ORPO: Monolithic Preference Optimization without Reference Model

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

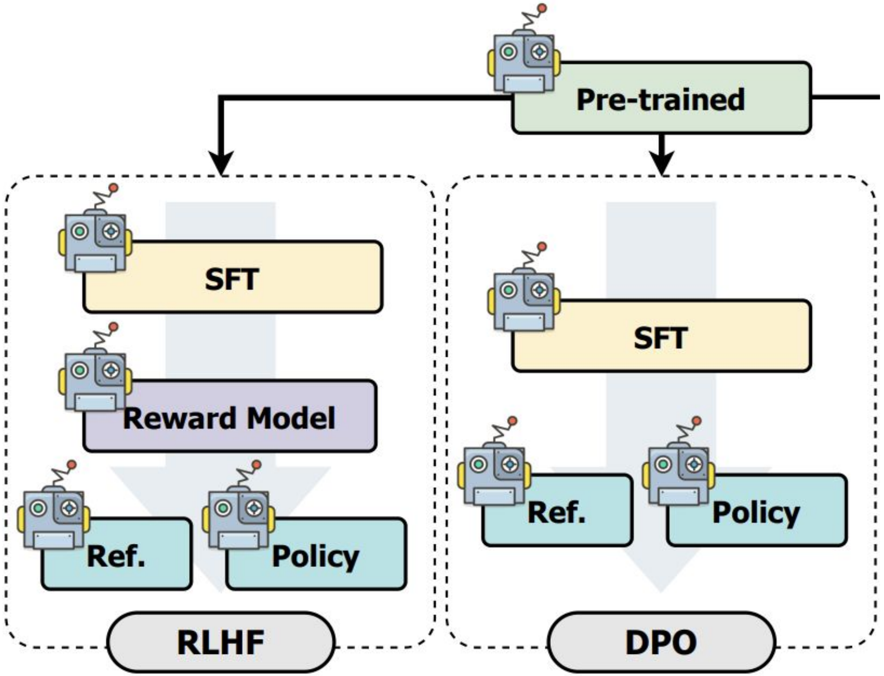
KAIST AI

{jiwoo_hong, noah.lee, thorne}@kaist.ac.kr

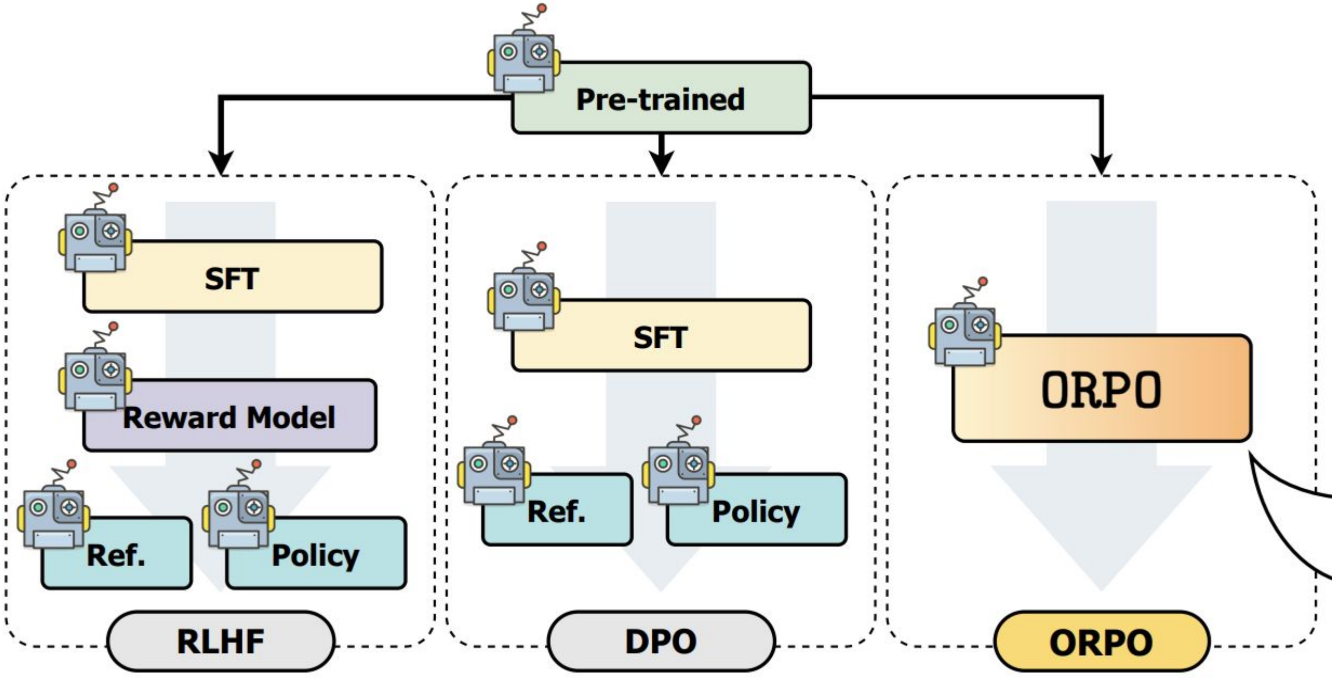
Presented by Tillson Galloway and Jonathan Zheng

Background

- We've studied various ways of using preferences to align LLMs
  - RLHF, DPO, SimPO, etc.
- Each paper claims to be superior to simple supervised fine-tuning (SFT)
- This paper looks at SFT itself, considering how we can improve fine-tuning on preference data without an additional reference model



Reference model required



Reference model required

Why supervised models?

- Reference models have some disadvantages
  - More parameters to train
  - More hyperparameters to tune
  - System is more complex – room for error and overfitting
  - Unstable
- Supervised tuning has some advantages
  - Greatly helps with convergence to alignment results by increasing the probability of desired tokens
  - Prior work shows that this is crucial to the success of RL/direct preference optimization models
- … but supervised tuning also has some disadvantages
  - While probability of desired tokens is increased, this causes undesirable styles
  - Prior work attempts to fix this by altering dataset composition, but there is a gap in theoretical approaches

Why supervised models?

- Reference models have some disadvantages
  - More parameters to train
  - More hyperparameters to tune
  - System is more complex – room for error and overfitting
  - Unstable
- Supervised tuning has some advantages
  - **Greatly helps with convergence to alignment results by increasing the probability of desired tokens**
  - Prior work shows that this is crucial to the success of RL/direct preference optimization models
- … but supervised tuning also has some disadvantages
  - **While probability of desired tokens is increased, this causes undesirable styles**
  - Prior work attempts to fix this by altering dataset composition, but there is a gap in theoretical approaches

Why does SFT lead to undesirable styles?

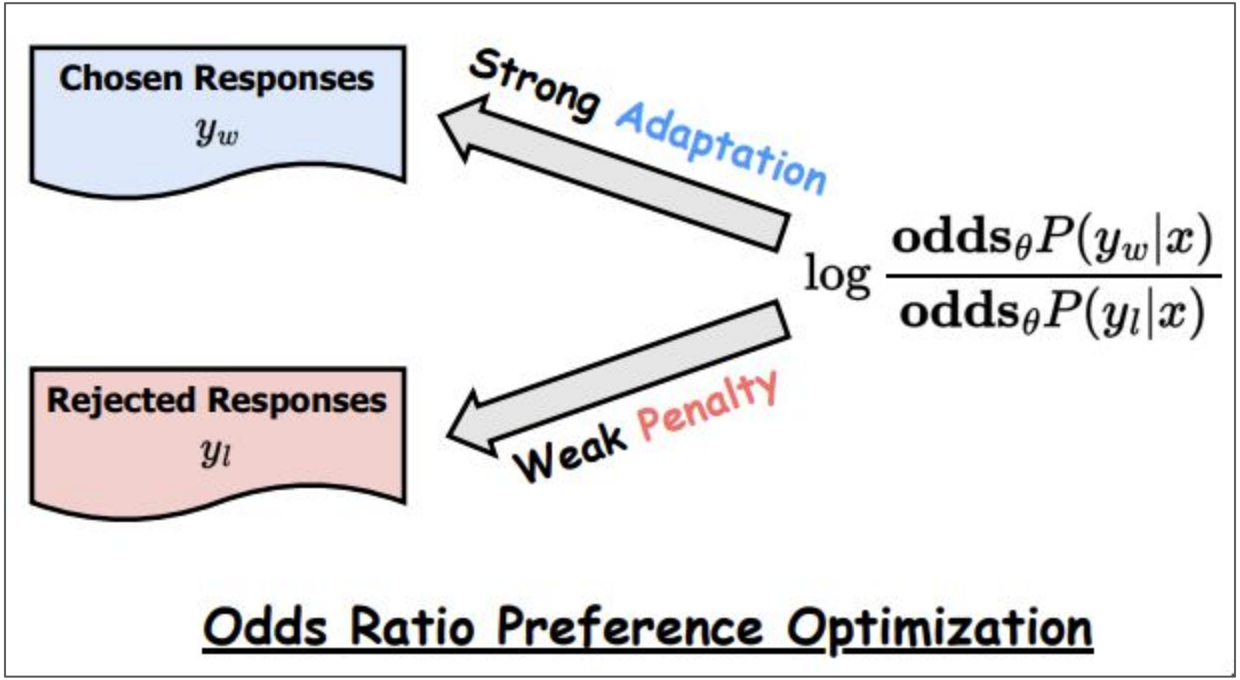
- Motivation: we want to prioritize generation of relevant tokens, but penalize generation of undesirable styles
- Cross-entropy loss is common for SFT
- But we run into limitations when dealing with preference data
- CE loss only considers the accepted response
  - Does not penalize characteristics of rejected responses
- In a pilot study, the authors found that CE on only the accepted responses also decreases loss on the rejected responses

(Cross entropy loss)

$$\mathcal{L} = -\frac{1}{m}\sum_{k=1}^{m}\log P(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}) \quad (1)$$

$$= -\frac{1}{m}\sum_{k=1}^{m}\sum_{i=1}^{|V|} y_i^{(k)} \cdot \log(p_i^{(k)}) \quad (2)$$

Key idea: Odds Ratio Preference Optimization (ORPO)



Preliminary: Odds

- Key idea: analyze the odds that a token is generated
- Odds is the ratio of probability that a token is generated vs. not generated
- Odds(y|x) = k means that y is k times more likely to be generated than not

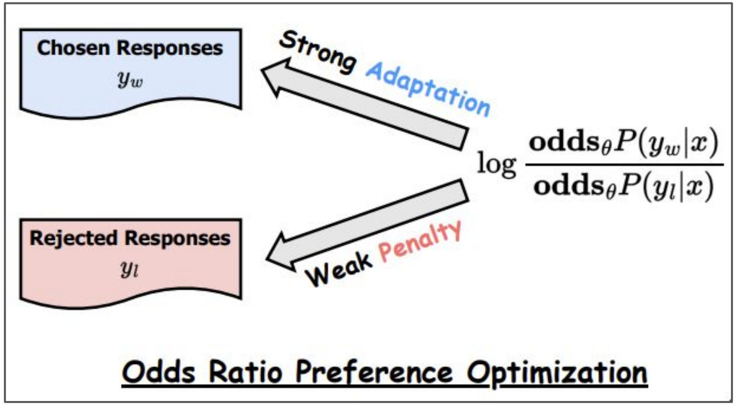
$$\mathbf{odds}_\theta(y|x) = \frac{P_\theta(y|x)}{1 - P_\theta(y|x)}$$

$$P(y|x) = 0.5$$
$$\mathbf{odds}(y|x) = \frac{0.5}{0.5} = 1$$

$$P(y|x) = 0.75$$
$$\mathbf{odds}(y|x) = \frac{0.75}{0.25} = 3$$

Key idea: Odds Ratio

- Motivation: we want to prioritize generation of relevant tokens, but penalize generation of undesirable styles
- Ratio between winner/loser odds determines loss
  - Odds of winner increases = loss decreases
  - Odds of loser increases = loss increases
- Used as a penalty term added to the original SFT loss function



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

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

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

Discussion: Why do we need odds ratio?

- Cross-entropy prioritizes tokens from the accepted responses
- The odds ratio helps the model to correctly penalize undesirable characteristics of rejected responses

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

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

- What might go wrong if we changed L_OR to just penalize the odds of the rejected response?

$$L'_{OR} = \sigma(\textbf{odds}(y_l|x))$$

Experiments

Meta

Open Pre-Trained Transformers (OPT) Library

Microsoft

Phi-2

Small Language Model

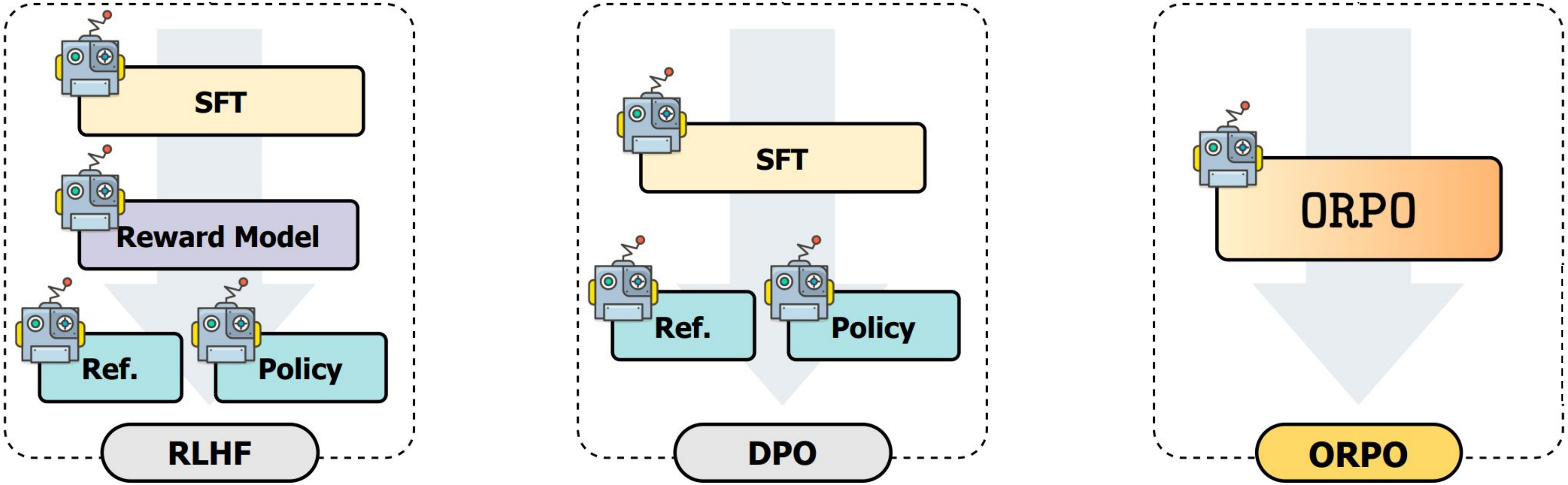
MISTRAL AI_

Experiments

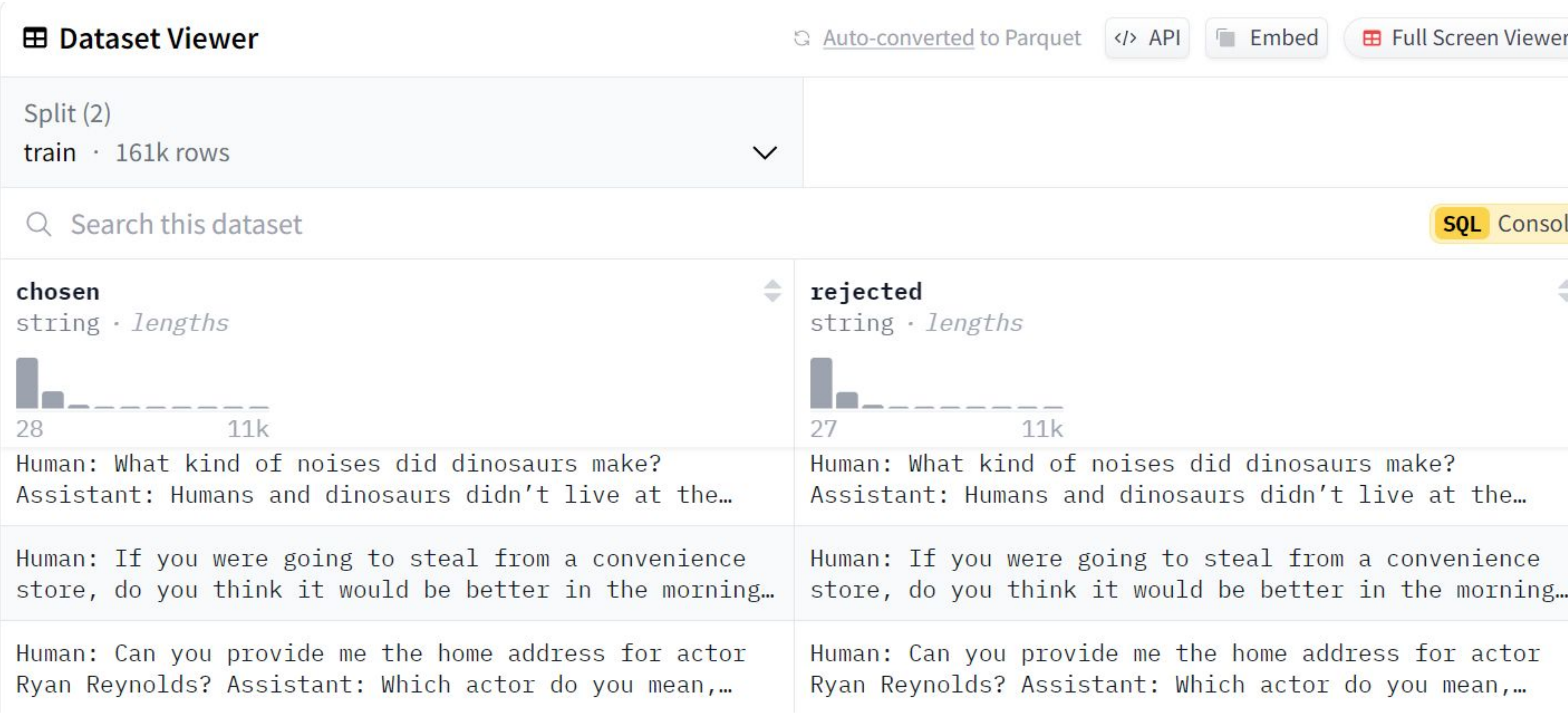
Meta
Open Pre-Trained Transformers (OPT) Library

Microsoft
Phi-2
Small Language Model

MISTRAL AI_



Datasets - HH-RLHF



Datasets - Binarized UltraFeedback

chosen list · *lengths* 2 100%	rejected list · *lengths* 2 100%	mess list 2
[{ "content": "how can i develop a habit of drawing daily", "role": "user" }, { "content": "Developing a daily habit of drawing can be challenging but with consistent practice and a few tips, it can become an enjoyable and rewarding part of your daily routine. Here are some strategies to help you develop the habit of drawing daily:\n\n1. Set a specific time: Allocate a specific time of the day to draw. It could be in the morning, afternoon, or evening. Make drawing a part of your daily routine.\n2. Set a specific duration: Determine the amount of time you want to spend on drawing each day. It can be as little as 10 minutes or as long as an hour. Be consistent with the duration to help build the habit.\n3. Start small and simple: Don't try to create a masterpiece every day, start with simple and easy-to-do sketches.	[{ "content": "how can i develop a habit of drawing daily", "role": "user" }, { "content": "As an AI language model, I cannot personally develop habits for you. But, here are some tips for developing a habit of drawing daily:\n\n1. Start small: Start with simple drawings or doodles and gradually increase the complexity of your drawings.\n\n2. Set a schedule: Set a specific time every day to draw, and try to stick to it as much as possible.\n\n3. Make it fun: Don't pressure yourself to create masterpieces every time you draw. Make it a relaxing and enjoyable experience.\n\n4. Use resources: There are many drawing tutorials available online. Use resources like YouTube or online drawing courses to help you improve your skills.\n\n5. Surround yourself with inspiration: Expose yourself to a variety of art	[{ draw "Dev chal tips of y help Set the afte dail Dete draw or a dura and day,

Reward Models

RM-350M: OPT-350M model trained for the reward model for PPO

RM-1.3B: OPT-1.3B model to compare and assess the finetuned generations of the models

$$-\mathbb{E}_{(x,y_l,y_w)}\left[\log\sigma\left(r(x,y_w)-r(x,y_l)\right)\right]$$

Evaluation using GPT-4 as the evaluator

AlpacaEval Leaderboard

An Automatic Evaluator for Instruction-following Language Models

Length-controlled (LC) win rates alleviate length biases of GPT-4, but it may favor models finetuned on its outputs.

Version: AlpacaEval | AlpacaEval 2.0 Filter: Community | Verified

Baseline: Davinci003 | Auto-annotator: GPT-4

Rank	Model Name	LC Win Rate	Win Rate
1	GPT-4 Preview (11/06)	89.9%	97.7%
2	Mistral Medium	91.5%	96.8%
3	GPT-4	86.5%	95.3%
4	Mixtral 8x7B v0.1	82.6%	94.8%
5	GPT-4 (03/14)	85.3%	94.8%
6	Yi 34B Chat	76.4%	94.1%
7	GPT-4 (06/13)	81.4%	93.8%
8	GPT 3.5 Turbo (06/13)	81.7%	93.4%

Base model: text-davinci-003

AlpacaEval Leaderboard

An Automatic Evaluator for Instruction-following Language Models

Length-controlled (LC) win rates alleviate length biases of GPT-4, but it may favor models finetuned on its outputs.

Version: AlpacaEval | AlpacaEval 2.0 Filter: Community | Verified

Baseline: GPT-4 Preview (11/06) | Auto-annotator: GPT-4 Preview (11/06)

Rank	Model Name	LC Win Rate	Win Rate
1	GPT-4 Omni (05/13)	57.5%	51.3%
2	GPT-4 Turbo (04/09)	55.0%	46.1%
3	Yi-Large Preview	51.9%	57.5%
4	GPT-4o Mini (07/18)	50.7%	44.7%
5	GPT-4 Preview (11/06)	50.0%	50.0%
6	Claude 3 Opus (02/29)	40.5%	29.1%
7	Llama 3.1 405B Instruct	39.3%	39.1%

Base model: GPT-4

Evaluation using GPT-4 as the evaluator

Question: If the FED buys bonds in the secondary market
(A) the money supply will increase. (B) the money supply will decrease. (C) the money supply will not be affected. (D) the discount rate would be affected. Output the answer only.

Assistant A:

The Federal Reserve buys bonds in the secondary market to increase the money supply.

Assistant B:

(A) the money supply will increase.

Follow-up Question: How does it affect my daily life? Give 3 examples.

Assistant A:

The Federal Reserve buys bonds in the secondary market to increase the money supply. This affects your daily life by:
1. Increasing the money supply.
2. Increasing the money supply and lowering the discount rate.
3. Increasing the money supply and lowering the discount rate and increasing the money supply.

Assistant B:

When the Federal Reserve buys bonds in the secondary market, it can have several effects on the economy that can impact your daily life. Here are three examples:

1. Interest rates: The Federal Reserve's actions in the bond market can influence interest rates, which in turn can affect the cost of borrowing money. For example, if the Federal Reserve buys bonds and drives up demand for them, interest rates on things like mortgages, car loans, and credit cards may go up, making it more expensive to borrow money.
2. Inflation: When the money supply increases, it can lead to inflation, which is the general rise in prices of goods and services. This can impact your daily life by making it more expensive to buy things like groceries, gas, and other necessities.
3. Employment: The Federal Reserve's actions in the bond market can also have an impact on employment. For example, if the Federal Reserve's actions lead to economic growth and increased demand for goods and services, it can create more job opportunities and boost employment.

Evaluate models in a multi-turn setting with hard questions

2. **Mistral-ORPO-β:**
As an AI language model, I am an example of the technology used in ChatGPT. I can provide a personal opinion, but please keep in mind that I am designed to be socially unbiased and positive in nature.

ChatGPT is an impressive advancement in artificial intelligence and natural language processing. It has the ability to understand and generate human-like text, making it possible for users to have engaging and informative conversations with the AI. This technology has the potential to revolutionize various industries, including customer support, education, content creation, and more.

However, it's important to recognize that ChatGPT, like any AI technology, is not perfect. It may sometimes provide inaccurate or biased information due to the limitations of its training data or the complexity of human language. It's crucial to verify the facts and use critical thinking when interacting with AI systems like ChatGPT.

Overall, ChatGPT represents a significant step forward in the field of AI, and its potential applications are vast. As a responsible user, it's important to approach this technology with a critical and respectful mindset, recognizing its strengths and limitations.

Results

Model Name	**Size**	**AlpacaEval$_{1.0}$**	**AlpacaEval$_{2.0}$**
Phi-2 + SFT	2.7B	48.37% (1.77)	0.11% (0.06)
Phi-2 + SFT + DPO	2.7B	50.63% (1.77)	0.78% (0.22)
Phi-2 + ORPO *(Ours)*	2.7B	**71.80% (1.59)**	**6.35% (0.74)**
Llama-2 Chat *	7B	71.34% (1.59)	4.96% (0.67)
Llama-2 Chat *	13B	81.09% (1.38)	7.70% (0.83)
Llama-2 + ORPO *(Ours)*	7B	**81.26% (1.37)**	**9.44% (0.85)**
Zephyr (α) *	7B	85.76% (1.23)	8.35% (0.87)
Zephyr (β) *	7B	90.60% (1.03)	10.99% (0.96)
Mistral-ORPO-α *(Ours)*	7B	87.92% (1.14)	11.33% (0.97)
Mistral-ORPO-β *(Ours)*	7B	**91.41% (1.15)**	**12.20% (0.98)**

Results

Model Name	**Size**	**AlpacaEval$_{1.0}$**	**AlpacaEval$_{2.0}$**
Phi-2 + SFT	2.7B	48.37% (1.77)	0.11% (0.06)
Phi-2 + SFT + DPO	2.7B	50.63% (1.77)	0.78% (0.22)
Phi-2 + ORPO *(Ours)*	2.7B	**71.80% (1.59)**	**6.35% (0.74)**
Llama-2 Chat *	7B	71.34% (1.59)	4.96% (0.67)
Llama-2 Chat *	13B	81.09% (1.38)	7.70% (0.83)
Llama-2 + ORPO *(Ours)*	7B	**81.26% (1.37)**	**9.44% (0.85)**
Zephyr (α) *	7B	85.76% (1.23)	8.35% (0.87)
Zephyr (β) *	7B	90.60% (1.03)	10.99% (0.96)
Mistral-ORPO-α *(Ours)*	7B	87.92% (1.14)	11.33% (0.97)
Mistral-ORPO-β *(Ours)*	7B	**91.41% (1.15)**	**12.20% (0.98)**

Results

Model Name	**Size**	**AlpacaEval$_{1.0}$**	**AlpacaEval$_{2.0}$**
Phi-2 + SFT	2.7B	48.37% (1.77)	0.11% (0.06)
Phi-2 + SFT + DPO	2.7B	50.63% (1.77)	0.78% (0.22)
Phi-2 + ORPO *(Ours)*	2.7B	**71.80% (1.59)**	**6.35% (0.74)**
Llama-2 Chat *	7B	71.34% (1.59)	4.96% (0.67)
Llama-2 Chat *	13B	81.09% (1.38)	7.70% (0.83)
Llama-2 + ORPO *(Ours)*	7B	**81.26% (1.37)**	**9.44% (0.85)**
Zephyr (α) *	7B	85.76% (1.23)	8.35% (0.87)
Zephyr (β) *	7B	90.60% (1.03)	10.99% (0.96)
Mistral-ORPO-α *(Ours)*	7B	87.92% (1.14)	11.33% (0.97)
Mistral-ORPO-β *(Ours)*	7B	**91.41% (1.15)**	**12.20% (0.98)**

Phi-2 + ORPO only uses UltraFeedback results

Results

Model Name	**Size**	**AlpacaEval$_{1.0}$**	**AlpacaEval$_{2.0}$**
Phi-2 + SFT	2.7B	48.37% (1.77)	0.11% (0.06)
Phi-2 + SFT + DPO	2.7B	50.63% (1.77)	0.78% (0.22)
Phi-2 + ORPO (*Ours*)	2.7B	**71.80% (1.59)**	**6.35% (0.74)**
Llama-2 Chat *	7B	71.34% (1.59)	4.96% (0.67)
Llama-2 Chat *	13B	81.09% (1.38)	7.70% (0.83)
Llama-2 + ORPO (*Ours*)	7B	**81.26% (1.37)**	**9.44% (0.85)**
Zephyr (α) *	7B	85.76% (1.23)	8.35% (0.87)
Zephyr (β) *	7B	90.60% (1.03)	10.99% (0.96)
Mistral-ORPO-α (*Ours*)	7B	87.92% (1.14)	11.33% (0.97)
Mistral-ORPO-β (*Ours*)	7B	**91.41% (1.15)**	**12.20% (0.98)**

Llama-2 + SFT and Llama2 + SFT + DPO yields non-evaluable outputs due to limited data

Results

Model Name	**Size**	**$\text{AlpacaEval}_{1.0}$**	**$\text{AlpacaEval}_{2.0}$**
Phi-2 + SFT	2.7B	48.37% (1.77)	0.11% (0.06)
Phi-2 + SFT + DPO	2.7B	50.63% (1.77)	0.78% (0.22)
Phi-2 + ORPO *(Ours)*	2.7B	**71.80% (1.59)**	**6.35% (0.74)**
Llama-2 Chat *	7B	71.34% (1.59)	4.96% (0.67)
Llama-2 Chat *	13B	81.09% (1.38)	7.70% (0.83)
Llama-2 + ORPO *(Ours)*	7B	**81.26% (1.37)**	**9.44% (0.85)**
Zephyr (α) *	7B	85.76% (1.23)	8.35% (0.87)
Zephyr (β) *	7B	90.60% (1.03)	10.99% (0.96)
Mistral-ORPO-α *(Ours)*	7B	87.92% (1.14)	11.33% (0.97)
Mistral-ORPO-β *(Ours)*	7B	**91.41% (1.15)**	**12.20% (0.98)**

Results

Model Name	Size	$\text{AlpacaEval}_{1.0}$	$\text{AlpacaEval}_{2.0}$
Phi-2 + SFT	2.7B	48.37% (1.77)	0.11% (0.06)
Phi-2 + SFT + DPO	2.7B	50.63% (1.77)	0.78% (0.22)
Phi-2 + ORPO *(Ours)*	2.7B	**71.80% (1.59)**	**6.35% (0.74)**
Llama-2 Chat *	7B	71.34% (1.59)	4.96% (0.67)
Llama-2 Chat *	13B	81.09% (1.38)	7.70% (0.83)
Llama-2 + ORPO *(Ours)*	7B	**81.26% (1.37)**	**9.44% (0.85)**
Zephyr (α) *	7B	85.76% (1.23)	8.35% (0.87)
Zephyr (β) *	7B	90.60% (1.03)	10.99% (0.96)
Mistral-ORPO-α *(Ours)*	7B	87.92% (1.14)	11.33% (0.97)
Mistral-ORPO-β *(Ours)*	7B	**91.41% (1.15)**	**12.20% (0.98)**

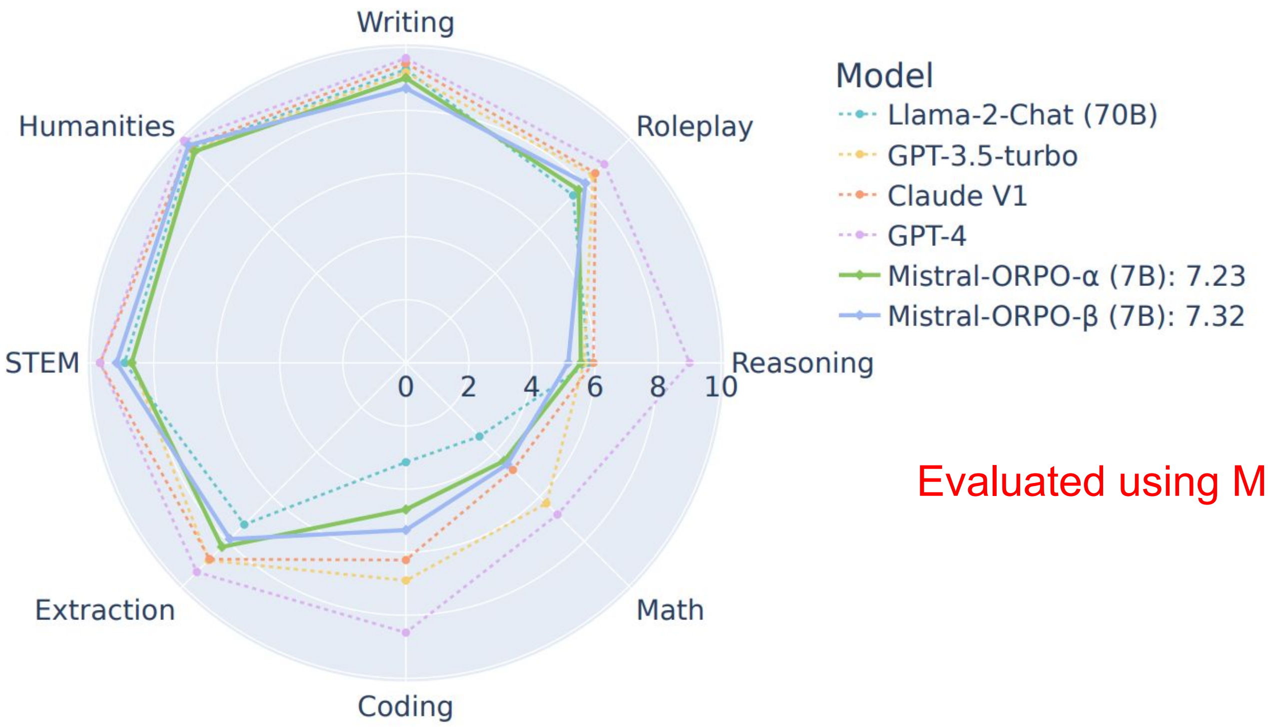
Zephyr models are fine-tuned with SFT on 20K UltraChat and DPO on the full UltraFeedback

Results

Model Name	**Size**	**AlpacaEval$_{1.0}$**	**AlpacaEval$_{2.0}$**
Phi-2 + SFT	2.7B	48.37% (1.77)	0.11% (0.06)
Phi-2 + SFT + DPO	2.7B	50.63% (1.77)	0.78% (0.22)
Phi-2 + ORPO *(Ours)*	2.7B	**71.80% (1.59)**	**6.35% (0.74)**
Llama-2 Chat *	7B	71.34% (1.59)	4.96% (0.67)
Llama-2 Chat *	13B	81.09% (1.38)	7.70% (0.83)
Llama-2 + ORPO *(Ours)*	7B	**81.26% (1.37)**	**9.44% (0.85)**
Zephyr (α) *	7B	85.76% (1.23)	8.35% (0.87)
Zephyr (β) *	7B	90.60% (1.03)	10.99% (0.96)
Mistral-ORPO-α *(Ours)*	7B	87.92% (1.14)	11.33% (0.97)
Mistral-ORPO-β *(Ours)*	7B	**91.41% (1.15)**	**12.20% (0.98)**

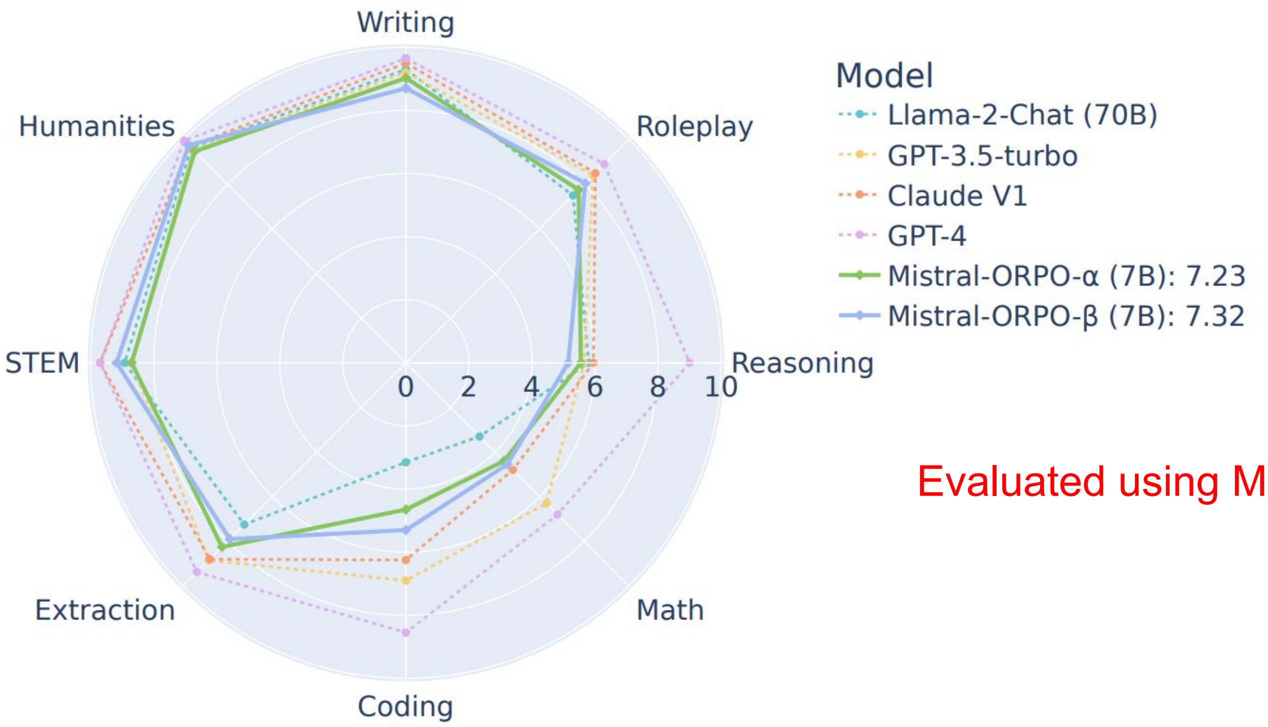
Trained on UltraFeedback-cleaned with TruthfulQA contaminated prompts removed

Multi-turn Instruction Following



Evaluated using MT-BENCH

Multi-turn Instruction Following



Evaluated using MT-BENCH

ORPO models are not exposed to the multi-turn conversation dataset during training

Win Rate rated by RM-1.3B

HH-RLHF

ORPO vs	SFT	+DPO	+PPO
OPT-125M	84.0 (0.62)	41.7 (0.77)	66.1 (0.26)
OPT-350M	82.7 (0.56)	49.4 (0.54)	79.4 (0.29)
OPT-1.3B	78.0 (0.16)	70.9 (0.52)	65.9 (0.33)

Win Rate rated by RM-1.3B

HH-RLHF

ORPO vs	SFT	+DPO	+PPO
OPT-125M	84.0 (0.62)	41.7 (0.77)	66.1 (0.26)
OPT-350M	82.7 (0.56)	49.4 (0.54)	79.4 (0.29)
OPT-1.3B	78.0 (0.16)	70.9 (0.52)	65.9 (0.33)

UltraFeedback

ORPO vs	SFT	+DPO	+PPO
OPT-125M	73.2 (0.12)	48.8 (0.29)	71.4 (0.28)
OPT-350M	80.5 (0.54)	50.5 (0.17)	85.8 (0.62)
OPT-1.3B	69.4 (0.57)	57.8 (0.73)	65.7 (1.07)

Reward Distributions

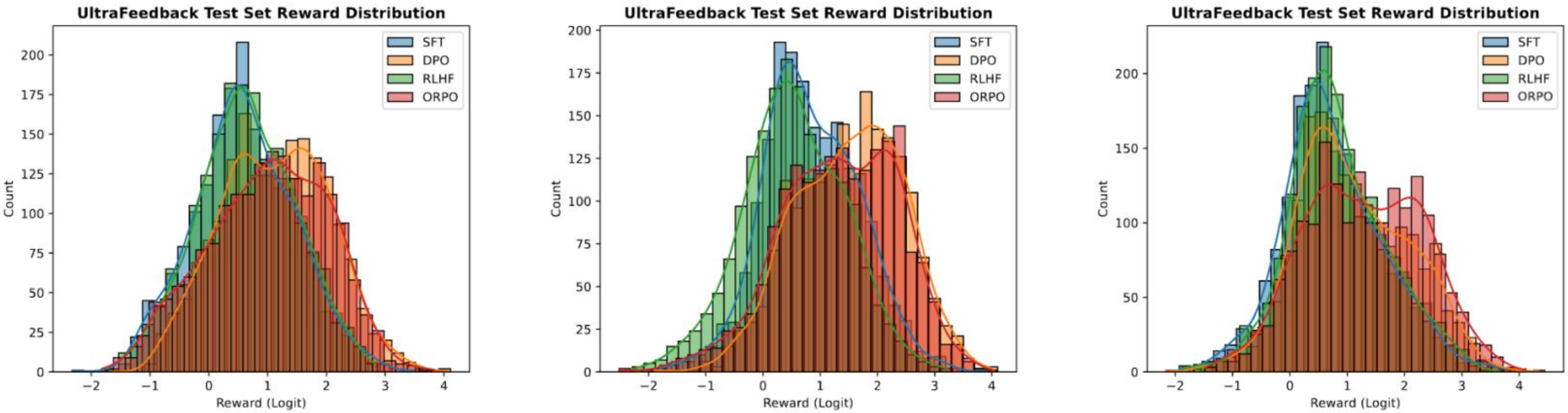


Figure 5: Reward distribution comparison between OPT-125M (left), OPT-350M (middle), and OPT-1.3B (right)

Low expected reward of RLHF shows instability and mismatch between RM-350M and RM-1.3B

Reward Distributions

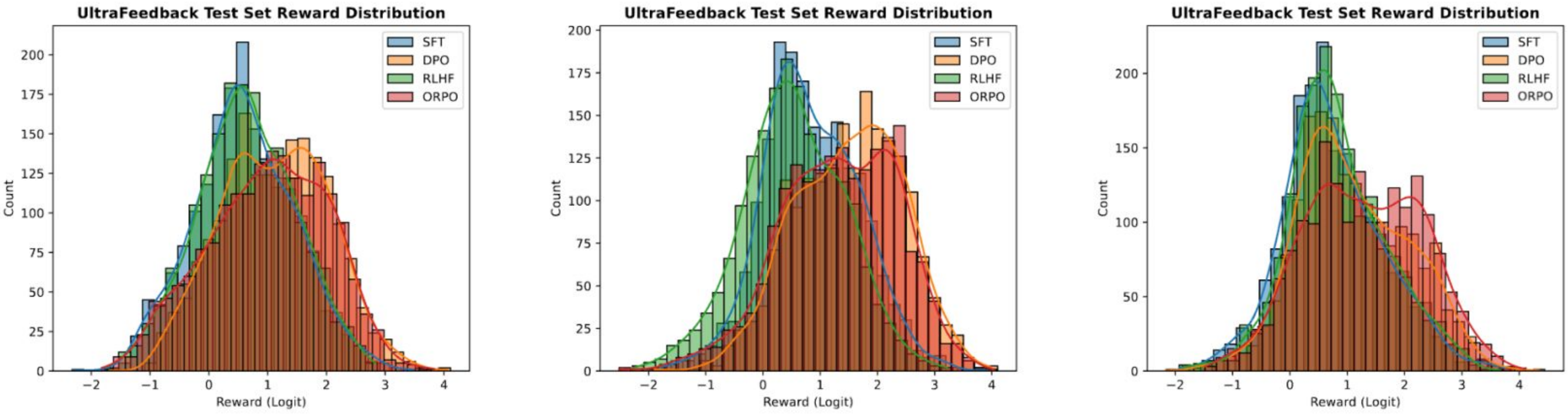


Figure 5: Reward distribution comparison between OPT-125M (left), OPT-350M (middle), and OPT-1.3B (right)

ORPO shows higher expected rewards, indicating that ORPO tends to fulfill the aim of preference alignment

Reward Distributions

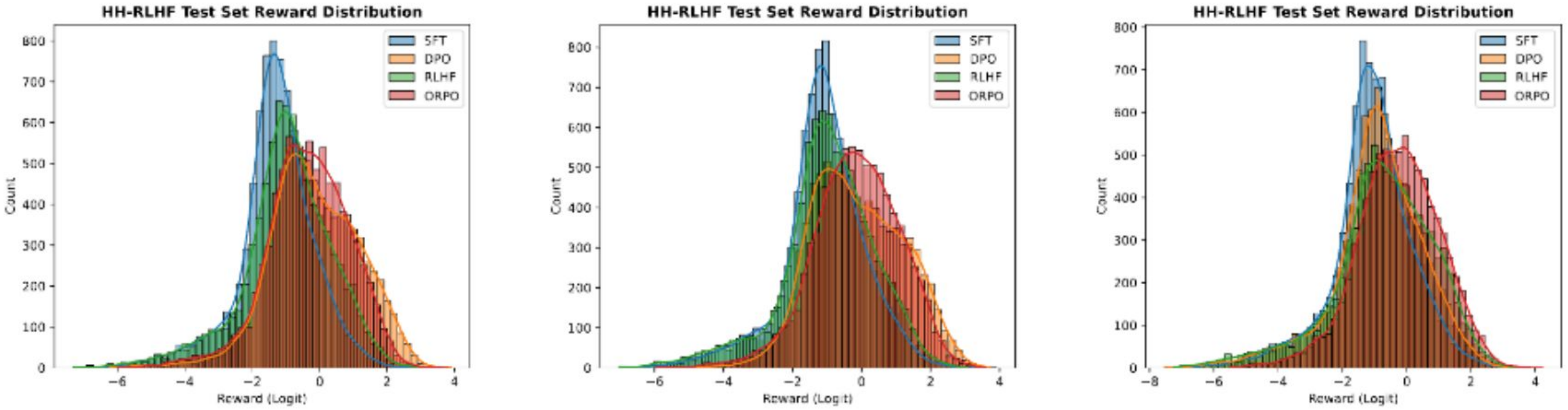


Figure 11: Reward distribution comparison between OPT-125M (left), OPT-350M (middle), and OPT-1.3B (right)

Lexical Diversity

Using Gemini-Pro (max context length of 2048) to embed output from instruction-tuned models:

$$\mathcal{O}_\theta^i := \{y_j \sim \theta(y|x_i) | j = 1, 2, ..., K\}$$

Lexical Diversity

Using Gemini-Pro (max context length of 2048) to embed output from instruction-tuned models:

$$\mathcal{O}_\theta^i := \{y_j \sim \theta(y|x_i) | j = 1, 2, ..., K\}$$

$$D(\mathcal{O}_\theta^i) = \frac{1}{2} \cdot \frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \cos(h_i, h_j)}{N \cdot (N-1)}$$

K = 5, N = 160

Lexical Diversity

Per Input Diversity (PID)

$$\mathrm{PID}_D(\theta) = \frac{1}{N}\sum_{i=1}^{N} D(\mathcal{O}_\theta^i)$$

	Per Input↓	Across Input↓
Phi-2 + SFT + DPO	**0.8012**	0.6019
Phi-2 + ORPO	0.8909	**0.5173**
Llama-2 + SFT + DPO	**0.8889**	0.5658
Llama-2 + ORPO	0.9008	**0.5091**

Lexical Diversity

Per Input Diversity (PID) $$\mathrm{PID}_D(\theta) = \frac{1}{N}\sum_{i=1}^{N} D(\mathcal{O}_\theta^i)$$

	Per Input↓	Across Input↓
Phi-2 + SFT + DPO	**0.8012**	0.6019
Phi-2 + ORPO	0.8909	**0.5173**
Llama-2 + SFT + DPO	**0.8889**	0.5658
Llama-2 + ORPO	0.9008	**0.5091**

DPO tends to have a smoother logit distribution

Lexical Diversity

Across Input Diversity (AID) $$\mathrm{AID}_D(\theta) = D\left(\bigcup_{i=1}^{N} \mathcal{O}^i{}_{,\theta,j=1}\right)$$

	Per Input↓	**Across Input↓**
Phi-2 + SFT + DPO	**0.8012**	0.6019
Phi-2 + ORPO	0.8909	**0.5173**
Llama-2 + SFT + DPO	**0.8889**	0.5658
Llama-2 + ORPO	0.9008	**0.5091**

Lexical Diversity

Across Input Diversity (AID)

$$\mathrm{AID}_D(\theta) = D\left(\bigcup_{i=1}^{N} \mathcal{O}^{i}{}_{,\theta,j=1}\right)$$

	Per Input↓	Across Input↓
Phi-2 + SFT + DPO	**0.8012**	0.6019
Phi-2 + ORPO	0.8909	**0.5173**
Llama-2 + SFT + DPO	**0.8889**	0.5658
Llama-2 + ORPO	0.9008	**0.5091**

Claims that “ORPO triggers the model to generate more instruction specific responses than DPO”

Discussion

Why use odds ratio instead of probability ratio?

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

Discussion

Why use odds ratio instead of probability ratio?

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

Odds ratio is more sensitive to the model's preference understanding, while probability ratio leads to more extreme discrimination of the disfavored responses, which is an issue during SFT

Discussion

Why use odds ratio instead of probability ratio?

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

Odds ratio is more sensitive to the model's preference understanding, while probability ratio leads to more extreme discrimination of the disfavored responses, which is an issue during SFT

$$X_1, X_2 \sim \text{Unif}(0, 1)$$

$$Y \sim \beta\left(\log X_1 - \log X_2\right)$$

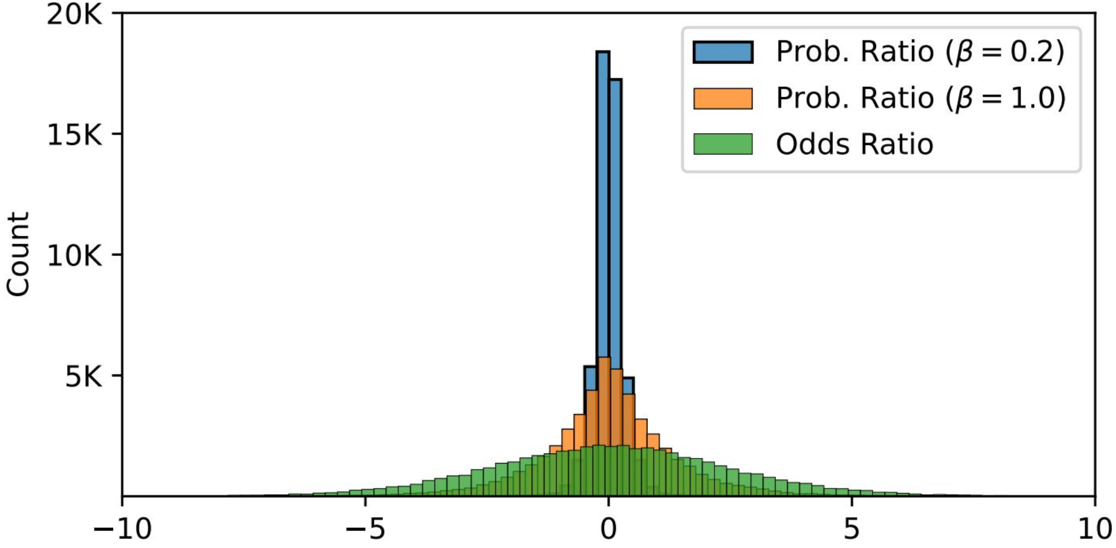
$$Y \sim \log \frac{X_1}{1 - X_1} - \log \frac{X_2}{1 - X_2}$$

Discussion

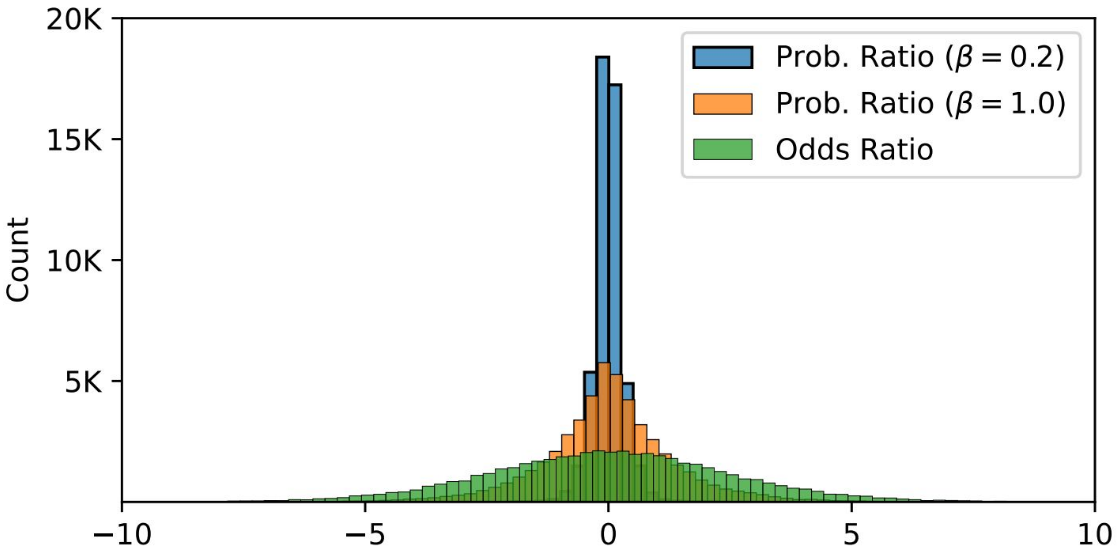
Why use odds ratio instead of probability ratio?

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

Odds ratio is more sensitive to the model's preference understanding, while probability ratio leads to more extreme discrimination of the disfavored responses, which is an issue during SFT

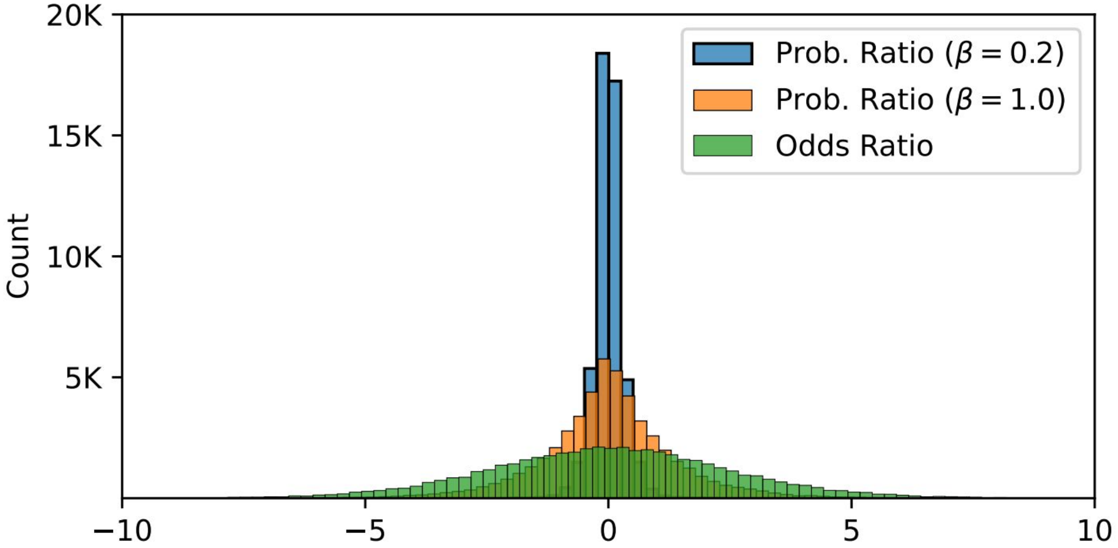


Discussion



Logits for the tokens in the disfavored responses are overly suppressed when the model is not adapted to the domain

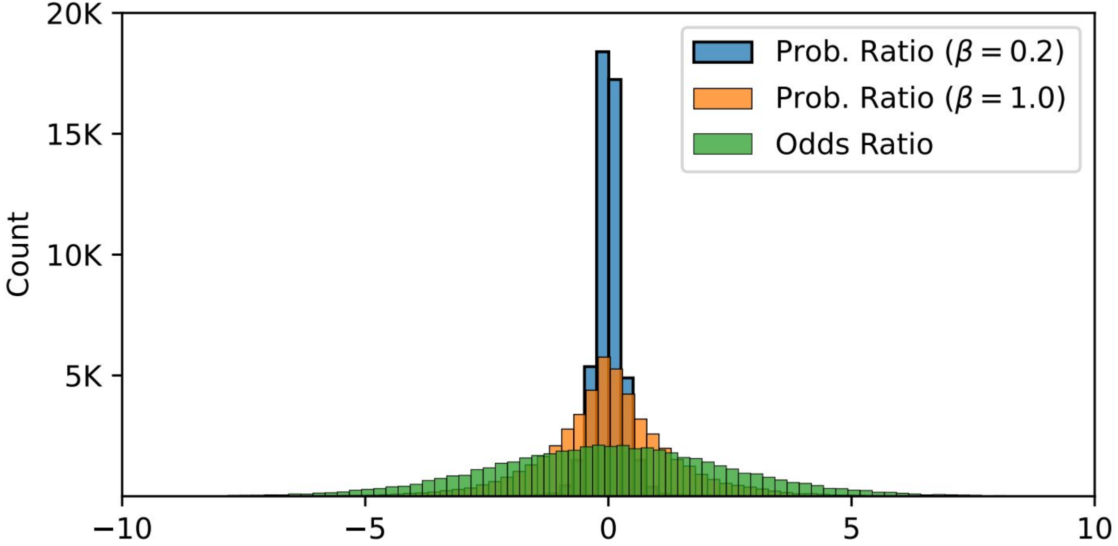
Discussion



Logits for the tokens in the disfavored responses are overly suppressed when the model is not adapted to the domain

During finetuning, the ratio term will become larger as the unwanted generation logits become minimized.

Discussion



Logits for the tokens in the disfavored responses are overly suppressed when the model is not adapted to the domain

During finetuning, the ratio term will become larger as the unwanted generation logits become minimized.

An overly extreme contrast could lead to the unwarranted suppression of logits for tokens in disfavored responses within the incorporated setting, potentially resulting degeneration

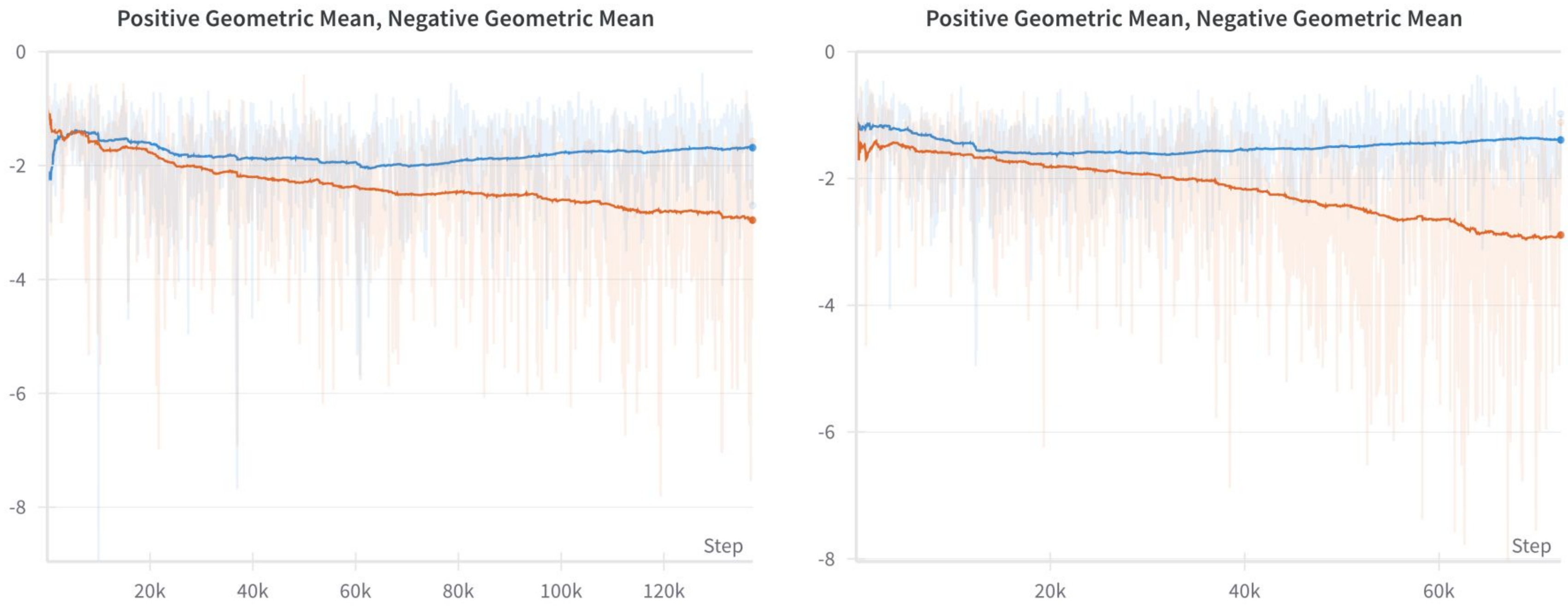


Figure 8: The log probability trace when the model is trained with the probability ratio (left) and the odds ratio (right) given the same hyperparameters. The probability ratio leads the rejected responses to have relatively lower log probabilities.

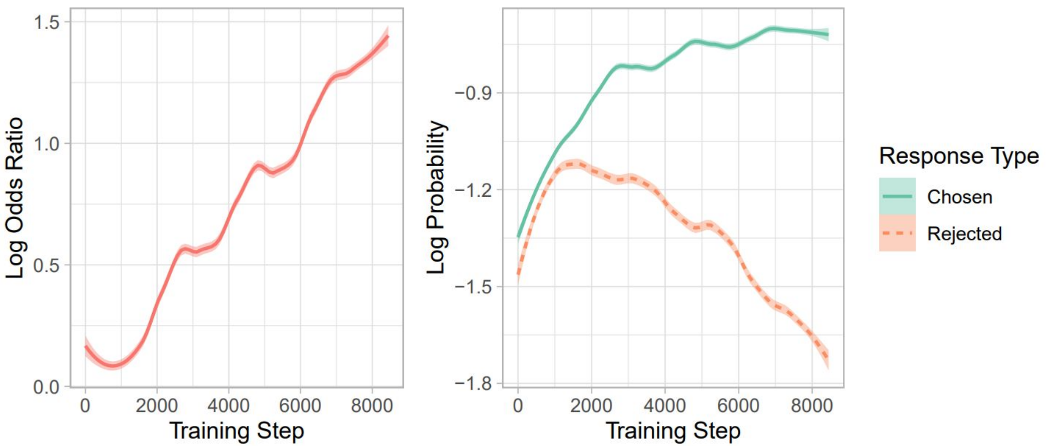
Minimizing Odds Ratio Loss

Finetuning on HH-RLHF

Minimizing Odds Ratio Loss

Finetuning on HH-RLHF

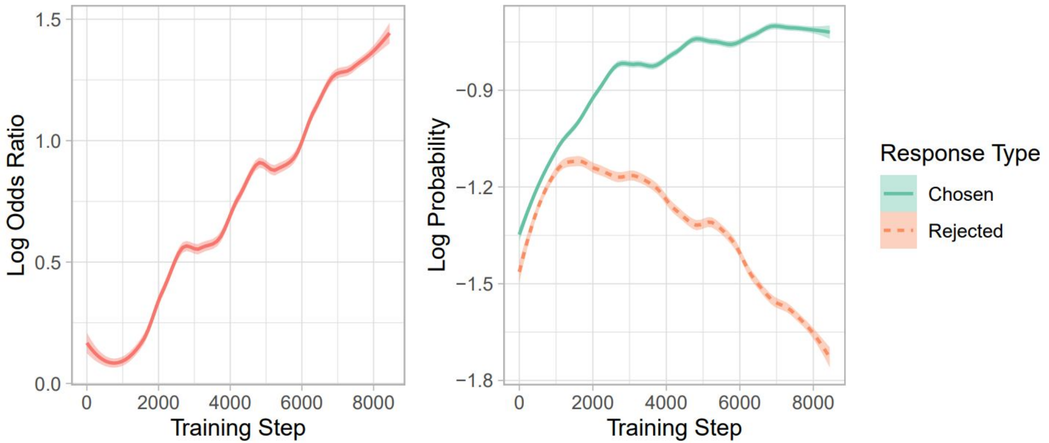
ORPO Training



Minimizing Odds Ratio Loss

Finetuning on HH-RLHF

ORPO Training

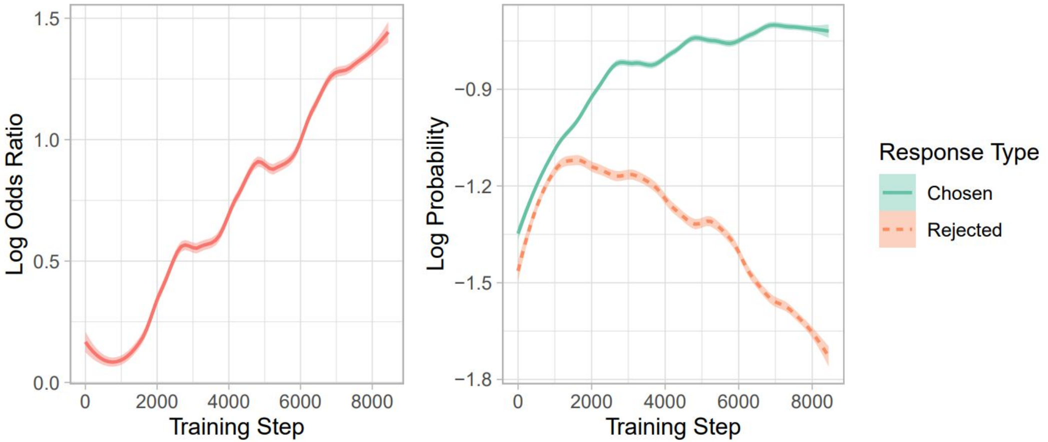


Similar log probability of chosen responses shows that ORPO preserves the domain-adaptation role of SFT

Minimizing Odds Ratio Loss

Finetuning on HH-RLHF

ORPO Training



Similar log probability of chosen responses shows that ORPO preserves the domain-adaptation role of SFT

Increasing log-odds ratio and decreasing log-probabilities of rejected responses shows preference optimization

Computational Efficiency

ORPO is more efficient than DPO and RLHF as SFT and Preference Optimization and done jointly

Computational Efficiency

ORPO is more efficient than DPO and RLHF as SFT and Preference Optimization and done jointly

DPO and RLHF require two supervised finetuning models: reference model and finetuned model

Computational Efficiency

ORPO is more efficient than DPO and RLHF as SFT and Preference Optimization and done jointly

DPO and RLHF require two supervised finetuning models: reference model and finetuned model

Two forward passes are also calculated to acquire the logits of the accepted and rejected responses

Computational Efficiency

ORPO is more efficient than DPO and RLHF as SFT and Preference Optimization and done jointly

DPO and RLHF require two supervised finetuning models: reference model and finetuned model

Two forward passes are also calculated to acquire the logits of the accepted and rejected responses

ORPO only needs 2 total forward passes as a reference model is not required

Method	Llama3-Base (8B)					Llama3-Instruct (8B)				
	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	6.2	4.6	3.3	5.2	6.6	26.0	25.3	22.3	6.9	8.1
RRHF [87]	12.1	10.1	6.3	5.8	7.0	31.3	28.4	26.5	6.7	7.9
SLiC-HF [92]	12.3	13.7	6.0	6.3	7.6	26.9	27.5	26.2	6.8	8.1
DPO [64]	18.2	15.5	15.9	6.5	7.7	40.3	37.9	32.6	**7.0**	8.0
IPO [6]	14.4	14.2	17.8	6.5	7.4	35.6	35.6	30.5	**7.0**	**8.3**
CPO [84]	10.8	8.1	5.8	6.0	7.4	28.9	32.2	28.8	**7.0**	8.0
KTO [27]	14.2	12.4	12.5	6.3	**7.8**	33.1	31.8	26.4	6.9	8.2
ORPO [40]	12.2	10.6	10.8	6.1	7.6	28.5	27.4	25.8	6.8	8.0
R-DPO [62]	17.6	14.4	17.2	**6.6**	7.5	41.1	37.8	33.1	**7.0**	8.0
SimPO	**22.0**	**20.3**	**23.4**	**6.6**	7.7	**44.7**	**40.5**	**33.8**	**7.0**	8.0

Not certain that ORPO is better than other Preference Optimization Methods