\pi_\theta; \pi_w\right) \\
\triangleq &\mathcal{L}\left(\pi_\theta; \pi_w\right) + \underbrace{\mathbb{E}_{(x,y_l)\sim\mathcal{D}}\left[\log \pi_w\left(y_l \mid x\right)^\beta\right]}_{C \text{ in the Theorem}} \\
= &-\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Bigg[\log \pi_\theta\left(y_w \mid x\right)^\beta \\
&\quad - \log\left(\pi_\theta\left(y_w \mid x\right)^\beta \cdot \pi_w\left(y_l \mid x\right)^\beta + \pi_\theta\left(y_l \mid x\right)^\beta\right) \\
\leq &-\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log \pi_\theta\left(y_w \mid x\right)^\beta - \log\left(\pi_\theta\left(y_w \mid x\right)^\beta \cdot 1 + \pi_\theta\left(y_l \mid x\right)^\beta\right)\right] \\
= &\mathcal{L}\left(\pi_\theta; U\right).
\end{aligned}
$$

CPO Objective

- Add a behavior cloning (BC) regularizer to prevent distribution drift:

$$\min_{\theta} \mathcal{L}\left(\pi_\theta, U\right)$$
$$\text{s.t. } \mathbb{E}_{(x,y_w)\sim\mathcal{D}}\left[\mathbb{KL}\left(\pi_w\left(y_w \mid x\right) \| \pi_\theta\left(y_w \mid x\right)\right)\right] < \epsilon,$$

CPO Objective

- Add a behavior cloning (BC) regularizer to prevent distribution drift:

$$\min_{\theta} \mathcal{L}\left(\pi_\theta, U\right)$$
$$\text{s.t. } \mathbb{E}_{(x,y_w)\sim\mathcal{D}}\left[\mathbb{KL}\left(\pi_w\left(y_w \mid x\right) \| \pi_\theta\left(y_w \mid x\right)\right)\right] < \epsilon,$$

- By Lagrangian duality, equivalent to

$$\min_{\theta} \mathcal{L}\left(\pi_\theta, U\right) + \lambda \cdot \mathbb{E}_{(x,y_w)\sim\mathcal{D}}\left[\mathbb{KL}\left(\pi_w\left(y_w \mid x\right) \| \pi_\theta\left(y_w \mid x\right)\right)\right],$$

CPO Objective

- Add a behavior cloning (BC) regularizer to prevent distribution drift:

$$\min_{\theta} \mathcal{L}\left(\pi_\theta, U\right)$$
$$\text{s.t. } \mathbb{E}_{(x,y_w)\sim\mathcal{D}}\left[\mathbb{KL}\left(\pi_w\left(y_w \mid x\right) \| \pi_\theta\left(y_w \mid x\right)\right)\right] < \epsilon,$$

- By Lagrangian duality, equivalent to

$$\min_{\theta} \mathcal{L}\left(\pi_\theta, U\right) + \lambda \cdot \mathbb{E}_{(x,y_w)\sim\mathcal{D}}\left[\mathbb{KL}\left(\pi_w\left(y_w \mid x\right) \| \pi_\theta\left(y_w \mid x\right)\right)\right],$$

- Set $\lambda = 1$, expand KL divergence with $\pi_w\left(y_w \mid x\right) = 1$ to get CPO loss:

$$\min_{\theta} \underbrace{\mathcal{L}\left(\pi_\theta, U\right)}_{\mathcal{L}_{\text{prefer}}} \underbrace{-\mathbb{E}_{(x,y_w)\sim\mathcal{D}}\left[\log \pi_\theta\left(y_w \mid x\right)\right]}_{\mathcal{L}_{\text{NLL}}}$$

Experimental Setup

- Training Data:
  - 10 translation directions: $\mathrm{cs} \leftrightarrow \mathrm{en}$,$\mathrm{de} \leftrightarrow \mathrm{en}$, $\mathrm{is} \leftrightarrow \mathrm{en}, \mathrm{zh} \leftrightarrow \mathrm{en}, \mathrm{ru} \leftrightarrow \mathrm{en}$
  - 20K paired sentences derived from FLORES-200
- Test data: WMT'21, WMT'22, WMT;23
- Baselines:
  - ALMA-13B-LoRA (Base Model)
  - GPT-4 Zero-shot Translation
  - WMT'21, WMT'22 Competition Winners
  - SFT & DPO Models
- Reference-free evaluation models: KIWI-XXL, XCOMET

Experiments

Models	de			cs			is		
	KIWI-22	KIWI-XXL	XCOMET	KIWI-22	KIWI-XXL	XCOMET	KIWI-22	KIWI-XXL	XCOMET
Gold Reference	82.67	84.01	**97.85**	83.19	81.83	90.27	80.51	85.20	91.52
WMT Winners	**83.56**	83.70	96.99	85.31	**87.27**	**94.38**	81.77	84.94	91.61
GPT-4	83.48	**84.91**	97.56	84.81	85.35	93.48	81.03	81.21	90.00
ALMA-13B-LoRA	82.62	81.64	96.49	84.14	84.24	92.38	81.71	83.31	91.20
+ SFT on preferred data	82.75	81.85	96.67	84.14	83.46	91.99	81.48	82.11	90.30
+ DPO	82.40	81.20	96.40	83.86	83.45	91.68	81.43	82.66	90.33
+ CPO (Ours, ALMA-13B-R)	83.28	84.25	97.48	**84.99**	87.06	93.61	**82.18**	**85.68**	**91.93**

Models	zh			ru			Avg.		
	KIWI-22	KIWI-XXL	XCOMET	KIWI-22	KIWI-XXL	XCOMET	KIWI-22	KIWI-XXL	XCOMET
Gold Reference	80.92	81.70	90.42	82.96	84.62	94.17	82.05	83.47	92.85
WMT Winners	82.04	81.13	91.14	**84.35**	87.01	94.79	**83.41**	84.81	93.78
GPT-4	81.73	81.53	90.79	83.64	86.15	94.3	82.94	83.83	93.23
ALMA-13B-LoRA	80.82	79.96	89.92	83.10	84.17	93.79	82.48	82.66	92.76
+ SFT on preferred data	81.25	80.51	90.18	83.23	84.15	93.54	82.57	82.42	92.54
+ DPO	80.74	79.64	89.58	82.94	83.40	93.25	82.27	82.07	92.25
+ CPO (Ours, ALMA-13B-R)	**82.25**	**84.32**	**92.03**	83.98	**87.37**	**95.22**	83.34	**85.74**	**94.05**

Figure: overall results in $\mathrm{en} \rightarrow \mathrm{xx}$ for WMT'21 and WMT'22

- Incorporating CPO, performance of ALMA matches or even surpass that of GPT-4 and WMT winners
- SFT enhances ALMA's performance for $\mathrm{xx} \rightarrow \mathrm{en}$ but decreases for $\mathrm{en} \rightarrow \mathrm{xx}$. Similarly, DPO slightly decreases performance.
- CPO consistently improves performance across all metrics.

Experiments

Models	de			cs			is		
	KIWI-22	KIWI-XXL	XCOMET	KIWI-22	KIWI-XXL	XCOMET	KIWI-22	KIWI-XXL	XCOMET
Gold Reference	78.74	78.56	88.82	82.08	83.11	84.60	80.88	85.04	76.16
WMT Winners	81.38	83.59	93.74	82.47	82.53	85.65	81.39	85.60	78.14
GPT-4	**81.50**	**84.58**	**94.47**	82.52	83.55	**88.48**	81.49	**85.90**	**81.11**
ALMA-13B-LoRA	81.14	83.57	93.30	81.96	82.97	83.95	80.90	85.49	76.68
+ SFT on preferred data	81.36	83.98	93.84	82.36	83.15	86.67	81.32	85.61	80.20
+ DPO	81.13	83.52	93.25	81.82	82.69	83.84	80.89	85.22	76.09
+ CPO (Ours, ALMA-13B-R)	**81.50**	83.97	94.20	**82.63**	**83.75**	88.03	**81.57**	85.73	80.49

Models	zh			ru			Avg.		
	KIWI-22	KIWI-XXL	XCOMET	KIWI-22	KIWI-XXL	XCOMET	KIWI-22	KIWI-XXL	XCOMET
Gold Reference	77.09	74.19	90.70	80.74	79.59	88.56	79.91	80.10	85.77
WMT Winners	77.66	73.28	87.2	81.71	80.97	90.91	80.92	81.19	87.13
GPT-4	**79.33**	**77.65**	**92.06**	81.57	81.34	90.95	81.28	**82.60**	**89.41**
ALMA-13B-LoRA	77.32	74.41	89.88	81.31	81.05	89.89	80.53	81.50	86.74
+ SFT on preferred data	78.32	76.03	90.65	81.46	81.17	90.65	80.96	81.99	88.40
+ DPO	77.50	74.50	89.94	81.19	80.88	89.76	80.51	81.36	86.58
+ CPO (Ours, ALMA-13B-R)	79.24	77.17	91.65	**81.72**	**81.54**	**91.18**	**81.33**	82.43	89.11

Figure: overall results in $\mathrm{xx} \rightarrow \mathrm{en}$ for WMT'21 and WMT'22

- Incorporating CPO, performance of ALMA matches or even surpass that of GPT-4 and WMT winners
- SFT enhances ALMA's performance for $\mathrm{xx} \rightarrow \mathrm{en}$ but decreases for $\mathrm{en} \rightarrow \mathrm{xx}$. Similarly, DPO slightly decreases performance.
- CPO consistently improves performance across all metrics.

More Results

	KIWI-22	KIWI-XXL	XCOMET
Gold Reference	78.74	75.56	86.30
WMT Winners	**80.57**	77.72	88.24
TowerInstruct	80.31	77.18	88.11
ALMA-13B-LoRA	79.48	76.00	87.16
+ CPO (Ours, ALMA-13B-R)	80.55	**78.97**	**89.74**

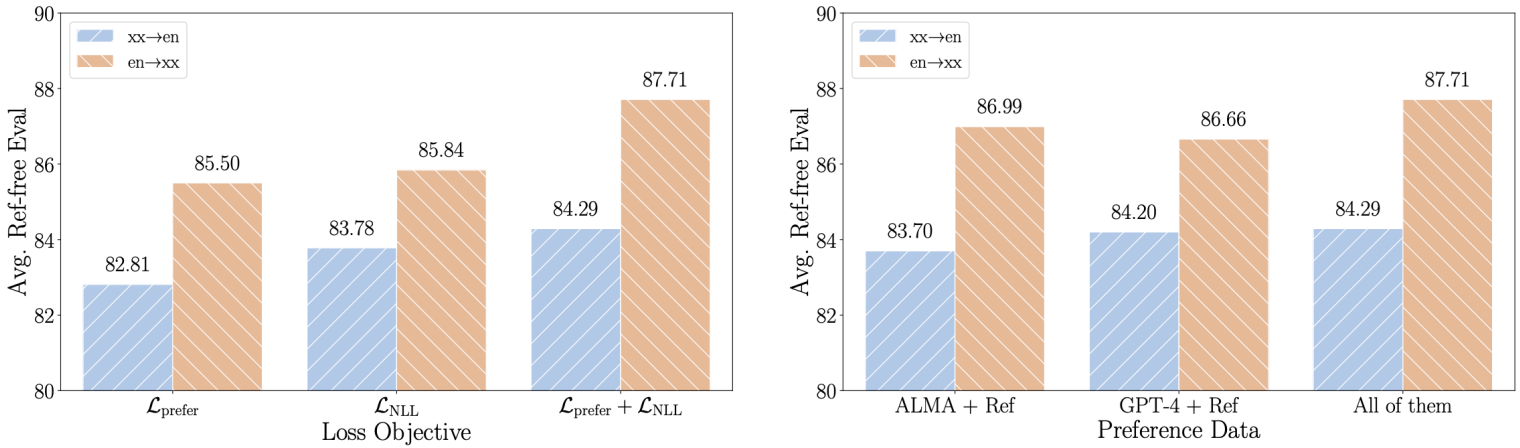
Figure: Overall Results for WMT'23

Table 7. The results of human evaluation on sampled `zh`→`en` WMT'22 test data. ↑ indicates that higher values are better, while ↓ indicates that lower values are better.

	Avg. score ↑	Avg. rank ↓	Avg. win ratio (%)	Ties (%)
ALMA-13B-LoRA	4.86	1.60	62.50	40.30
ALMA-13B-R	**5.16**	**1.40**	**77.80**	40.30

Figure: Human Evaluation

Ablation Study



- Removing L_{prefer} or L_{NLL} leads to lower performance.
- ALMA data is essential for $\text{en} \rightarrow \text{xx}$ while GPT-4's data is essential for $\text{xx} \rightarrow \text{en}$.

Ablation Study

Table 18. The impact of applying $\mathcal{L}_{\text{NLL}}$ to the original DPO loss.

Loss Objective	KIWI-22	KIWI-XXL	XCOMET	Memory Cost	FLOPs/tok
Translating to English (xx→en)					
$\mathcal{L}_{\text{DPO}}$	80.51	81.36	86.58	2×	2×
$\mathcal{L}_{\text{DPO}} + \mathcal{L}_{\text{NLL}}$	81.28	82.42	89.05	2×	2×
$\mathcal{L}_{\text{prefer}} + \mathcal{L}_{\text{NLL}}$ (CPO)	**81.33**	**82.43**	**89.11**	1×	1×
Translating from English (en→xx)					
$\mathcal{L}_{\text{DPO}}$	82.27	82.07	92.25	2×	2×
$\mathcal{L}_{\text{DPO}} + \mathcal{L}_{\text{NLL}}$	83.13	84.74	93.53	2×	2×
$\mathcal{L}_{\text{prefer}} + \mathcal{L}_{\text{NLL}}$ (CPO)	**83.34**	**85.74**	**94.05**	1×	1×

- Removing L_{prefer} or L_{NLL} leads to lower performance.
- ALMA data is essential for $\text{en} \rightarrow \text{xx}$ while GPT-4's data is essential for $\text{xx} \rightarrow \text{en}$.
- $\mathcal{L}_{\text{prefer}}$ is a successful approximation of the DPO loss, offering savings in memory and speed, and it can even outperform the original BC-regularized DPO loss $\mathcal{L}_{\text{DPO}} + \mathcal{L}_{\text{NLL}}$.

Table of Contents

Recap of RLHF and DPO

RRHF

CPO

KTO

IPO

ORPO

SimPO

Experiments (from SimPO)

KTO: Model Alignment as Prospect Theoretic Optimization

Kawin Ethayarajh [1] **Winnie Xu** [2] **Niklas Muennighoff** [2] **Dan Jurafsky** [1] **Douwe Kiela** [1 2]

Background: Prospect Theory

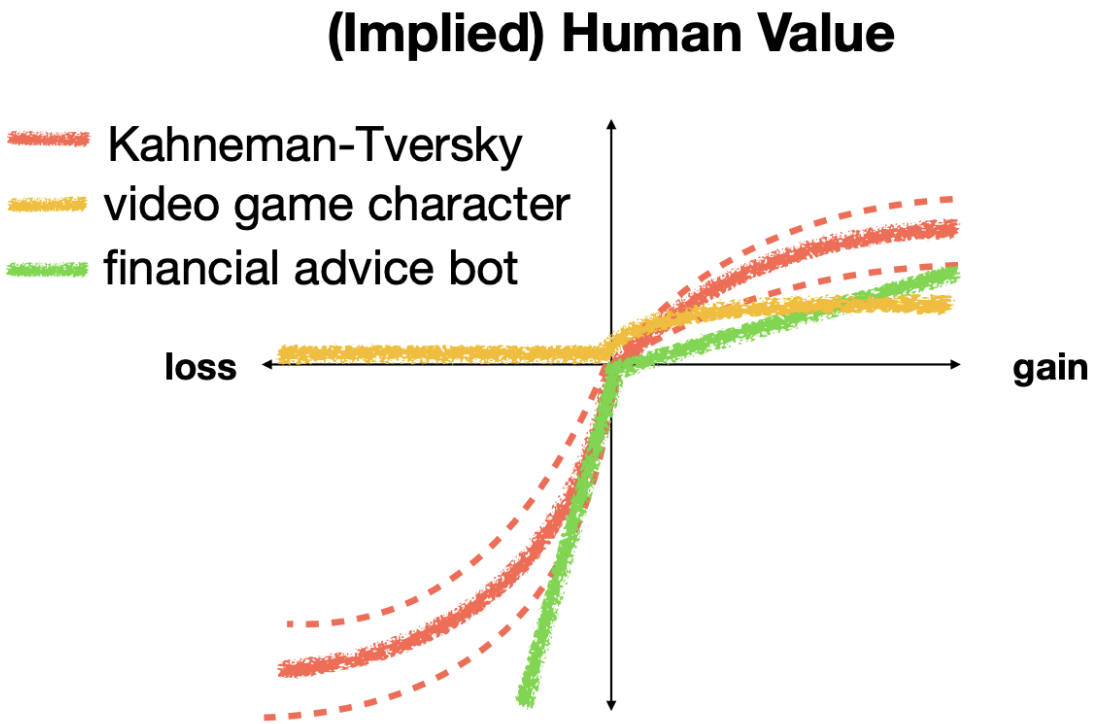
- Key Point: Relative to a **reference point**, humans are more sensitive to losses than gains, a property called **loss aversion**.
- Tversky and Kahneman (1992) proposed the following functional form for human value: ($\alpha \approx 0.88, \lambda \approx 2.25$)

$$v\left(z;\lambda,\alpha,z_0\right) = \begin{cases} \left(z-z_0\right)^{\alpha} & \text{if } z \geq z_0 \\ -\lambda\left(z_0-z\right)^{\alpha} & \text{if } z < z_0 \end{cases}$$

Background: Prospect Theory

- Key Point: Relative to a **reference point**, humans are more sensitive to losses than gains, a property called **loss aversion**.
- Tversky and Kahneman (1992) proposed the following functional form for human value: ($\alpha \approx 0.88, \lambda \approx 2.25$)

$$v(z; \lambda, \alpha, z_0) = \begin{cases} (z - z_0)^{\alpha} & \text{if } z \geq z_0 \\ -\lambda (z_0 - z)^{\alpha} & \text{if } z < z_0 \end{cases}$$



Why PPO/DPO Works?

- Popular alignment methods such as DPO and PPO-Clip implicitly model those human biases

Why PPO/DPO Works?

- Popular alignment methods such as DPO and PPO-Clip implicitly model those human biases
- Human-aware losses (HALOs):

Definition 3.4 (HALOs). Let θ denote the trainable parameters of the model $\pi_\theta : \mathcal{X} \to \mathcal{P}(\mathcal{Y})$ being aligned, π_{ref} the reference model, $l : \mathcal{Y} \to \mathbb{R}^+$ a normalizing factor, and $r_\theta(x, y) = l(y) \log[\pi_\theta(y|x)/\pi_{\text{ref}}(y|x)]$ the implied reward. Where $Q(Y'|x)$ is a reference point distribution over $\mathcal{Y}$ and $v : \mathbb{R} \to \mathbb{R}$ is non-decreasing everywhere and concave in $(0, \infty)$, the *human value* of (x, y) is

$$v(r_\theta(x, y) - \mathbb{E}_Q[r_\theta(x, y')]) \tag{5}$$

A function f is a *human-aware loss* for v if $\exists\ a_{x,y} \in \{-1, +1\}$ such that:

$$f(\pi_\theta, \pi_{\text{ref}}) = \mathbb{E}_{x,y\sim\mathcal{D}}[a_{x,y} v(r_\theta(x, y) - \mathbb{E}_Q[r_\theta(x, y')])] + C_\mathcal{D} \tag{6}$$

where $\mathcal{D}$ is the feedback data and $C_\mathcal{D} \in \mathbb{R}$ is a data-specific constant.

HALOs

- DPO and PPO-Clip are HALOs
- Example of DPO:

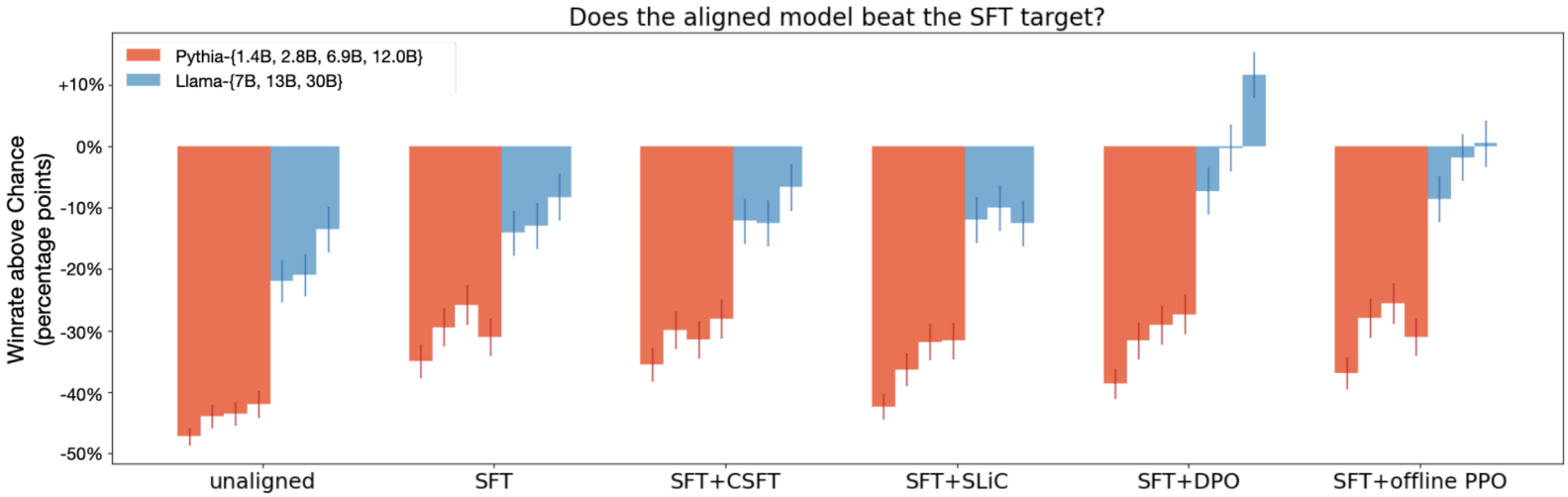
The DPO loss is

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

where $\beta > 0$ is a hyperparameter. DPO meets the criteria with the following construction: $l(y) = \beta$; $r_\theta = \beta\log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$; $v(\cdot) = \log\sigma(\cdot)$ is increasing and concave everywhere; Q places all mass on (x, y_l), where y_l is a dispreferred output for x such that $y \succ y_l$; and $a_{x,y} = -1$.

- Conditional SFT (CSFT) and Sequence Likelihood Calibration (SLiC) are not HALOs.

Does being a HALO matter?



- HALOs either match or outperform non-HALOs at every scale
- Up to a scale of 7B parameters, alignment provides virtually no gains over SFT alone
- Despite only using dummy $+1/\text{-}1$ rewards, the offline PPO variant performs as well as DPO except for Llama-30B.

KTO

- KTO: a preference-free HALO that takes binary feedback as input and directly maximizes the utility of generations

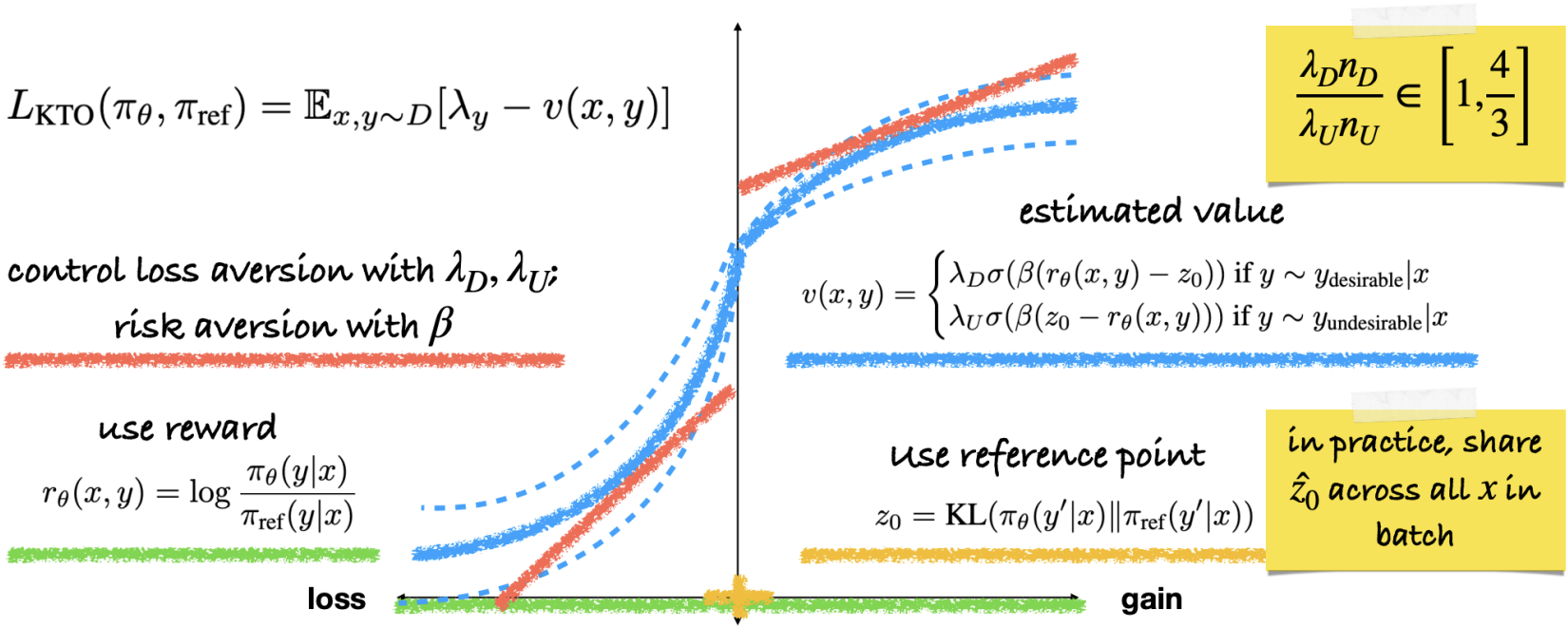
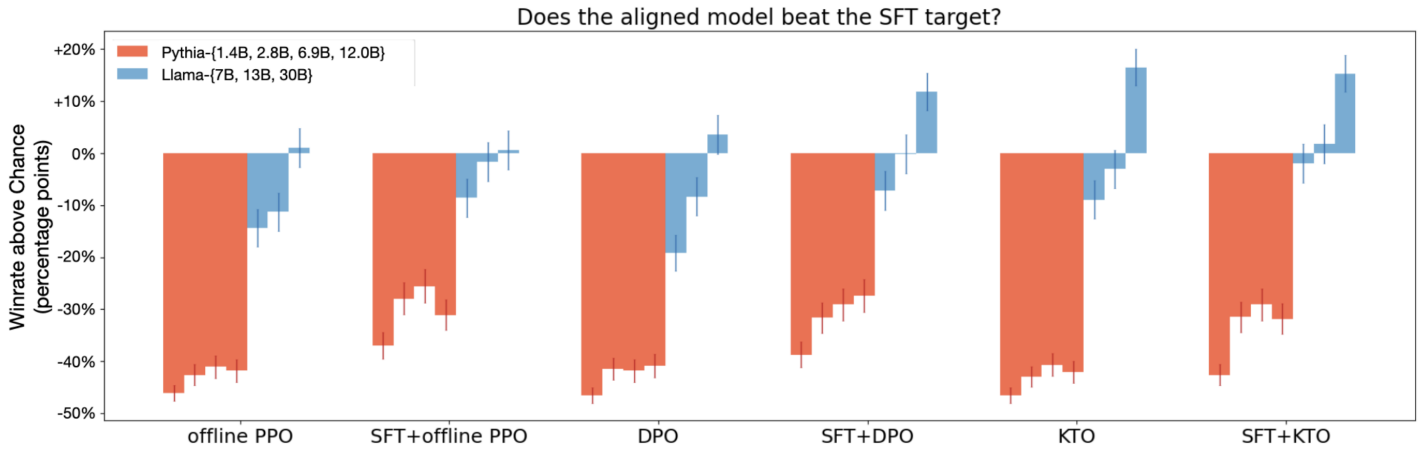


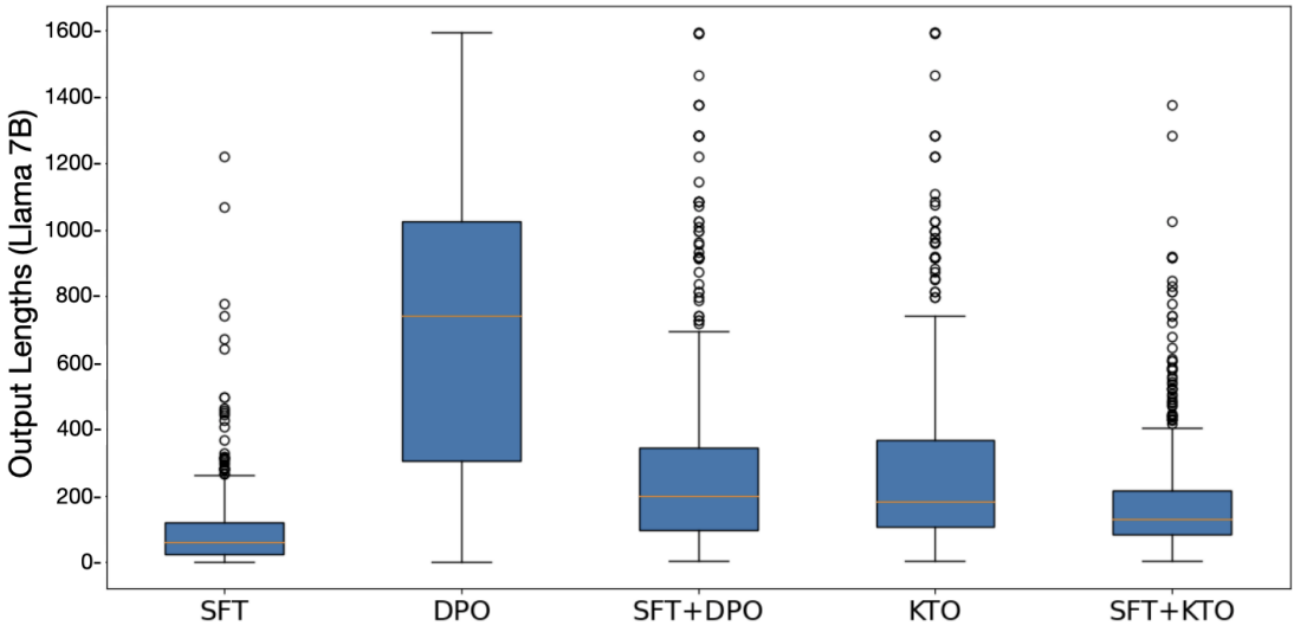
Figure: Kawin's Slide on KTO Loss

Experiments



- KTO matches or exceeds the performance of DPO

Experiments



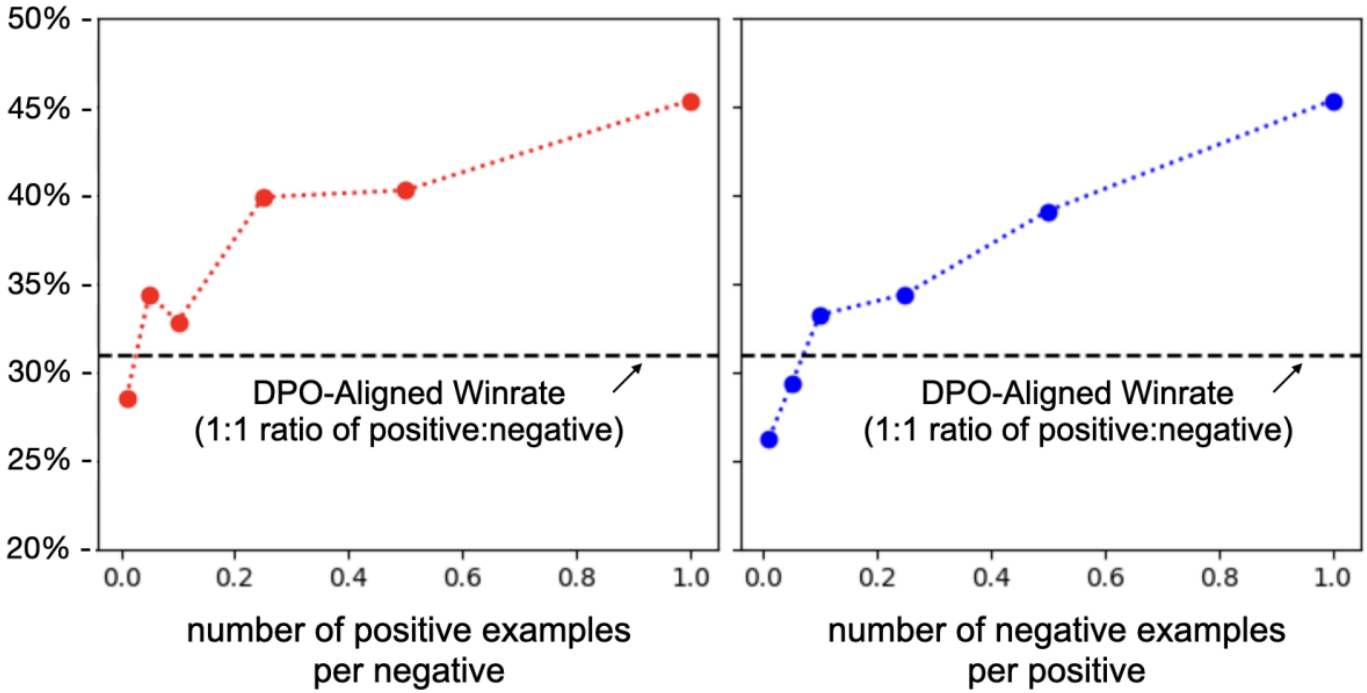
- KTO matches or exceeds the performance of DPO
- At sufficient scale, KTO does not need SFT

Experiments

Dataset (→) Metric (→)	MMLU EM	GSM8k EM	HumanEval pass@1	BBH EM
SFT	57.2	39.0	30.1	46.3
DPO	58.2	40.0	30.1	44.1
ORPO ($\lambda = 0.1$)	57.1	36.5	29.5	47.5
KTO ($\beta = 0.1, \lambda_D = 1$)	**58.6**	**53.5**	**30.9**	**52.6**
KTO (one-y-per-x)	58.0	50.0	30.7	49.9
KTO (no z_0)	58.5	49.5	30.7	49.0
KTO (concave, $v = \log \sigma$)	58.3	42.5	30.6	43.2
KTO (risk-neutral, $v(\cdot) = \cdot$)	57.3	42.0	28.8	6.1
KTO (no π_{ref}, $\lambda_D = 1.75$)	57.5	47.5	29.5	51.6

- KTO matches or exceeds the performance of DPO
- At sufficient scale, KTO does not need SFT
- KTO is also better than DPO on other benchmarks, e.g. GSM8K, a mathematical reasoning dataset
- Changing the design of KTO makes it significantly worse

More Results



- KTO can handle highly imbalanced datasets by adjusting

$$\frac{\lambda_D n_D}{\lambda_U n_U} \in \left[1, \frac{3}{2}\right]$$

DPO vs KTO: Which alignment method to use

- When human feedback is in a binary format $\rightarrow$ KTO

DPO vs KTO: Which alignment method to use

- When human feedback is in a binary format $\rightarrow$ KTO
- When data is in the form of preferences:
  - sufficiently little noise and intransitivity $\rightarrow$ DPO
  - high enough noise and intransitivity $\rightarrow$ KTO
  - Most publicly available preference datasets (e.g., SHP, OpenAssistant) contain noisy feedback

DPO vs KTO: Which alignment method to use

- When human feedback is in a binary format $\rightarrow$ KTO
- When data is in the form of preferences:
  - sufficiently little noise and intransitivity $\rightarrow$ DPO
  - high enough noise and intransitivity $\rightarrow$ KTO
  - Most publicly available preference datasets (e.g., SHP, OpenAssistant) contain noisy feedback
- Open research: design your HALO. There is no one-loss-fits-all!

Table of Contents

Recap of RLHF and DPO

RRHF

CPO

KTO

IPO

ORPO

SimPO

Experiments (from SimPO)

IPO

A General Theoretical Paradigm to Understand Learning from Human Preferences

Mohammad Gheshlaghi Azar **Mark Rowland** **Bilal Piot**
Daniel Guo **Daniele Calandriello** **Michal Valko** **Rémi Munos**
Google DeepMind

IPO: overview

- IPO = Identity Preference Optimization algorithm
- A special case of ΨPO, when $\Psi = \mathrm{id}$
- Proposed by Azar et al. (2024) (DeepMind)
- ΨPO is a general optimization framework that learns directly from preference data, and encompasses RLHF/DPO as a special case
- IPO is advantageous over DPO in the case of deterministic preferences
- Relies on a reference model

ΨPO: objective

- Given a reference model π, behavior policy μ, nondecreasing $\Psi : [0, 1] \to \mathbb{R}$, we have the **ΨPO objective**:

$$\max_{\pi} \mathop{\mathbb{E}}_{\substack{x\sim\rho \\ y\sim\pi(.|x) \\ y'\sim\mu(.|x)}} [\Psi(p^*(y \succ y'|x))] - \tau D_{\mathrm{KL}}(\pi \,||\, \pi_{\mathrm{ref}})\,.$$

- When $\Psi = \mathrm{id}$, we obtain IPO
- However, note that at this stage, one would need to learn $p^*(y \succ y'|x)$ in order to actually optimize this objective
- We will avoid this with some tricks

ΨPO: recovering RLHF/DPO

- **Claim:** if $\Psi(q) = \log \frac{q}{1-q}$ is the logit map, and if the BT model holds, we recover the optimal policy for RLHF/DPO.
- Proof: since BT holds, we have:

$$p^*(y \succ y'|x) = \sigma(r(x,y) - r(x,y'))$$

Apply Ψ to both sides:

$$\Psi(p^*(y \succ y'|x)) = (\Psi \circ \sigma)(r(x,y) - r(x,y')) = r(x,y) - r(x,y')$$

The objective becomes:

$$\mathbb{E}[r(x,y) - r(x,y')] - \tau D_{KL}(\pi||\pi_{ref}) \simeq \mathbb{E}[r(x,y)] - \tau D_{KL}(\pi||\pi_{ref})$$

Issues with $\Psi = \text{logit}$

- The main criticism of taking $\Psi = \text{logit}$: overfitting in the case of "deterministic preferences"
- Imagine that $p^*(y \succ y'|x) = 1$ for some y, y'. Then because $\Psi(1) = +\infty$, we have $r(x, y) - r(x, y') = +\infty$, which implies $\frac{\pi^*(y'|x)}{\pi^*(y|x)} = 0$, regardless of the regularization strength τ

$$\pi^*(y) \propto \pi_{\text{ref}}(y) \exp\left(\tau^{-1} \mathbb{E}_{y' \sim \mu}[\Psi(p^*(y \succ y'))]\right).$$

- In finite samples, a given y might "win" (or "lose") in 100% of its comparisons
- Thus, it is natural to consider alternatives that are bounded on $[0, 1]$, like $\Psi = \text{id}$

IPO: rewriting objective

- **Simplification 1:** use the "DPO trick" to reduce it to solving an equation. Then, solve this equation by minimizing $(\text{LHS} - \text{RHS})^2$
- Fix a single context x throughout.
  Defining the "reward" $g(y) := \mathbb{E}_{y' \sim \mu}[\Psi(p^*(y \succ y'))]$:

$$\pi^*(y) \propto \pi_{\text{ref}}(y) \exp(\tau^{-1} g(y)) .$$

$$\frac{\pi^*(y)}{\pi^*(y')} = \frac{\pi_{\text{ref}}(y)}{\pi_{\text{ref}}(y')} \exp(\tau^{-1}(g(y) - g(y'))) .$$

$$h^*(y, y') = \log \left(\frac{\pi^*(y)\pi_{\text{ref}}(y')}{\pi^*(y')\pi_{\text{ref}}(y)} \right)$$

$$h^*(y, y') = \tau^{-1}\big(g(y) - g(y')\big) .$$

$$h_\pi(y, y') = \log \left(\frac{\pi(y)\pi_{\text{ref}}(y'}{\pi(y')\pi_{\text{ref}}(y)} \right) ,$$

$$h_\pi(y, y') = \tau^{-1}\big(g(y) - g(y')\big) .$$

IPO: rewriting objective

- **Simplification 1**, continued: specialize to $\Psi = \mathrm{id}$:

$$h_\pi(y, y') = \tau^{-1}\big(g(y) - g(y')\big)\,.$$

$$h_\pi(y, y') = \tau^{-1}\big(p^*(y \succ \mu) - p^*(y' \succ \mu)\big)\,.$$

  where we use the notation $p^*(y \succ \mu) := \mathbb{E}_{y' \sim \mu}[p^*(y \succ y')]$
- We would like to solve this equation for π. Assuming that a solution exists, this is equivalent to minimizing $(\mathrm{LHS} - \mathrm{RHS})^2$ with respect to π. We arrive at the **preliminary IPO objective**:

$$L(\pi) = \mathop{\mathbb{E}}_{y,y' \sim \mu}\left[\left(h_\pi(y, y') - \frac{p^*(y \succ \mu) - p^*(y' \succ \mu)}{\tau}\right)^2\right].$$

- Intuition for h_π: write $h_\pi(y, y') = \log \frac{\pi(y)}{\pi_{ref}(y)} - \log \frac{\pi(y')}{\pi_{ref}(y')}$

IPO: rewriting objective

- **Simplification 2**: replace $p^*(y \succ \mu) - p^*(y' \succ \mu)$ with the random variable $I(y, y') \sim \text{Bern}(p^*(y \succ y'))$
- **Population IPO objective**:

$$\mathop{\mathbb{E}}_{y,y' \sim \mu} \left[\left(h_\pi(y, y') - \tau^{-1} I(y, y') \right)^2 \right],$$

where we also average over the Bernoulli randomness

IPO: rewriting objective

- Let's prove that the objectives agree, up to an additive constant:

$$\begin{aligned}&\mathbb{E}[(h_\pi(y,y') - \tau^{-1}(p^*(y \succ \mu) - p^*(y' \succ \mu)))^2] \\ &\simeq \mathbb{E}[(h_\pi(y,y') - \tau^{-1} I(y,y'))^2]\end{aligned}$$

- It suffices to consider the cross-term:

$$\mathbb{E}[h_\pi(y,y')(p^*(y \succ \mu) - p^*(y' \succ \mu)))] = \mathbb{E}[h_\pi(y,y') I(y,y')]$$

- Note that

$$h_\pi(y,y') = \log \frac{\pi(y)}{\pi_{ref}(y)} - \log \frac{\pi(y')}{\pi_{ref}(y')} =: f(y) - f(y')$$

is additively separable. Also, $h_\pi(y,y') = -h_\pi(y',y)$

IPO: rewriting objective

- (Continuing the proof:)

$$
\begin{aligned}
\text{RHS} &= \mathbb{E}[h_\pi(y, y')I(y, y')] \\
&= \mathbb{E}[h_\pi(y, y')p^*(y \succ y')] \\
&= \mathbb{E}[f(y)p^*(y \succ y')] - \mathbb{E}[f(y')p^*(y \succ y')] \\
&= \mathbb{E}[f(y)p^*(y \succ y')] - \mathbb{E}[f(y')(1 - p^*(y' \succ y))] \\
&= \mathbb{E}[f(y)p^*(y \succ y')] + \mathbb{E}[f(y')p^*(y' \succ y)] - \mathbb{E}[f(y')] \\
&= \mathbb{E}[f(y)(2p^*(y \succ y') - 1)]
\end{aligned}
$$

IPO: rewriting objective

- (Continuing the proof:)

$$
\begin{aligned}
\text{LHS} &= \mathbb{E}[h_\pi(y, y')(p^*(y \succ \mu) - p^*(y' \succ \mu))] \\
&= \mathbb{E}[h_\pi(y, y')p^*(y \succ \mu)] - \mathbb{E}[h_\pi(y, y')p^*(y' \succ \mu)] \\
&= \mathbb{E}[h_\pi(y, y')p^*(y \succ \mu)] + \mathbb{E}[h_\pi(y', y)p^*(y' \succ \mu)] \\
&= \mathbb{E}[h_\pi(y, y') \cdot 2p^*(y \succ \mu)] \\
&= \mathbb{E}[f(y) \cdot 2p^*(y \succ \mu)] - \mathbb{E}[f(y') \cdot 2p^*(y \succ \mu)] \\
&= \mathbb{E}[f(y) \cdot 2p^*(y \succ \mu)] - \mathbb{E}[f(y')]\mathbb{E}[2p^*(y \succ \mu)] \\
&= \mathbb{E}[f(y) \cdot 2p^*(y \succ y')] - \mathbb{E}[f(y')] \cdot 2 \cdot \frac{1}{2} \\
&= \mathbb{E}[f(y)(2p^*(y \succ y') - 1)],
\end{aligned}
$$

which agrees with the RHS

IPO: empirical objective

- Consider a single x. Given a preference dataset $\{(y_w^{(i)}, y_l^{(i)}) : i \in [N]\}$, we augment the dataset by including the "flip" of each (y_w, y_l). The first term of the objective becomes:

$$\frac{1}{2}(h_\pi - \tau^{-1} \cdot 1)^2 + \frac{1}{2}(h_\pi - \tau^{-1} \cdot 0)^2 \simeq \left(h_\pi - \frac{\tau^{-1}}{2}\right)^2$$

- **Empirical IPO loss**:

$$\mathop{\mathbb{E}}_{(y_w, y_l) \sim D} \left[\left(h_\pi(y_w, y_l) - \frac{\tau^{-1}}{2} \right)^2 \right].$$

- Intuition: we're fitting the difference $h_\pi(y, y') = f(y) - f(y')$ in log-likelihood ratios to $\frac{\tau^{-1}}{2}$ using least-squares
- (SimPO, which we will discuss later, calls $\frac{\tau^{-1}}{2}$ a "margin")

IPO: effect of regularization

- Toy dataset of size three: $y_1 \succ y_2$, $y_1 \succ y_3$, and $y_2 \succ y_3$
- Uniform reference model: $\pi_{ref} = \frac{1}{3}\delta_{y_1} + \frac{1}{3}\delta_{y_2} + \frac{1}{3}\delta_{y_3}$
- IPO regularization τ has an effect, unlike DPO, which converges to $\pi^* = \delta_{y_1}$, regardless of τ

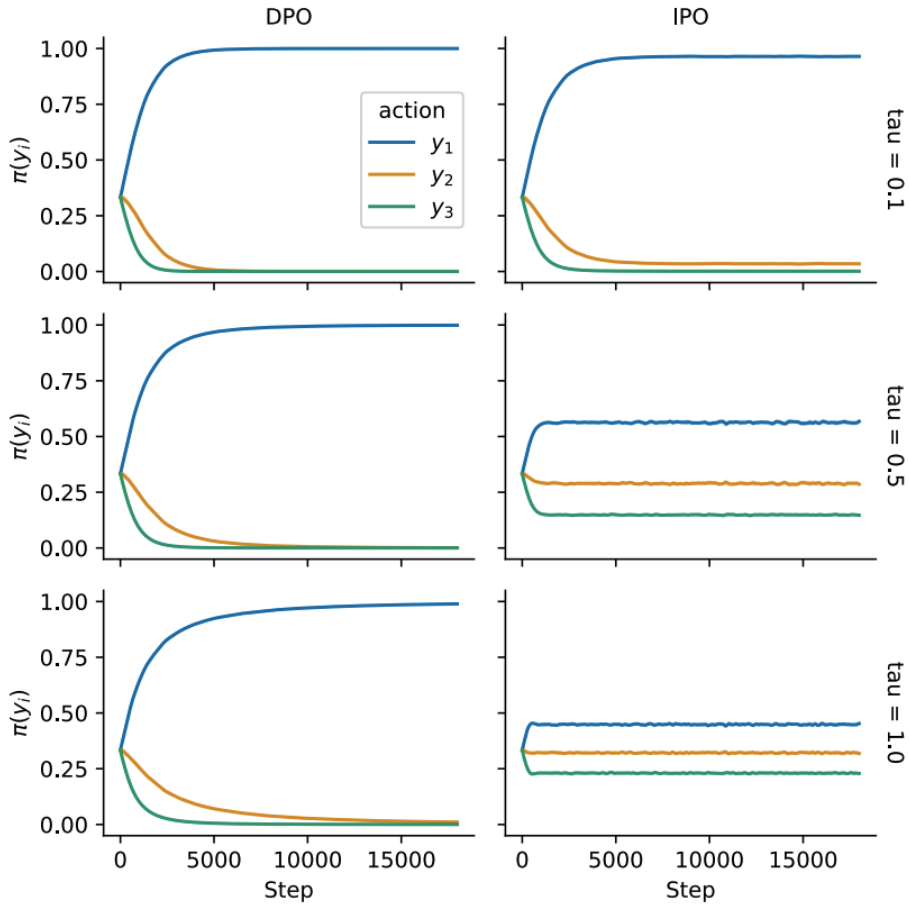


Figure 1: Comparison Between the Learning Curves of Action Probabilities of `IPO` and `DPO` for $\mathcal{D}_1$

Table of Contents

Recap of RLHF and DPO

RRHF

CPO

KTO

IPO

ORPO

SimPO

Experiments (from SimPO)

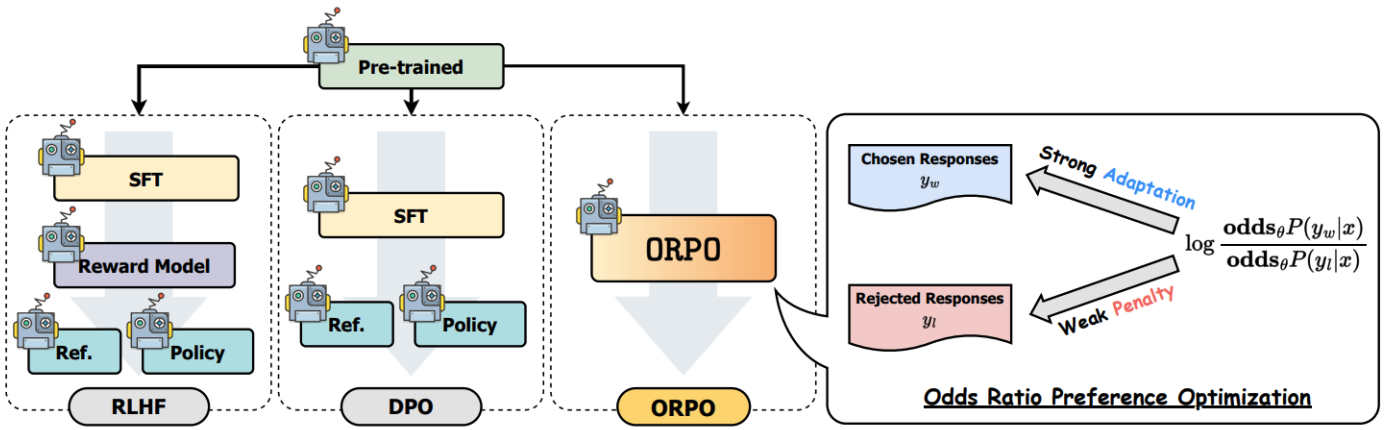
ORPO: Monolithic Preference Optimization without Reference Model

Jiwoo Hong **Noah Lee** **James Thorne**

KAIST AI

ORPO: overview

- ORPO = Odds Ratio Preference Optimization algorithm
- Proposed by Hong et al. (2024) (KAIST)
- Prior works use a two-stage alignment procedure: (1) SFT, (2) Preference Optimization
- ORPO: merge the Preference Optimization step into the SFT step with an added penalty
- Since this is a one-stage procedure, no need for a reference model



ORPO: choice of penalty

- SFT by itself increases the log-probs of tokens in the finetuning dataset, but is unconstrained on unseen tokens
- Empirically, SFT increases the log-probs of unwanted generations



Figure 3: Log probabilities for chosen and rejected responses during OPT-350M model fine-tuning on HH-RLHF dataset. Despite only chosen responses being used for supervision, rejected responses show a comparable likelihood of generation.

ORPO: choice of penalty

- To prevent this, we use the **log odds ratio**, defined as:

$$\mathbf{OR}_\theta(y_w, y_l) = \frac{\mathbf{odds}_\theta(y_w|x)}{\mathbf{odds}_\theta(y_l|x)}$$

where $P_\theta(y|x)$ and $\text{odds}_\theta(y|x)$ are defined as:

$$\log P_\theta(y|x) = \frac{1}{m}\sum_{t=1}^{m} \log P_\theta(y_t|x, y_{<t})$$

$$\mathbf{odds}_\theta(y|x) = \frac{P_\theta(y|x)}{1 - P_\theta(y|x)}$$

- Intuition: the reward associated to y is $r_\theta(x, y) = \log \text{odds}_\theta(y|x)$
- (The intuitive choice, the probability ratio, is “too harsh" of a penalty, according to the authors)

ORPO: objective

- **ORPO objective**:

$$\mathcal{L}_{ORPO} = \mathbb{E}_{(x,y_w,y_l)} \left[\mathcal{L}_{SFT} + \lambda \cdot \mathcal{L}_{OR} \right]$$

where the penalty is given by

$$\mathcal{L}_{OR} = -\log \sigma \left(\log \frac{\mathbf{odds}_\theta(y_w|x)}{\mathbf{odds}_\theta(y_l|x)} \right)$$

- Note that we pass the log odds ratio through $-\log \sigma(\cdot)$ to obtain a negative log likelihood

ORPO: effect on unwanted generations

- ORPO performs finetuning while reducing the likelihood of unwanted generations

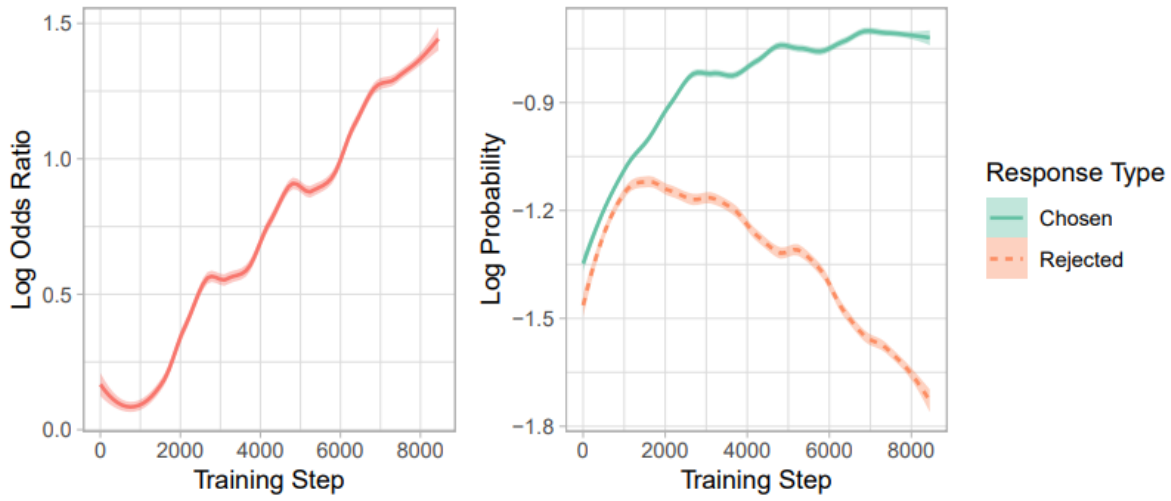


Figure 7: Average log-likelihood for chosen and rejected responses and log odds ratio per batch. The odds consistently increase during training with ORPO.

Table of Contents

Recap of RLHF and DPO

RRHF

CPO

KTO

IPO

ORPO

SimPO

Experiments (from SimPO)

SimPO: Simple Preference Optimization with a Reference-Free Reward

Yu Meng[1]* **Mengzhou Xia**[2]* **Danqi Chen**[2]
[1]Computer Science Department, University of Virginia
[2]Princeton Language and Intelligence (PLI), Princeton University

SimPO: overview

- SimPO = Simple Preference Optimization algorithm
- Proposed by Meng et al. (2024) (UVA, Princeton)
- Replaces the implicit reward model in DPO with the average log-prob of y's tokens given x
- Also, includes a reward margin γ (aka “home advantage")
- No need for a reference model
- Performs an extensive experimental comparison of previous methods

SimPO: objective

- Recall the DPO objective:

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

- The implicit reward model in DPO is:

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

- **Issue**: $r(x, y_w) > r(x, y_l)$ does not imply $p_\theta(y_w|x) > p_\theta(y_l|x)$, where

$$p_\theta(y \mid x) = \frac{1}{|y|}\log\pi_\theta(y \mid x) = \frac{1}{|y|}\sum_{i=1}^{|y|}\log\pi_\theta(y_i \mid x, y_{<i}).$$

  is the average log likelihood
- "In our experiments, we observed that only 50% of the triples from the training set satisfy this condition when trained with DPO."
- **Idea**: simply set $r(x, y)$ to be $\beta p_\theta(y|x)$, for some $\beta > 0$

SimPO: objective

- **SimPO objective**:

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

- Here $\gamma > 0$ is a reward margin term. (Not clear why they included it, but it helps)
- Note that we no longer require a reference model
- Choice of hyperparameters: $\beta \in [2.0, 2.5]$, $\gamma \in [0.5, 1.5]$
- No explicit KL regularization, but empirically low KL divergence from the base model

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

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

Table of Contents

Recap of RLHF and DPO

RRHF

CPO

KTO

IPO

ORPO

SimPO

Experiments (from SimPO)

Models

- Two setups: **Base** and **Instruct**
- Llama-3-8B and Mistral-7B
- **Base**: finetune on the UltraChat-200k dataset, do preference optimization with the UltraFeedback dataset
- **Instruct**: treat Llama-3-Instruct + Mistral-Instruct as post-SFT models; **regenerate** the Instruct preference data by sampling from the Instruct models
  - To select y_w and y_l, they score five generations using the pairwise reward model PairRM from LLM-Blender, and take the top + bottom
- Yields Llama-3-Base, Llama-3-Instruct, Mistral-Base, and Mistral-Instruct

Benchmarks

- Open-ended instruction-following benchmarks:
  - AlpacaEval 2 (805 questions from 5 datasets)
  - MT-Bench (multi-turn, 8 categories with 80 questions)
  - Arena-Hard v0.1 (a hard version of MT-Bench, 500 technical queries)

	# Exs.	Baseline Model	Judge Model	Scoring Type	Metric
AlpacaEval 2	805	GPT-4 Turbo	GPT-4 Turbo	Pairwise comparison	LC & raw win rate
Arena-Hard	500	GPT-4-0314	GPT-4 Turbo	Pairwise comparison	Win rate
MT-Bench	80	-	GPT-4/GPT-4 Turbo	Single-answer grading	Rating of 1-10

Hyperparameters

Setting	β	γ	Learning rate
Mistral-Base	2.0	1.6	3e-7
Mistral-Instruct	2.5	0.3	5e-7
Llama-3-Base	2.0	1.0	6e-7
Llama-3-Instruct	2.5	1.4	1e-6

Method	Objective	Hyperparameter
RRHF [91]	$\max\left(0, -\frac{1}{\lvert y_w\rvert}\log \pi_\theta(y_w\lvert x) + \frac{1}{\lvert y_l\rvert}\log \pi_\theta(y_l\lvert x)\right) - \lambda \log \pi_\theta(y_w\lvert x)$	$\lambda \in [0.1, 0.5, 1.0, 10.0]$
SLiC-HF [96]	$\max\left(0, \delta - \log \pi_\theta(y_w\lvert x) + \log \pi_\theta(y_l\lvert x)\right) - \lambda \log \pi_\theta(y_w\lvert x)$	$\lambda \in [0.1, 0.5, 1.0, 10.0]$ $\beta \in [0.1, 0.5, 1.0, 2.0]$
DPO [66]	$-\log \sigma\left(\beta \log \frac{\pi_\theta(y_w\lvert x)}{\pi_{\text{ref}}(y_w\lvert x)} - \beta \log \frac{\pi_\theta(y_l\lvert x)}{\pi_{\text{ref}}(y_l\lvert x)}\right)$	$\beta \in [0.01, 0.05, 0.1]$
IPO [6]	$\left(\log \frac{\pi_\theta(y_w\lvert x)}{\pi_{\text{ref}}(y_w\lvert x)} - \log \frac{\pi_\theta(y_l\lvert x)}{\pi_{\text{ref}}(y_l\lvert x)} - \frac{1}{2\tau}\right)^2$	$\tau \in [0.01, 0.1, 0.5, 1.0]$
CPO [88]	$-\log \sigma\left(\beta \log \pi_\theta(y_w\lvert x) - \beta \log \pi_\theta(y_l\lvert x)\right) - \lambda \log \pi_\theta(y_w\lvert x)$	$\lambda = 1.0,\ \beta \in [0.01, 0.05, 0.1]$
KTO [29]	$-\lambda_w \sigma\left(\beta \log \frac{\pi_\theta(y_w\lvert x)}{\pi_{\text{ref}}(y_w\lvert x)} - z_{\text{ref}}\right) + \lambda_l \sigma\left(z_{\text{ref}} - \beta \log \frac{\pi_\theta(y_l\lvert x)}{\pi_{\text{ref}}(y_l\lvert x)}\right)$, where $z_{\text{ref}} = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\beta \text{KL}\left(\pi_\theta(y\lvert x)\lvert\lvert \pi_{\text{ref}}(y\lvert x)\right)\right]$	$\lambda_l = \lambda_w = 1.0$ $\beta \in [0.01, 0.05, 0.1]$
ORPO [42]	$-\log p_\theta(y_w\lvert x) - \lambda \log \sigma\left(\log \frac{p_\theta(y_w\lvert x)}{1-p_\theta(y_w\lvert x)} - \log \frac{p_\theta(y_l\lvert x)}{1-p_\theta(y_l\lvert x)}\right)$, where $p_\theta(y\lvert x) = \exp\left(\frac{1}{\lvert y\rvert}\log \pi_\theta(y\lvert x)\right)$	$\lambda \in [0.1, 0.5, 1.0, 2.0]$
R-DPO [64]	$-\log \sigma\left(\beta \log \frac{\pi_\theta(y_w\lvert x)}{\pi_{\text{ref}}(y_w\lvert x)} - \beta \log \frac{\pi_\theta(y_l\lvert x)}{\pi_{\text{ref}}(y_l\lvert x)} + (\alpha\lvert y_w\rvert - \alpha\lvert y_l\rvert)\right)$	$\alpha \in [0.05, 0.1, 0.5, 1.0]$ $\beta \in [0.01, 0.05, 0.1]$
SimPO	$-\log \sigma\left(\frac{\beta}{\lvert y_w\rvert}\log \pi_\theta(y_w\lvert x) - \frac{\beta}{\lvert y_l\rvert}\log \pi_\theta(y_l\lvert x) - \gamma\right)$	$\beta \in [2.0, 2.5]$ $\gamma \in [0.3, 0.5, 1.0, 1.2, 1.4, 1.6]$

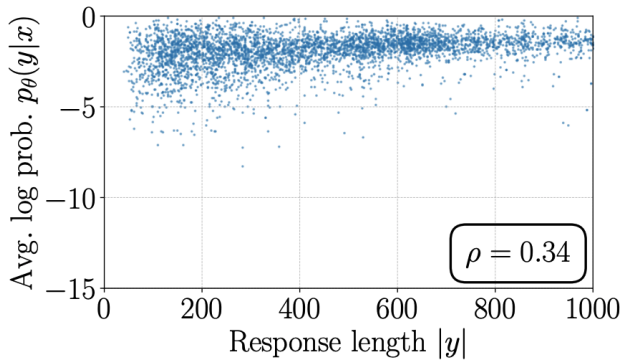
Comparison

- "We find that many variants of DPO do not empirically present an advantage over standard DPO."

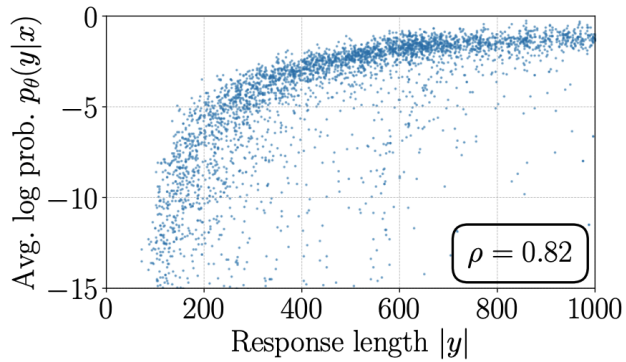
Method	Mistral-Base (7B)					Mistral-Instruct (7B)				
	AlpacaEval 2		Arena-Hard	MT-Bench		AlpacaEval 2		Arena-Hard	MT-Bench	
	LC (%)	WR (%)	WR (%)	GPT-4 Turbo	GPT-4	LC (%)	WR (%)	WR (%)	GPT-4 Turbo	GPT-4
SFT	8.4	6.2	1.3	4.8	6.3	17.1	14.7	12.6	6.2	7.5
RRHF [91]	11.6	10.2	5.8	5.4	6.7	25.3	24.8	18.1	6.5	7.6
SLiC-HF [96]	10.9	8.9	7.3	5.8	**7.4**	24.1	24.6	18.9	6.5	**7.8**
DPO [66]	15.1	12.5	10.4	5.9	7.3	26.8	24.9	16.3	6.3	7.6
IPO [6]	11.8	9.4	7.5	5.5	7.2	20.3	20.3	16.2	6.4	**7.8**
CPO [88]	9.8	8.9	6.9	5.4	6.8	23.8	28.8	**22.6**	6.3	7.5
KTO [29]	13.1	9.1	5.6	5.4	7.0	24.5	23.6	17.9	6.4	7.7
ORPO [42]	14.7	12.2	7.0	5.8	7.3	24.5	24.9	20.8	6.4	7.7
R-DPO [64]	17.4	12.8	8.0	5.9	**7.4**	27.3	24.5	16.1	6.2	7.5
SimPO	**21.5**	**20.8**	**16.6**	**6.0**	7.3	**32.1**	**34.8**	21.0	**6.6**	7.6

SimPO ablations

- Ablation studies using the Mistral-Base setting
- Length normalization $\frac{1}{|y|}$:

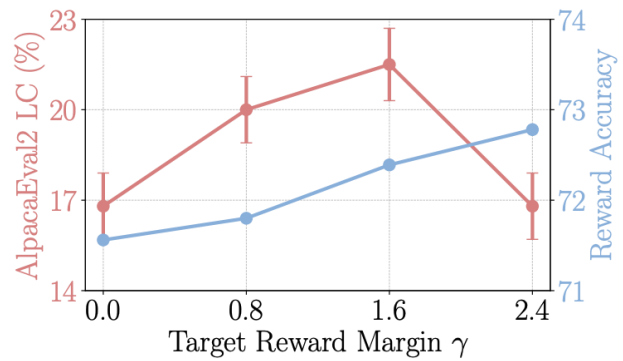


(b) SimPO.

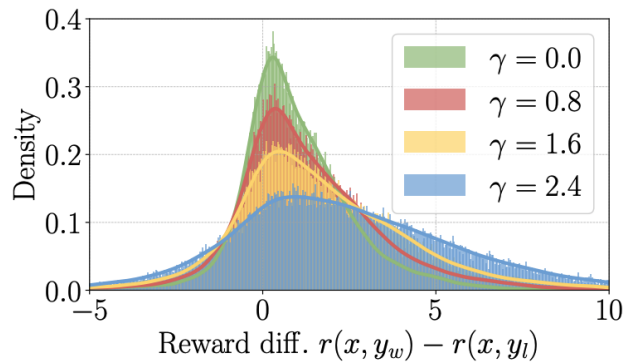


(c) SimPO without LN.

- Reward margin γ:



(a) Performance w/ different γ.



(b) Reward diff. distribution.

References

M. G. Azar, Z. D. Guo, B. Piot, R. Munos, M. Rowland, M. Valko, and D. Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR, 2024.

K. Ethayarajh, W. Xu, N. Muennighoff, D. Jurafsky, and D. Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

J. Hong, N. Lee, and J. Thorne. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*, 2024.

Y. Meng, M. Xia, and D. Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37: 124198–124235, 2024.

L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

A. Tversky and D. Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, 5:297–323, 1992.

H. Xu, Y. J. Kim, A. Sharaf, and H. H. Awadalla. A paradigm shift in machine translation: Boosting translation performance of large language models. *arXiv preprint arXiv:2309.11674*, 2023.

H. Xu, A. Sharaf, Y. Chen, W. Tan, L. Shen, B. Van Durme, K. Murray, and Y. J. Kim. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *arXiv preprint arXiv:2401.08417*, 2024.

Z. Yuan, H. Yuan, C. Tan, W. Wang, S. Huang, and F. Huang. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint*