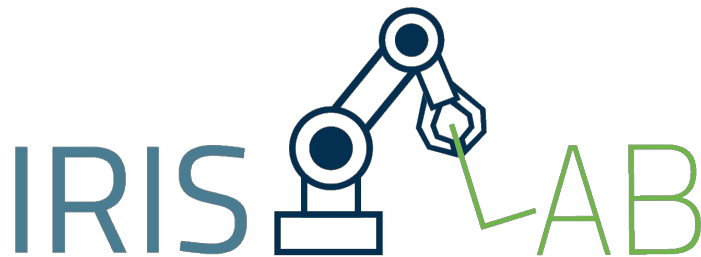


Direct Preference Optimization: A New RLHF Approach

Rafael Rafailov Archit Sharma Eric Mitchell



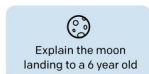
RLHF: Reinforcement Learning From Human Feedback

RLHF: Reinforcement Learning From Human Feedback

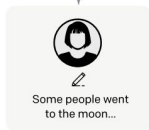
Step 1

**Collect demonstration data,
and train a supervised policy.**

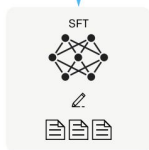
A prompt is
sampled from our
prompt dataset.



A labeler
demonstrates the
desired output
behavior.



This data is used
to fine-tune GPT-3
with supervised
learning.



Training language models to follow instructions with human feedback, Ouyang et. al. 2022

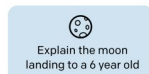
Stanford University

RLHF: Reinforcement Learning From Human Feedback

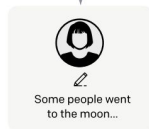
Step 1

**Collect demonstration data,
and train a supervised policy.**

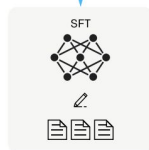
A prompt is
sampled from our
prompt dataset.



A labeler
demonstrates the
desired output
behavior.



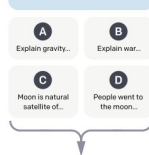
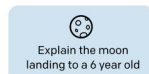
This data is used
to fine-tune GPT-3
with supervised
learning.



Step 2

**Collect comparison data,
and train a reward model.**

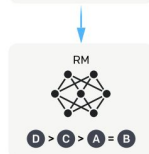
A prompt and
several model
outputs are
sampled.



A labeler ranks
the outputs from
best to worst.



This data is used
to train our
reward model.



Training language models to follow instructions with human feedback, Ouyang et. al. 2022

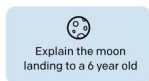
Stanford University

RLHF: Reinforcement Learning From Human Feedback

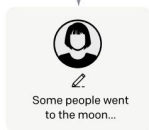
Step 1

Collect demonstration data, and train a supervised policy.

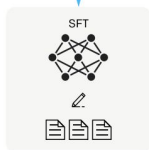
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



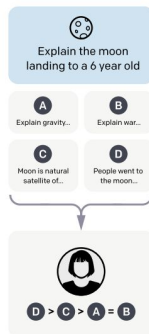
This data is used to fine-tune GPT-3 with supervised learning.



Step 2

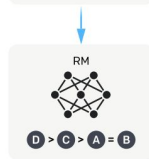
Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



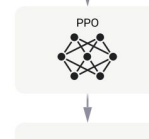
Step 3

Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.



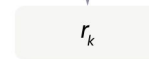
The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.



Training language models to follow instructions with human feedback, Ouyang et. al. 2022

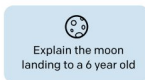
Stanford University

RLHF: Learning a reward model from human feedback

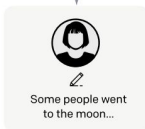
Step 1

Collect demonstration data, and train a supervised policy.

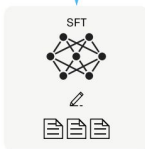
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



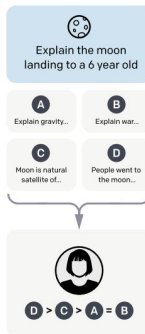
This data is used to fine-tune GPT-3 with supervised learning.



Step 2

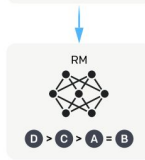
Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



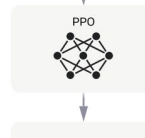
Step 3

Optimize a policy against the reward model using reinforcement learning.

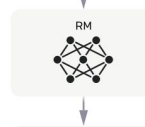
A new prompt is sampled from the dataset.



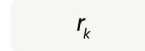
The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.




RLHF: Learning a **reward model** from human feedback

RLHF: Learning a **reward model** from human feedback

Feedback comes as **preferences over model samples**: $\mathcal{D} = \{x^i, y_w^i, y_l^i\}$

RLHF: Learning a **reward model** from human feedback

Feedback comes as **preferences over model samples**: $\mathcal{D} = \{x^i, y_w^i, y_l^i\}$



The diagram illustrates the components of the dataset \mathcal{D} . It shows the equation $\mathcal{D} = \{x^i, y_w^i, y_l^i\}$ with three arrows pointing from labels below to the variables in the set. The label 'Prompt' points to x^i , 'Preferred response' points to y_w^i , and 'Dispreferred response' points to y_l^i .

RLHF: Learning a **reward model** from human feedback

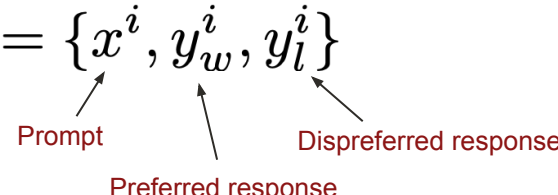
Feedback comes as **preferences over model samples**: $\mathcal{D} = \{x^i, y_w^i, y_l^i\}$

Prompt Preferred response Dispreferred response

Bradley-Terry Model connects rewards to preferences:

RLHF: Learning a **reward model** from human feedback

Feedback comes as **preferences over model samples**: $\mathcal{D} = \{x^i, y_w^i, y_l^i\}$



Prompt Preferred response Dispreferred response

Bradley-Terry Model connects rewards to preferences:

$$p(y_w \succ y_l \mid x) = \sigma(r(x, y_w) - r(x, y_l))$$

RLHF: Learning a **reward model** from human feedback

Feedback comes as **preferences over model samples**: $\mathcal{D} = \{x^i, y_w^i, y_l^i\}$

Prompt Preferred response Dispreferred response

Bradley-Terry Model connects rewards to preferences:

Reward assigned to **preferred** and **dispreferred** responses

$$p(y_w \succ y_l \mid x) = \sigma(r(x, y_w) - r(x, y_l))$$

RLHF: Learning a **reward model** from human feedback

Feedback comes as **preferences over model samples**: $\mathcal{D} = \{x^i, y_w^i, y_l^i\}$

Prompt Preferred response Dispreferred response

Bradley-Terry Model connects rewards to preferences:

Reward assigned to **preferred** and **dispreferred** responses

$$p(y_w \succ y_l \mid x) = \sigma(r(x, y_w) - r(x, y_l))$$

Train the reward model by **minimizing negative log likelihood**:

RLHF: Learning a **reward model** from human feedback

Feedback comes as **preferences over model samples**: $\mathcal{D} = \{x^i, y_w^i, y_l^i\}$

Prompt Preferred response Dispreferred response

Bradley-Terry Model connects rewards to preferences:

Reward assigned to **preferred** and **dispreferred** responses

$$p(y_w \succ y_l \mid x) = \sigma(r(x, y_w) - r(x, y_l))$$

Train the reward model by **minimizing negative log likelihood**:

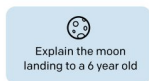
$$\mathcal{L}_R(\phi, \mathcal{D}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} [\log \sigma(r_\phi(x, y_w) - r_\phi(x, y_l))]$$

RLHF: Reinforcement Learning From Human Feedback

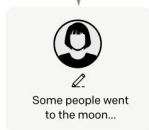
Step 1

Collect demonstration data, and train a supervised policy.

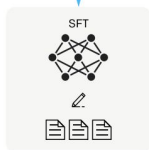
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



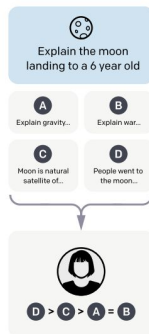
This data is used to fine-tune GPT-3 with supervised learning.



Step 2

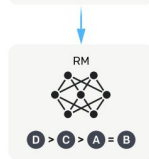
Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



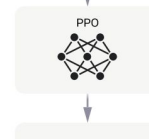
Step 3

Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.



The policy generates an output.

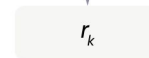


Once upon a time...

The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.



Training language models to follow instructions with human feedback, Ouyang et. al. 2022

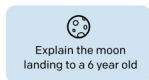
Stanford University

RLHF: Learning a **policy** that optimizes the **reward**

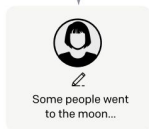
Step 1

Collect demonstration data, and train a supervised policy.

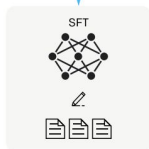
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



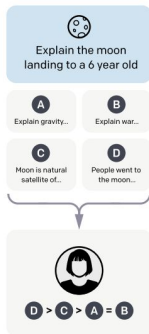
This data is used to fine-tune GPT-3 with supervised learning.



Step 2

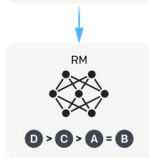
Collect comparison data, and train a reward model.

A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.

This data is used to train our reward model.



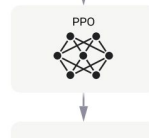
Step 3

Optimize a policy against the reward model using reinforcement learning.

A new prompt is sampled from the dataset.

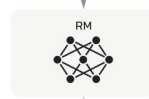


The policy generates an output.

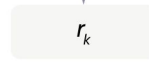


Once upon a time...

The reward model calculates a reward for the output.



The reward is used to update the policy



RLHF: Learning a **policy** that optimizes the **reward**

Now we have a **reward model** r_ϕ that represents* **goodness according to humans**

RLHF: Learning a **policy** that optimizes the **reward**

Now we have a **reward model** r_ϕ that represents* **goodness according to humans**

Now, learn a policy π_θ achieving **high reward**

RLHF: Learning a **policy** that optimizes the **reward**

Now we have a **reward model** r_ϕ that represents* **goodness according to humans**

Now, learn a policy π_θ achieving **high reward**

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

RLHF: Learning a **policy** that optimizes the **reward**

Now we have a **reward model** r_ϕ that represents* **goodness according to humans**

Now, learn a policy π_θ achieving **high reward**

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

The diagram illustrates the optimization objective. The equation is $\max_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y|x)} [r_\phi(x, y)]$. Below the equation, there are two text labels with arrows pointing to parts of the equation. The label "Sample from policy" has an arrow pointing to the π_θ in the subscript of the expectation operator. The label "Want high reward..." has an arrow pointing to the $r_\phi(x, y)$ term inside the expectation operator.

RLHF: Learning a **policy** that optimizes the **reward**

Now we have a **reward model** r_ϕ that represents* **goodness according to humans**

Now, learn a policy π_θ achieving **high reward** while **staying close** to original model π_{ref}

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

Sample from policy

Want high reward...

The diagram illustrates the optimization objective. The equation is $\max_{\pi_\theta} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(y|x)} [r_\phi(x, y)]$. An arrow points from the text 'Sample from policy' to the π_θ term in the denominator of the expectation. Another arrow points from the text 'Want high reward...' to the $r_\phi(x, y)$ term inside the expectation.

RLHF: Learning a **policy** that optimizes the **reward**

Now we have a **reward model** r_ϕ that represents* **goodness according to humans**

Now, learn a policy π_θ achieving **high reward** while **staying close** to original model π_{ref}

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

Sample from policy



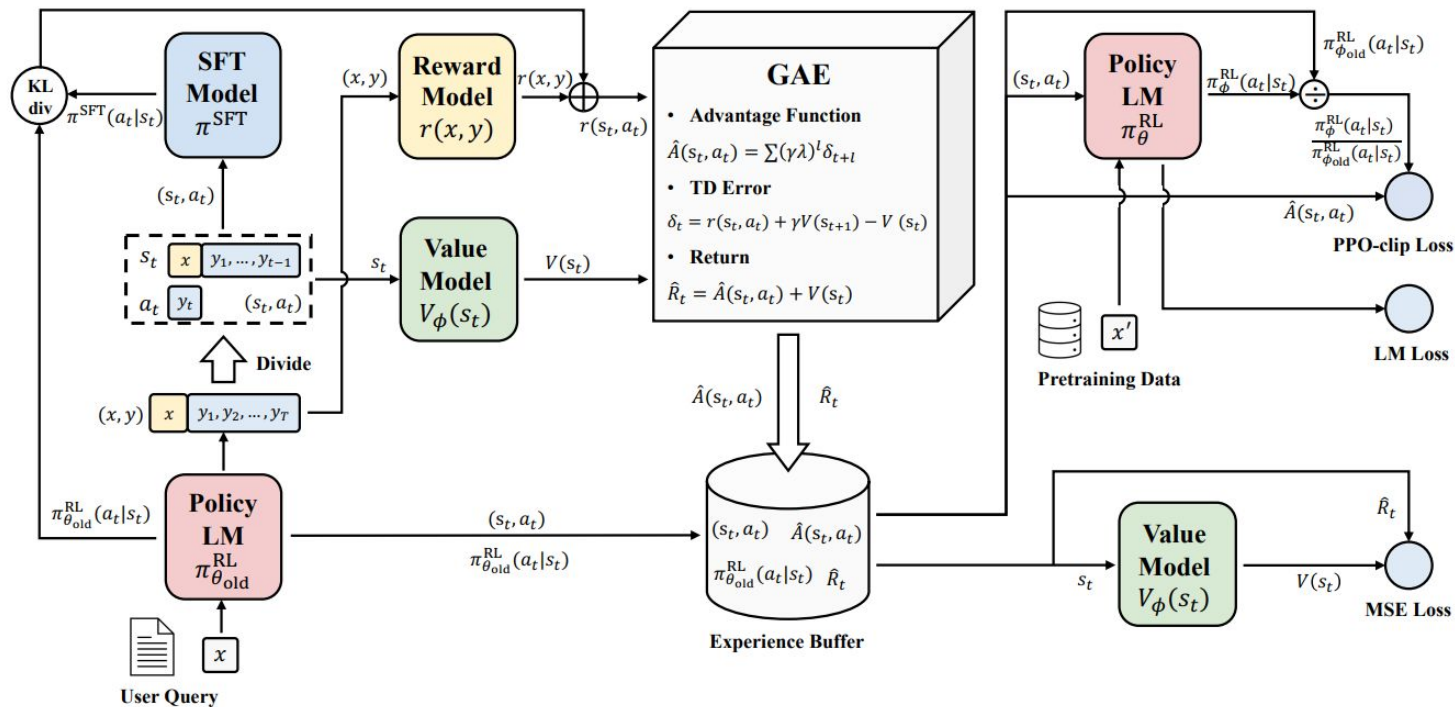
Want high reward...



...but keep KL to original model small!



RLHF: Learning a **policy** that optimizes the **reward**



Direct Preference Optimization

Direct Preference Optimization

RLHF Objective

(get **high reward**, stay **close**
to reference model)

Direct Preference Optimization

RLHF Objective

(get **high reward**, stay close
to reference model)

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

Direct Preference Optimization

RLHF Objective

(get **high reward**, stay close to reference model)

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

← **any** reward function

Direct Preference Optimization

RLHF Objective

(get **high reward**, stay close to reference model)

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

← **any** reward function

Closed-form Optimal Policy

(write **optimal policy** as function of **reward function**; from prior work)

Direct Preference Optimization

RLHF Objective

(get **high reward**, stay close to reference model)

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

← **any** reward function

Closed-form Optimal Policy

(write **optimal policy** as function of **reward function**; from prior work)

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

Direct Preference Optimization

RLHF Objective

(get **high reward**, stay close to reference model)

$$\max_{\pi} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi(y|x)} [r(x, y)] - \beta \mathbb{D}_{\text{KL}}(\pi(\cdot | x) \parallel \pi_{\text{ref}}(\cdot | x))$$

← **any** reward function

Closed-form Optimal Policy

(write **optimal policy** as function of **reward function**; from prior work)

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

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

Direct Preference Optimization

RLHF Objective

(get **high reward**, stay close to reference model)

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

← **any** reward function

Closed-form Optimal Policy

(write **optimal policy** as function of **reward function**; from prior work)

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

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

← Note **intractable sum** over possible responses; can't immediately use this

Direct Preference Optimization

RLHF Objective

(get **high reward**, stay close to reference model)

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

← **any** reward function

Closed-form Optimal Policy

(write **optimal policy** as function of **reward function**; from prior work)

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

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

← Note **intractable sum** over possible responses; can't immediately use this

Rearrange

(write **any reward function** as function of **optimal policy**)

Direct Preference Optimization

RLHF Objective

(get **high reward**, stay close to reference model)

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

← **any** reward function

Closed-form Optimal Policy

(write **optimal policy** as function of **reward function**; from prior work)

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

with $Z(x) = \sum_y \pi_{\text{ref}}(y | x) \exp\left(\frac{1}{\beta} r(x, y)\right)$ ← Note **intractable sum** over possible responses; can't immediately use this

Rearrange

(write **any** reward function as function of **optimal policy**)

$$r(x, y) = \underbrace{\beta \log \frac{\pi^*(y | x)}{\pi_{\text{ref}}(y | x)} + \beta \log Z(x)}_{\text{some parameterization of a reward function}}$$

Direct Preference Optimization

RLHF Objective

(get **high reward**, stay close to reference model)

$$\max_{\pi} \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi(y|x)} [r(x, y)] - \beta \mathbb{D}_{\text{KL}}(\pi(\cdot | x) \parallel \pi_{\text{ref}}(\cdot | x))$$

← **any** reward function

Closed-form Optimal Policy

(write **optimal policy** as function of **reward function**; from prior work)

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

with $Z(x) = \sum_y \pi_{\text{ref}}(y | x) \exp\left(\frac{1}{\beta} r(x, y)\right)$ ← Note **intractable sum** over possible responses; can't immediately use this

Rearrange

(write **any reward function** as function of **optimal policy**)

$$r(x, y) = \underbrace{\beta \log \frac{\pi^*(y | x)}{\pi_{\text{ref}}(y | x)}}_{\text{some parameterization of a reward function}} + \beta \log Z(x)$$

Ratio is **positive** if policy likes response more than reference model, **negative** if policy likes response less than ref. model

Direct Preference Optimization: Putting it together

Direct Preference Optimization: Putting it together

A loss function on
reward functions

Direct Preference Optimization: Putting it together

A loss function on
reward functions



A transformation
between reward
functions and policies

Direct Preference Optimization: Putting it together

A loss function on
reward functions

+

A transformation
between reward
functions and policies

=

A loss function
on policies

Direct Preference Optimization: Putting it together

Derived from the Bradley-Terry model of human preferences:

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

A loss function on
reward functions

+

A transformation
between reward
functions and policies

=

A loss function
on policies

Direct Preference Optimization: Putting it together

A loss function on reward functions

+

A transformation between reward functions and policies

=

A loss function on policies

Derived from the Bradley-Terry model of human preferences:

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

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

Direct Preference Optimization: Putting it together

Derived from the Bradley-Terry model of human preferences:

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

A loss function on reward functions



A transformation between reward functions and policies

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



A loss function on policies

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

Reward of preferred response

Reward of dispreferred response

Direct Preference Optimization: Putting it together

Derived from the Bradley-Terry model of human preferences:

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

A loss function on reward functions



A transformation between reward functions and policies

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

When substituting, the **log Z** term cancels, because the loss only cares about **difference** in rewards

Reward of preferred response

Reward of dispreferred response

A loss function on policies

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

Direct Preference Optimization: Putting it together

Derived from the Bradley-Terry model of human preferences:

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

A loss function on
reward functions



A transformation
between reward
functions and policies

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

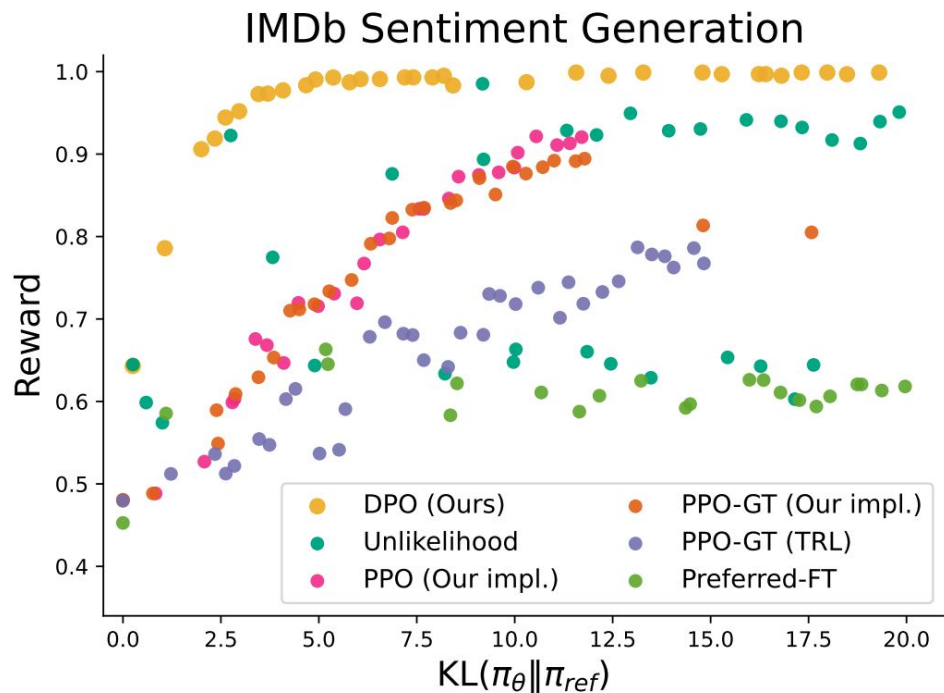
$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w | x)}{\pi_{\text{ref}}(y_w | x)} - \beta \log \frac{\pi_\theta(y_l | x)}{\pi_{\text{ref}}(y_l | x)} \right) \right]$$

Reward of **preferred** response

Reward of **dispreferred** response

Results

How Efficiently does DPO Trade off Reward & KL?



1. Generate positive IMDB reviews from GPT2-XL
2. Use pre-trained sentiment classifier as Gold RM
3. Create preferences based on Gold RM
4. Optimize with PPO and DPO

Models Trained With DPO

The screenshot shows the Open LLM Leaderboard interface. At the top, there's a navigation bar with 'Spaces', 'HuggingFaceH4', 'open_llm_leaderboard', and 'Building on CPU STORAGE'. Below that is the title 'Open LLM Leaderboard' and a description: 'The Open LLM Leaderboard aims to track, rank and evaluate open LLMs and chatbots. Submit a model for automated evaluation on the GPU cluster on the "Submit" page! The leaderboard's backend runs the great Eleuther AI Language Model Evaluation Harness - read more details in the "About" page!'.

The main content area has a search bar and filter options. The filters include 'Model types' (pretrained, fine-tuned, instruction-tuned, RL-tuned), 'Precision' (float16, bfloat16, 8bit, 4bit, GPTQ), and 'Model sizes (in billions of parameters)' (7, -1.5, -3, -7, -13, -35, -60, 70+). There are also checkboxes for 'Show private/deleted models', 'Show merges', 'Show MoE', and 'Show flagged models'.

The table below shows the following data:

T	Model	Average	ARC	HellaSwag	MMLU	TruthfulQA	Winogrande	GSM8K
1	udhrai/Tuzdus	74.66	73.38	88.56	64.52	67.11	86.66	67.7
2	fblgit/UVA-TheBeagle-7B-v1	73.87	73.04	88	63.48	69.85	82.16	66.72
3	argilla/distilabelled-Marcos04-7B-sleep	73.63	70.73	87.47	65.22	65.1	82.08	71.19
4	mlabonne/NeuralMarco04-7B	73.57	71.42	87.59	64.84	65.64	81.22	70.74
5	ahideen/NexoNimbus-7B	73.5	70.82	87.86	64.69	62.43	84.85	70.36
6	Neutonsovo/neutonsovo-7B-v0.2	73.44	73.04	88.32	65.15	71.62	80.66	62.47
7	argilla/distilabelled-Marcos04-7B-sleep-full	73.4	70.65	87.55	65.33	64.21	82	70.66
8	Cultrix/MistralTrix-v1	73.39	72.27	88.33	65.24	70.73	80.98	62.77
9	xyandi/MusingGatesollax	73.33	72.53	88.34	65.26	70.93	80.66	62.24
10	Neutonsovo/neutonsovo-7B-v0.3	73.29	72.7	88.26	65.1	71.35	80.9	61.41
11	Cultrix/MistralTrixTest	73.17	72.53	88.4	65.22	70.77	81.37	60.73
12	sanir-fama/SanirGPT-v1	73.11	69.54	87.04	65.3	63.37	81.69	71.72
13	SanjiMatsuki/Lelantos-DPO-7B	73.09	71.08	87.22	64	67.77	80.03	68.46

Handwritten annotations in red and black ink are present on the table:

- 'DPO' is written in red above the first row.
- 'DPO (& VNA)' is written in red above the second row.
- 'DPO' is written in red above the third row.
- 'Merge (of DPO models)' is written in black above the fourth and fifth rows.
- 'DPO' is written in red above the sixth row.
- 'DPO' is written in red above the seventh row.
- 'DPO' is written in red above the eighth row.
- 'DPO' is written in red above the ninth row.
- 'DPO' is written in red above the tenth row.
- 'No info bit prob DPO, given' is written in black above the eleventh row.
- 'Merge (incl. DPO)' is written in black above the twelfth row.
- 'DPO' is written in red above the thirteenth row.

Large-Scale DPO Training

Large-Scale DPO Training

Mistral

4 Instruction Fine-tuning

We train Mistral – Instruct using supervised fine-tuning (SFT) on an instruction dataset followed by Direct Preference Optimization (DPO) [25] on a paired feedback dataset. Mistral – Instruct reaches a score of 8.30 on MT-Bench [33] (see Table 2), making it the best open-weights model as of December 2023. Independent human evaluation conducted by LMSys is reported in Figure 6³ and shows that Mistral – Instruct outperforms GPT-3.5-Turbo, Gemini Pro, Claude-2.1, and Llama 2 70B chat.

Model	★ Arena Elo rating	📄 MT-bench (score)	License
GPT-4-Turbo	1243	9.32	Proprietary
GPT-4-0314	1192	8.96	Proprietary
GPT-4-0613	1158	9.18	Proprietary
Claude-1	1149	7.9	Proprietary
Claude-2.0	1131	8.06	Proprietary
Mistral-8x7b-Instruct-v0.1	1121	8.3	Apache 2.0
Claude-2.1	1117	8.18	Proprietary
GPT-3.5-Turbo-0613	1117	8.39	Proprietary
Gemini Pro	1111		Proprietary
Claude-Instant-1	1110	7.85	Proprietary
Tulu-2-DPO-70B	1110	7.89	A12 ImpACT Low-risk
Yi-34B-Chat	1110		Yi License
GPT-3.5-Turbo-0314	1105	7.94	Proprietary
Llama-2-70b-chat	1077	6.86	Llama 2 Community

Figure 6: LMSys Leaderboard. (Screenshot from Dec 22, 2023) Mistral 8x7B Instruct v0.1 achieves an Arena Elo rating of 1121 outperforming Claude-2.1 (1117), all versions of GPT-3.5-Turbo (1117 best), Gemini Pro (1111), and Llama-2-70b-chat (1077). Mistral is currently the best open-weights model by a large margin.

Large-Scale DPO Training

Mistral

4 Instruction Fine-tuning

We train Mistral – Instruct using supervised fine-tuning (SFT) on an instruction dataset followed by Direct Preference Optimization (DPO) [25] on a paired feedback dataset. Mistral – Instruct reaches a score of 8.30 on MT-Bench [33] (see Table 2), making it the best open-weights model as of December 2023. Independent human evaluation conducted by LMSys is reported in Figure 6³ and shows that Mistral – Instruct outperforms GPT-3.5-Turbo, Gemini Pro, Claude-2.1, and Llama 2 70B chat.

Model	Arena Elo rating	MT-bench (score)	License
GPT-4-Turbo	1243	9.32	Proprietary
GPT-4-0314	1192	8.96	Proprietary
GPT-4-0613	1158	9.18	Proprietary
Claude-1	1149	7.9	Proprietary
Claude-2.0	1131	8.06	Proprietary
Mistral-8x7b-Instruct-v0.1	1121	8.3	Apache 2.0
Claude-2.1	1117	8.18	Proprietary
GPT-3.5-Turbo-0613	1117	8.39	Proprietary
Gemini Pro	1111		Proprietary
Claude-Instant-1	1110	7.85	Proprietary
Tulu-2-DPO-70B	1110	7.89	A12 Impact Low-risk
Yi-34B-Chat	1110		Yi License
GPT-3.5-Turbo-0314	1105	7.94	Proprietary
Llama-2-70b-chat	1077	6.86	Llama 2 Community

Figure 6: LMSys Leaderboard. (Screenshot from Dec 22, 2023) Mistral 8x7B Instruct v0.1 achieves an Arena Elo rating of 1121 outperforming Claude-2.1 (1117), all versions of GPT-3.5-Turbo (1117 best), Gemini Pro (1111), and Llama-2-70b-chat (1077). Mistral is currently the best open-weights model by a large margin.

LLaMa3

Instruction fine-tuning

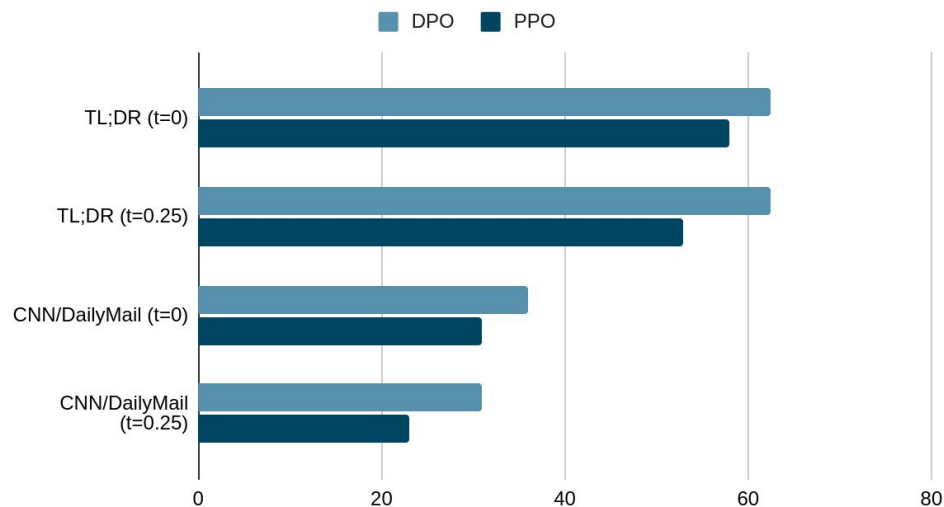
To fully unlock the potential of our pretrained models in chat use cases, we innovated on our approach to instruction-tuning as well. Our approach to post-training is a combination of supervised fine-tuning (SFT), rejection sampling, proximal policy optimization (PPO), and direct preference optimization (DPO). The quality of the prompts that are used in SFT and the preference rankings that are used in PPO and DPO has an outsized influence on the performance of aligned models. Some of our biggest improvements in model quality came from carefully curating this data and performing multiple rounds of quality assurance on annotations provided by human annotators.

Learning from preference rankings via PPO and DPO also greatly improved the performance of Llama 3 on reasoning and coding tasks. We found that if you ask a model a reasoning question that it struggles to answer, the model will sometimes produce the right reasoning trace: The model knows how to produce the right answer, but it does not know how to select it. Training on preference rankings enables the model to learn how to select it.

The DPO vs PPO Debate

DPO vs PPO: Empirics

Win Rates



1. DPO is trained only on the Reddit TL;DR feedback data.
2. PPO uses a trained reward function and additional prompts for RL training.
3. We evaluate the trained policies on OOD CNN/DailyMail news summarization task.

DPO vs PPO:

DPO vs PPO:

DPO fits an implicit reward function:

DPO vs PPO:

DPO fits an implicit reward function:

1. Is the DPO implicit reward as good as the explicit one?

DPO vs PPO:

DPO fits an implicit reward function:

1. Is the DPO implicit reward as good as the explicit one?
2. Does using a weaker optimizer, such as PPO provide a better solution (regularization).

DPO vs PPO:

DPO fits an implicit reward function:

- 1. Is the DPO implicit reward as good as the explicit one?**
2. Does using a weaker optimizer, such as PPO provide a better solution (regularization).

DPO vs PPO: Reward Function Quality - Chat

RewardBench: Evaluating Reward Models

Evaluating the capabilities, safety, and pitfalls of reward models

[Code](#) | [Eval_Dataset](#) | [Prior Test Sets](#) | [Results](#) | [Paper](#) | Total models: 74



🏆 RewardBench Leaderboard 🔍 RewardBench - Detailed Prior Test Sets About Dataset Viewer

Model Search (delimit with ,)

Seq. Classifiers DPO Custom Classifiers Generative AI2 Experiments

▲	Model	▲	Model Type	▲	Score	▲	Chat	▲	Chat Hard	▼	Safety	▲	Reasoning	▲	Prior Sets (0.5 weight)	▲
24	Owen/Owen1.5-14B-Chat		DPO		69.76		57.3		70.2		76.3		89.6		41.2	
26	Owen/Owen1.5-7B-Chat		DPO		68.75		53.6		69.1		74.8		90.4		42.9	
12	upstage/SOLAR-10.7B-Instruct-v1.0		DPO		73.99		81.6		68.6		85.5		72.5		49.5	
29	Owen/Owen1.5-72B-Chat		DPO		68.21		62.3		66		72		85.5		42.3	
3	openbmb/Eurus-RM-7b		Seq. Classifier		81.55		98		65.6		81.2		86.3		71.7	
1	Cohere_March_2024		Custom Classifier		85.69		94.7		65.1		90.3		98.2		74.6	
2	sfairXC/FsfairX-LLaMA3-RM-v0.1		Seq. Classifier		83.62		99.4		65.1		87.8		86.4		74.9	
11	mistralai/Mixtral-8x7B-Instruct-v0.1		DPO		74.74		95		64		73.4		78.7		50.3	
33	Owen/Owen1.5-MoE-A2.7B-Chat		DPO		67.54		72.9		63.2		67.8		77.4		45.4	
49	Owen/Owen1.5-0.5B-Chat		DPO		55.01		35.5		62.9		66.1		59.8		46.3	
17	HuggingFaceH4/zephyr-7b-beta		DPO		71.77		95.3		62.7		61		77.9		52.2	
48	Owen/Owen1.5-4B-Chat		DPO		56.14		38.8		62.7		61.8		66.9		44.7	
13	HuggingFaceH4/zephyr-7b-alpha		DPO		73.42		91.6		62.5		74.3		75.1		53.5	
											66.3		83.9		55.7	

RewardBench: Evaluating Reward Models for Language Modeling, Lambert et. al.

Stanford University

DPO vs PPO: Reward Function Quality - Reasoning

RewardBench: Evaluating Reward Models

Evaluating the capabilities, safety, and pitfalls of reward models

[Code](#) | [Eval. Dataset](#) | [Prior Test Sets](#) | [Results](#) | [Paper](#) | Total models: 74



RewardBench Leaderboard

Model Search (delimit with ,)

Seq. Classifiers DPO Custom Classifiers Generative AI2 Experiments

▲	Model	▲	Model Type	▲	Score	▲	Chat	▲	Chat Hard	▲	Safety	▲	Reasoning	▼	Prior Sets (0.5 weight)	▲
1	Cohere_March_2024		Custom Classifier		85.69		94.7		65.1		90.3		98.2		74.6	
26	Owen/Owen1.5-7B-Chat		DPO		68.75		53.6		69.1		74.8		90.4		42.9	
24	Owen/Owen1.5-14B-Chat		DPO		69.76		57.3		70.2		76.3		89.6		41.2	
7	stabilityai/stablelm-2-12b-chat		DPO		77.42		96.6		55.5		82.6		89.4		48.4	
19	jondurbin/bagel-dpo-34b-v0.5		DPO		71.5		93.9		55		61.5		88.9		44.9	
22	0-hero/Matter-0.1-7B-DPO-preview		DPO		71.19		89.4		57.7		58		88.5		53.5	
4	NexusFlow/Starling-RM-34B		Seq. Classifier		81.44		96.9		57.2		88.2		88.5		71.4	
2	sfairXC/FsfairX-LLaMA3-RM-v0.1		Seq. Classifier		83.62		99.4		65.1		87.8		86.4		74.9	
3	openbmb/Eurus-RM-7b		Seq. Classifier		81.55		98		65.6		81.2		86.3		71.7	
29	Owen/Owen1.5-72B-Chat		DPO		68.21		62.3		66		72		85.5		42.3	
15	0-hero/Matter-0.1-7B-boost-DPO-preview		DPO		73.35		91.1		61		66.3		83.9		55.7	
36	openbmb/MiniCPM-2B-dpo-fp32		DPO		66.25		89.1		49.3		52.5		82.3		49.6	
16	HuggingFaceH4/starcoder2-15b-v0.1		DPO		72.08		93.9		55.5		65.8		81.6		55.2	
11	stabilityai/stablelm-2-12b-chat		DPO		71.21		95		64		73.4		78.7		50.3	

RewardBench: Evaluating Reward Models for Language Modeling, Lambert et. al.

Stanford University

DPO vs PPO:

DPO fits an implicit reward function:

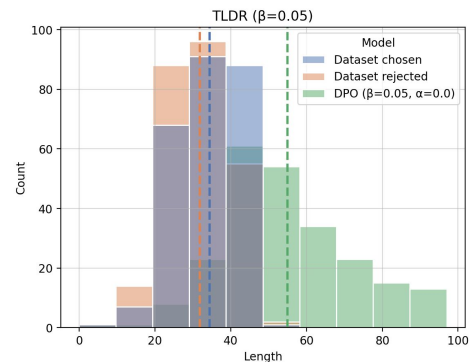
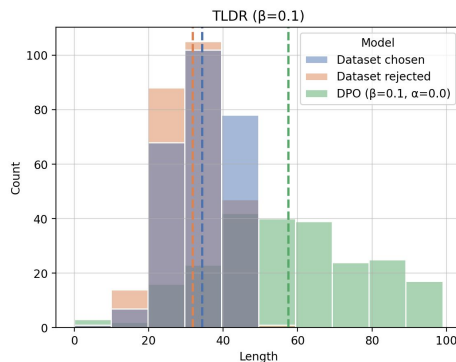
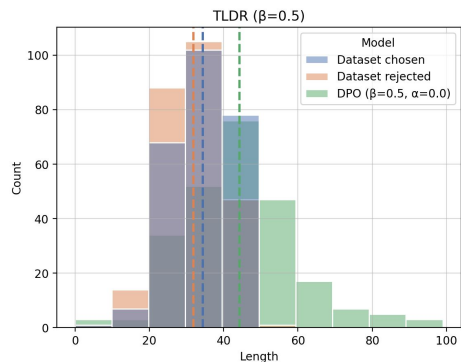
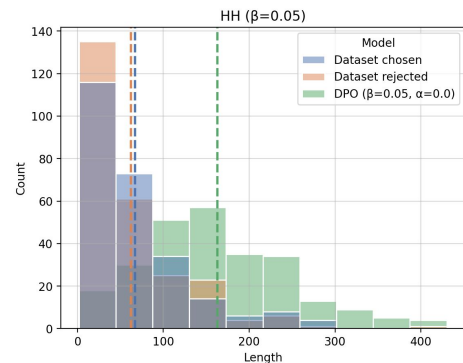
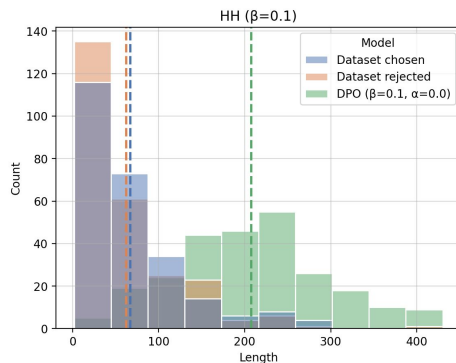
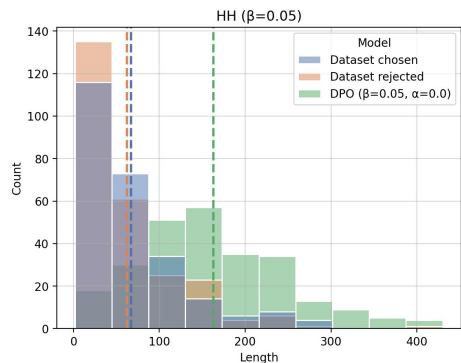
1. **Is the DPO implicit reward as good as the explicit one?**
2. Does using a weaker optimizer, such as PPO provide a better solution (regularization).

DPO vs PPO:

DPO fits an implicit reward function:

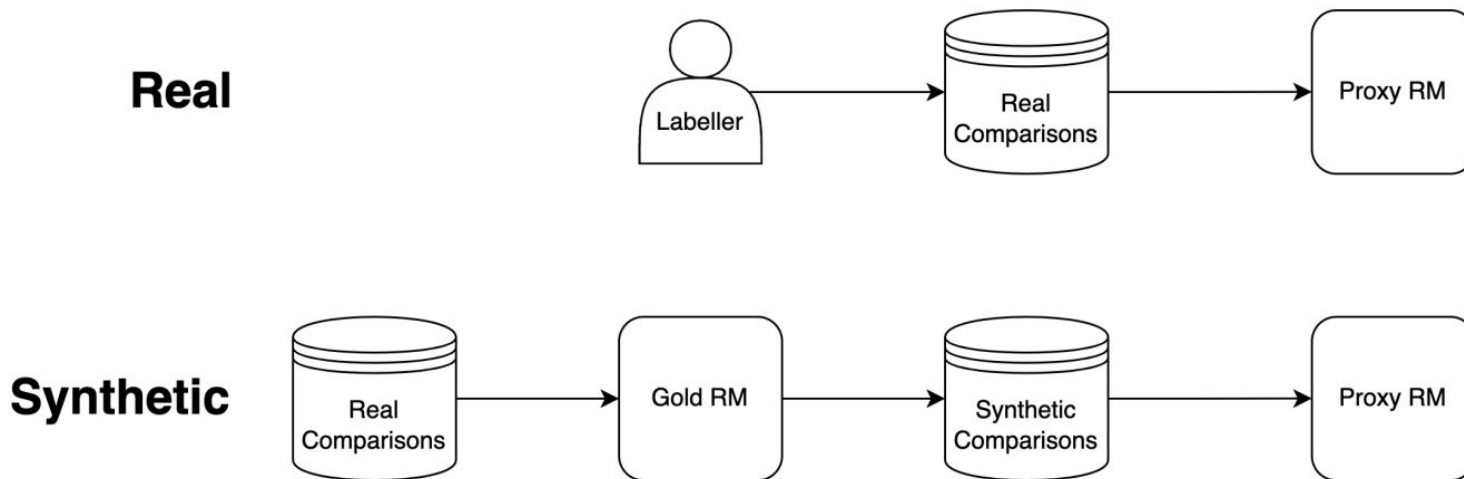
1. Is the DPO implicit reward as good as the explicit one?
2. **Does using a weaker optimizer, such as PPO provide a better solution (regularization).**

DPO vs PPO: Reward Hacking

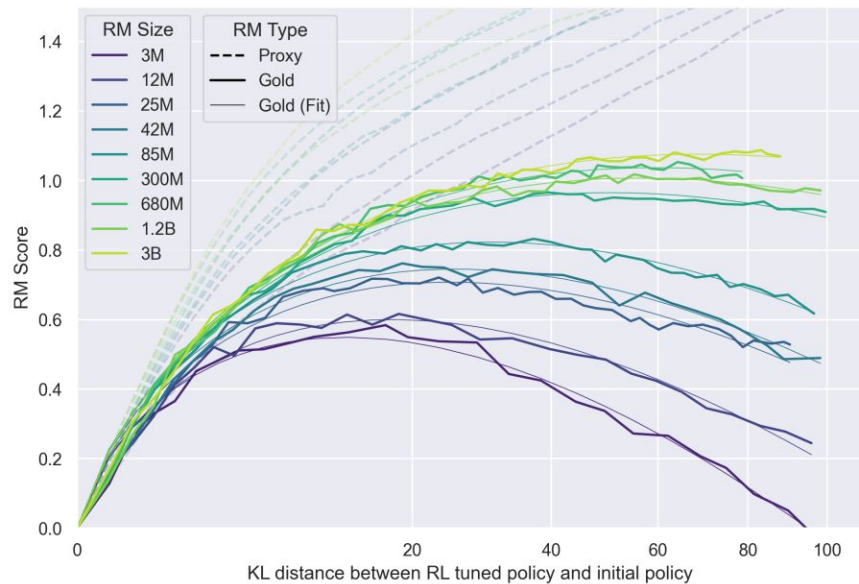


Disentangling Length from Quality in Direct Preference Optimization, Park et. al.

DPO vs PPO: Reward Hacking



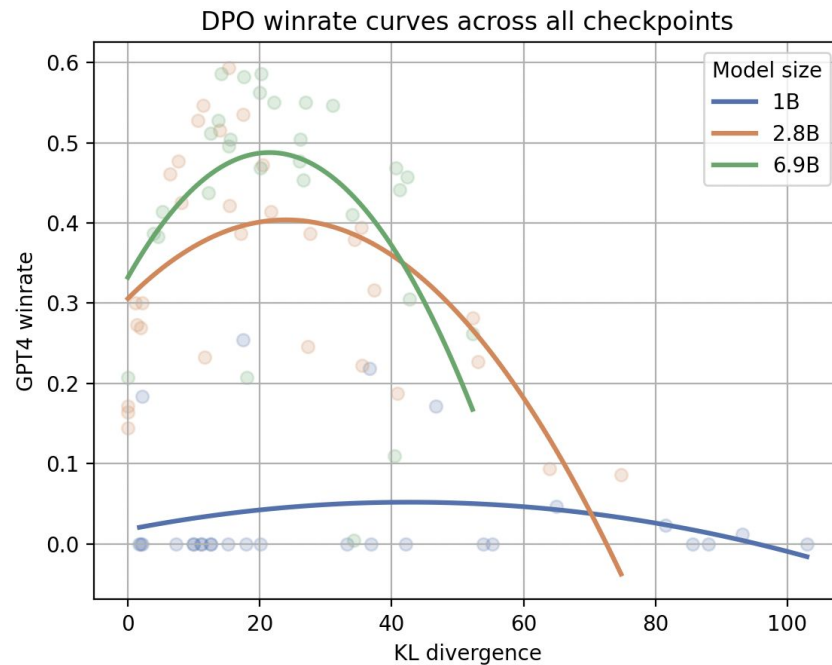
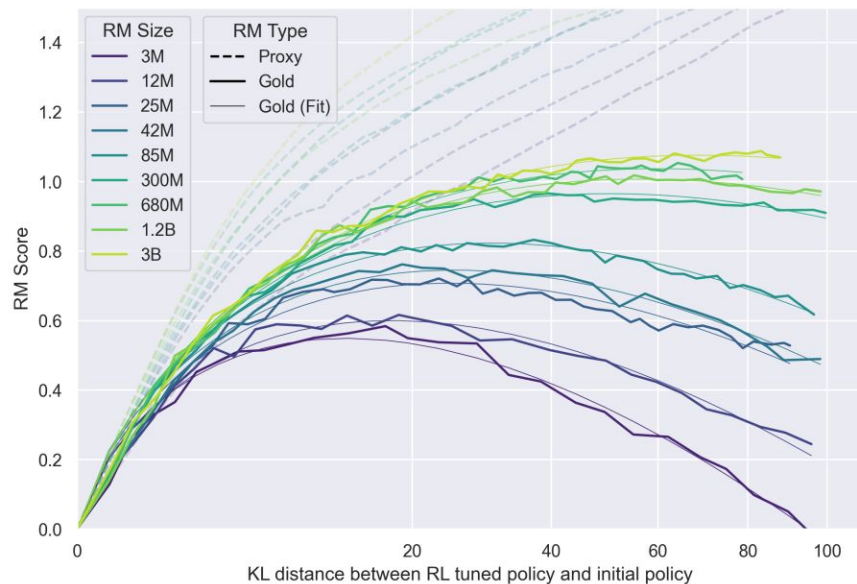
DPO vs PPO: Reward Hacking



Scaling Laws for Reward Model Overoptimization, Gao et. al.

Stanford University

DPO vs PPO: Reward Hacking



Scaling Laws for Reward Model Overoptimization, Gao et. al.

Stanford University

Conclusion

Conclusion

1. DPO optimizes the same classical RLHF objective

Conclusion

1. DPO optimizes the same classical RLHF objective
2. Is simple and computationally cheap

Conclusion

1. DPO optimizes the same classical RLHF objective
2. Is simple and computationally cheap
3. Like classical RLHF it is prone to hacking

Next Steps

1. How to optimize DPO robustly (prevent reward hacking)
2. Online fine-tuning (preference elicitation)
3. RLHF across modalities
 - a. Vision-Language Models
 - b. Diffusion Models
 - i. Text-to-image
 - ii. Text-to-video
 - iii. Speech and music
 - c. Protein and molecule generation
 - d. Robot Safety
4. Multi-turn interactions
5. Agents, tool use, etc..

DPO for Aligning Modalities in VLMs

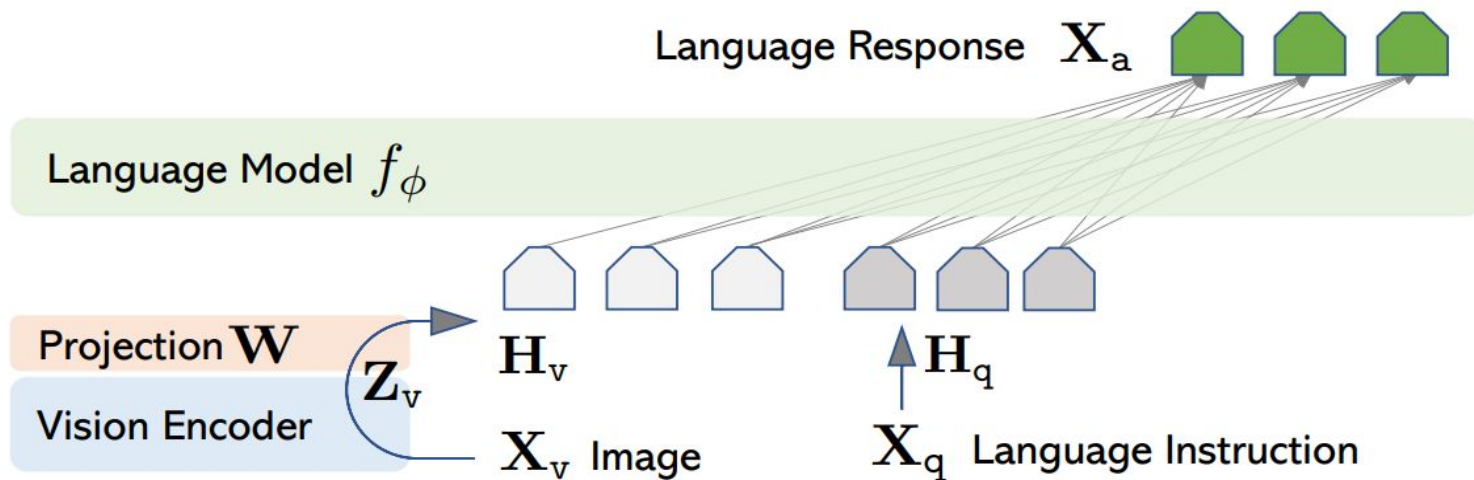


Figure 1: LLaVA network architecture.

DPO for Aligning Modalities in VLMs

a. Hallucinating Image Captioning Tasks

Prompt: Describe this image.



Preferred response

The image shows two men in a small boat on the water. One man is standing near the center of the boat, while the other is seated and rowing. The man wearing a headscarf is watching television, which is placed on the boat, adding an interesting element to the scene. There are a few other boats visible in the background, indicating that they might be in a city river or a busy waterway. The presence of the television in the boat adds a unique touch to their leisurely activity.

Dis-preferred response

The image shows two men in a small boat on the water, equipped with fishing poles (*hallucinatory entity*) for their leisurely activity. One man is sitting near the center of the boat (*entity attributes*), fishing with his fishing pole (*hallucinatory entity*), while the other is seated and rowing. They are both watching a television (*logical relationship*), which is placed on the boat, adding an interesting element to the scene. There are a few other boats visible in the background, indicating that they might be in a city river or a busy waterway. The presence of the fishing poles and the television in the boat adds a unique touch to their relaxing day on the water.

b. Hallucinating Reasoning Tasks

Prompt: Is there a person standing next to water in the image?



Preferred response

Reasoning: The picture depicts a sunset scene in which I see a lake and open sky.
Answer: Yes, there is a person standing next to some blue water in the image.

Dis-preferred response

Reasoning: The image depicts a sunset scene with a man standing on an open ice field (*false reasoning, hallucinatory entities*) with what appears to be a few buildings visible in the distance.
Answer: No, the person is standing next to a vast snowy field, not water (*incorrect answer*).

DPO for Aligning Modalities in VLMs

Prompt: Describe this image.



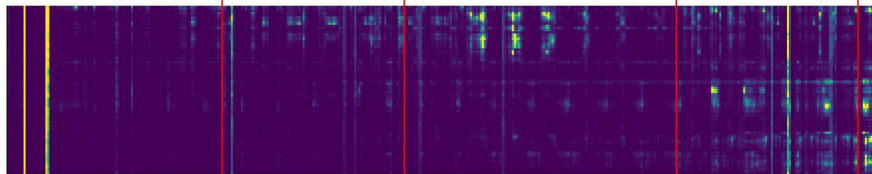
Ours

The image shows a man sitting down and brushing the teeth of a cat with his fingers. The cat appears to be reacting to or accepting the tooth brushing process. The man and the cat are the main focus of the scene.
The setting appears to be indoors, as there is a person visible in the image along with the close interaction between the man and the cat. The cat is located near the center of the scene, while the man is positioned on the left side of the image.

LLaVA 1.5

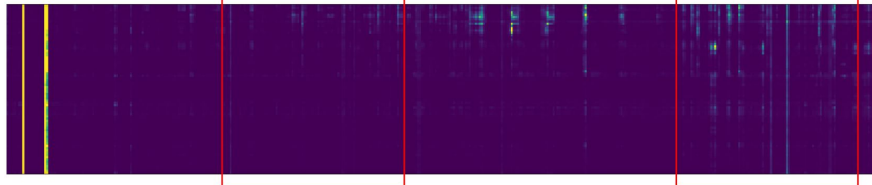
The image features a person brushing an open mouth cat's teeth with a small electric toothbrush. They are in a **kitchen** setting, focused on maintaining good oral hygiene. **An oven** is visible in the background, adding to the cozy **kitchen environment**.
There is also a **tie** in the scene, likely placed on or hung up nearby, possibly indicating that someone's clothing is being attended to or is hanging out to be worn.

Ours



Textual tokens

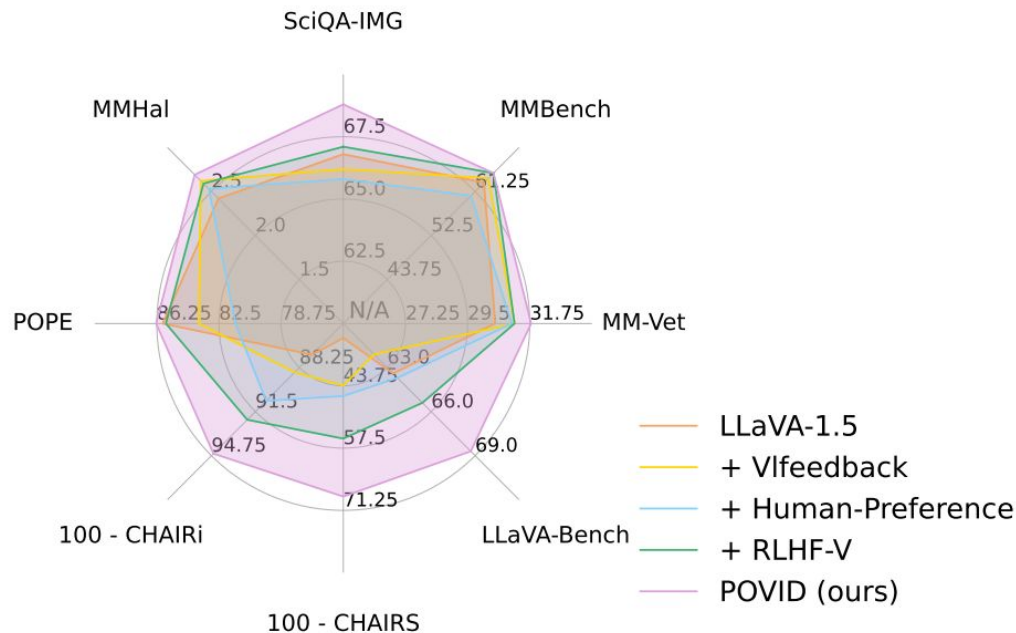
LLaVA 1.5



Visual tokens

Textual tokens

DPO for Aligning Modalities in VLMs

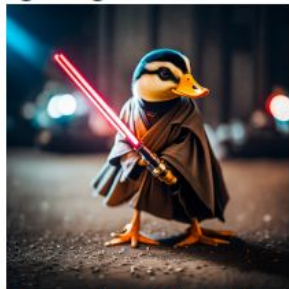


Aligning Modalities in Vision Large Language Models via Preference Fine-tuning, Zhou et. al.

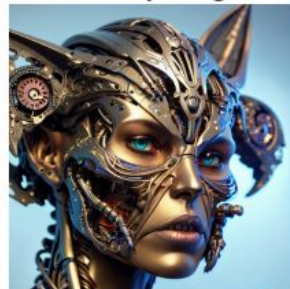
Stanford University

DPO for Diffusion

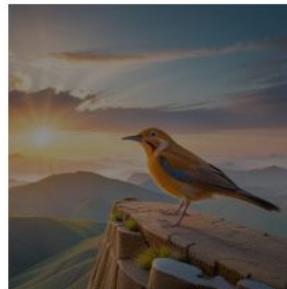
“jedi duck holding a lightsaber”



“Two-faced biomechanical cyborg...”



“A bird with 8 spider legs”



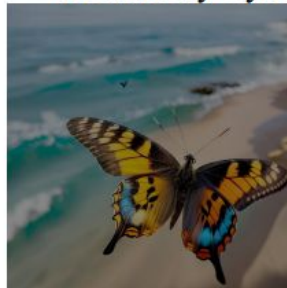
“a square green owl made of fimo”



“insanely detailed portrait, wise man”



“A butterfly flying above an ocean”



Diffusion Model Alignment Using Direct Preference Optimization, Wallace et. al.

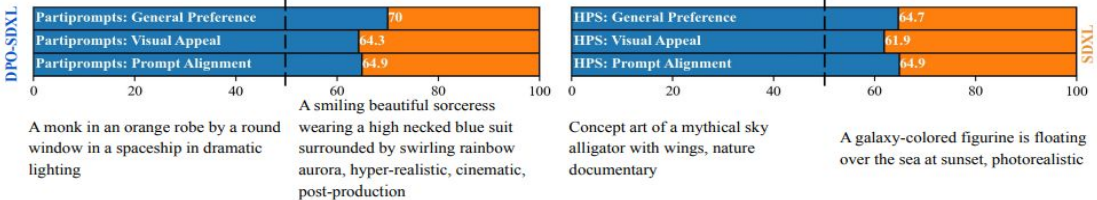
Stanford University

DPO for Diffusion

$$L(\theta) \leq -\mathbb{E}_{t, \epsilon^w, \epsilon^l} \log \sigma \left(-\beta T \omega(\lambda_t) \left(\underbrace{\|\epsilon^w - \epsilon_{\theta}(\mathbf{x}_t^w, t)\|^2 - \|\epsilon^w - \epsilon_{\text{ref}}(\mathbf{x}_t^w, t)\|^2}_{\text{reward of preferred image}} - \underbrace{\left(\|\epsilon^l - \epsilon_{\theta}(\mathbf{x}_t^l, t)\|^2 - \|\epsilon^l - \epsilon_{\text{ref}}(\mathbf{x}_t^l, t)\|^2 \right)}_{\text{reward of dispreferred image}} \right) \right)$$

“Diffuse along the **preferred image chain** and away from the **dispreferred image chain**”

DPO for Diffusion



A monk in an orange robe by a round window in a spaceship in dramatic lighting

A smiling beautiful sorceress wearing a high necked blue suit surrounded by swirling rainbow aurora, hyper-realistic, cinematic, post-production

Concept art of a mythical sky alligator with wings, nature documentary

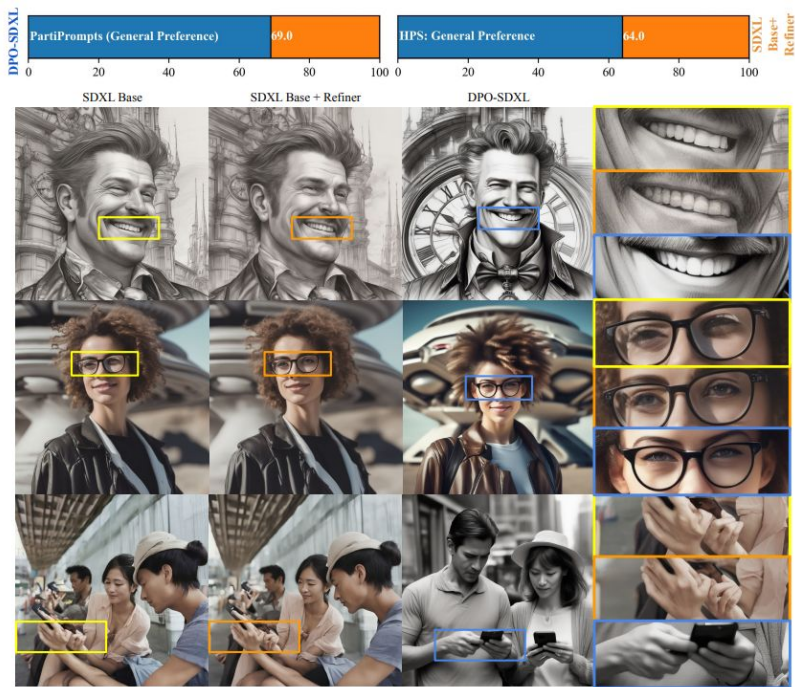
A galaxy-colored figurine is floating over the sea at sunset, photorealistic



Diffusion Model Alignment Using Direct Preference Optimization, Wallace et. al.

Stanford University

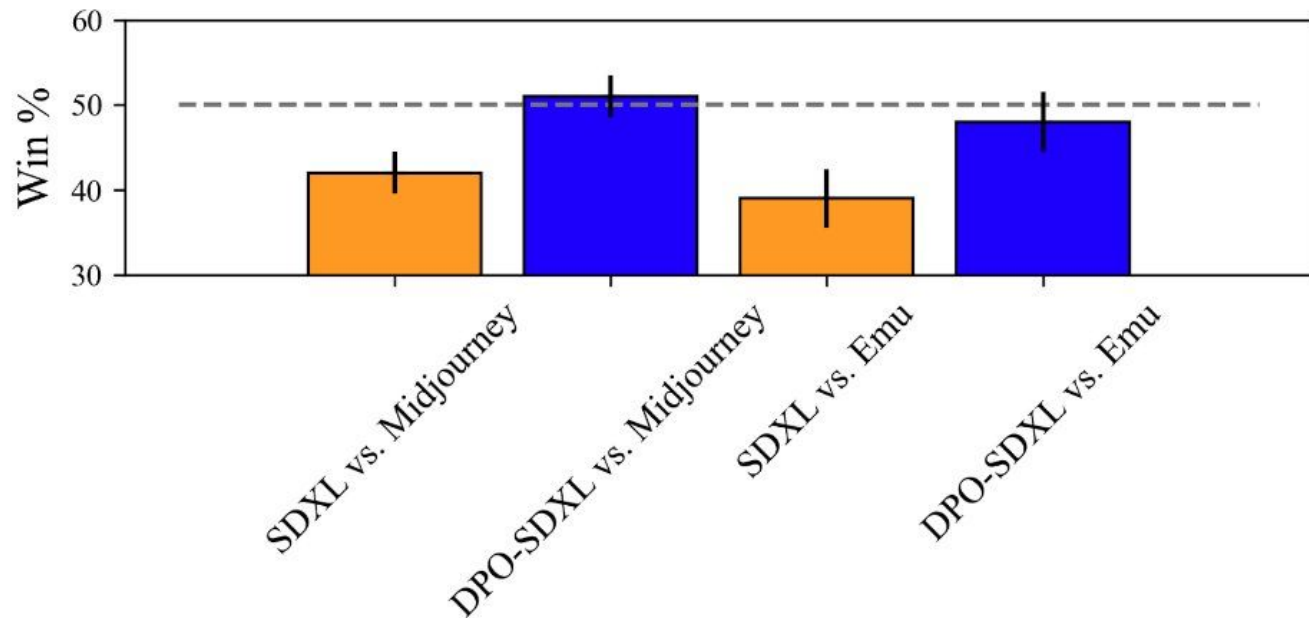
DPO for Diffusion



Diffusion Model Alignment Using Direct Preference Optimization, Wallace et. al.

Stanford University

DPO for Diffusion

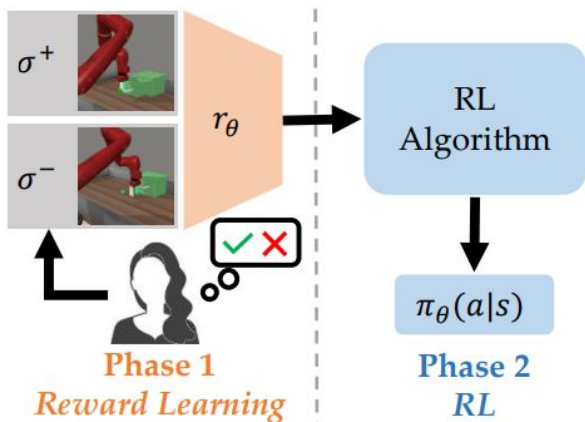


Diffusion Model Alignment Using Direct Preference Optimization, Wallace et. al.

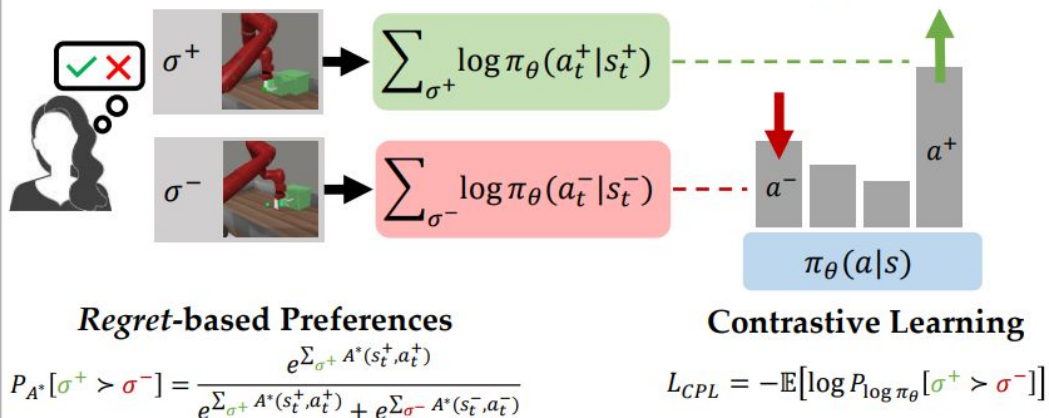
Stanford University

DPO and Control

Standard Two-Phase RLHF

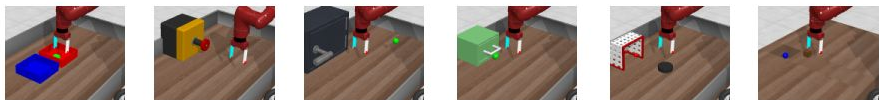


Contrastive Preference Learning



Contrastive Preference Learning: Learning from Human Feedback without RL, Hejna et. al.

DPO and Control



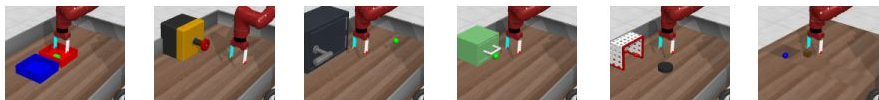
Bin Picking Button Press Door Open Drawer Open Plate Slide Sweep Into

State	2.5k Dense	SFT	66.9 ± 2.1	21.6 ± 1.6	63.3 ± 1.9	62.6 ± 2.4	41.6 ± 3.5	51.9 ± 2.1
		P-IQL	70.6 ± 4.1	16.2 ± 5.4	69.0 ± 6.2	71.1 ± 2.3	49.6 ± 3.4	60.6 ± 3.6
		CPL	80.0 ± 2.5	24.5 ± 2.1	80.0 ± 6.8	83.6 ± 1.6	61.1 ± 3.0	70.4 ± 3.0
Image	2.5k Dense	SFT	74.7 ± 4.8	20.8 ± 2.4	62.9 ± 2.3	64.5 ± 7.6	44.5 ± 3.2	52.5 ± 2.5
		P-IQL	83.7 ± 0.4	22.1 ± 0.8	68.0 ± 4.6	76.0 ± 4.6	51.2 ± 2.4	67.7 ± 4.4
		CPL	80.0 ± 4.9	27.5 ± 4.2	73.6 ± 6.9	80.3 ± 1.4	57.3 ± 5.9	68.3 ± 4.8
State	20k Sparse	SFT	67.0 ± 4.9	21.4 ± 2.7	63.6 ± 2.4	63.5 ± 0.9	41.9 ± 3.1	50.9 ± 3.2
		P-IQL	75.0 ± 3.3	19.5 ± 1.8	79.0 ± 6.6	76.2 ± 2.8	55.5 ± 4.2	73.4 ± 4.2
		CPL	83.2 ± 3.5	29.8 ± 1.8	77.9 ± 9.3	79.1 ± 5.0	56.4 ± 3.9	81.2 ± 1.6
Image	20k Sparse	SFT	71.5 ± 1.9	22.3 ± 2.9	65.2 ± 2.2	67.5 ± 1.1	41.3 ± 2.8	55.8 ± 2.9
		P-IQL	80.0 ± 2.3	27.2 ± 4.1	74.8 ± 5.8	80.3 ± 1.2	54.8 ± 5.8	72.5 ± 2.0
		CPL	78.5 ± 3.1	31.3 ± 1.6	70.2 ± 2.1	79.5 ± 1.4	61.0 ± 4.2	72.0 ± 1.8
Oracle	% BC	10%	62.6 ± 2.6	18.9 ± 1.7	57.5 ± 3.0	61.5 ± 3.7	39.1 ± 2.5	49.3 ± 2.1
	5%	64.6 ± 4.1	18.2 ± 0.6	59.8 ± 1.6	61.3 ± 1.8	38.6 ± 2.5	49.2 ± 1.9	

Behavior Cloning

Contrastive Preference Learning: Learning from Human Feedback without RL, Hejna et. al.

DPO and Control



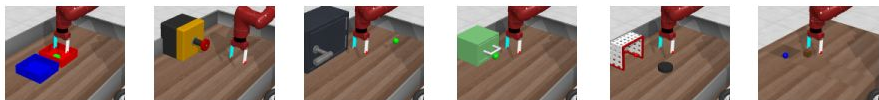
Bin Picking Button Press Door Open Drawer Open Plate Slide Sweep Into

		Bin Picking	Button Press	Door Open	Drawer Open	Plate Slide	Sweep Into
State 2.5k Dense	SFT	66.9 ± 2.1	21.6 ± 1.6	63.3 ± 1.9	62.6 ± 2.4	41.6 ± 3.5	51.9 ± 2.1
	P-IQL	70.6 ± 4.1	16.2 ± 5.4	69.0 ± 6.2	71.1 ± 2.3	49.6 ± 3.4	60.6 ± 3.6
	CPL	80.0 ± 2.5	24.5 ± 2.1	80.0 ± 6.8	83.6 ± 1.6	61.1 ± 3.0	70.4 ± 3.0
Image 2.5k Dense	SFT	74.7 ± 4.8	20.8 ± 2.4	62.9 ± 2.3	64.5 ± 7.6	44.5 ± 3.2	52.5 ± 2.5
	P-IQL	83.7 ± 0.4	22.1 ± 0.8	68.0 ± 4.6	76.0 ± 4.6	51.2 ± 2.4	67.7 ± 4.4
	CPL	80.0 ± 4.9	27.5 ± 4.2	73.6 ± 6.9	80.3 ± 1.4	57.3 ± 5.9	68.3 ± 4.8
State 20k Sparse	SFT	67.0 ± 4.9	21.4 ± 2.7	63.6 ± 2.4	63.5 ± 0.9	41.9 ± 3.1	50.9 ± 3.2
	P-IQL	75.0 ± 3.3	19.5 ± 1.8	79.0 ± 6.6	76.2 ± 2.8	55.5 ± 4.2	73.4 ± 4.2
	CPL	83.2 ± 3.5	29.8 ± 1.8	77.9 ± 9.3	79.1 ± 5.0	56.4 ± 3.9	81.2 ± 1.6
Image 20k Sparse	SFT	71.5 ± 1.9	22.3 ± 2.9	65.2 ± 2.2	67.5 ± 1.1	41.3 ± 2.8	55.8 ± 2.9
	P-IQL	80.0 ± 2.3	27.2 ± 4.1	74.8 ± 5.8	80.3 ± 1.2	54.8 ± 5.8	72.5 ± 2.0
	CPL	78.5 ± 3.1	31.3 ± 1.6	70.2 ± 2.1	79.5 ± 1.4	61.0 ± 4.2	72.0 ± 1.8
Oracle % BC	10%	62.6 ± 2.6	18.9 ± 1.7	57.5 ± 3.0	61.5 ± 3.7	39.1 ± 2.5	49.3 ± 2.1
	5%	64.6 ± 4.1	18.2 ± 0.6	59.8 ± 1.6	61.3 ± 1.8	38.6 ± 2.5	49.2 ± 1.9

Offline RL

Contrastive Preference Learning: Learning from Human Feedback without RL, Hejna et. al.

DPO and Control



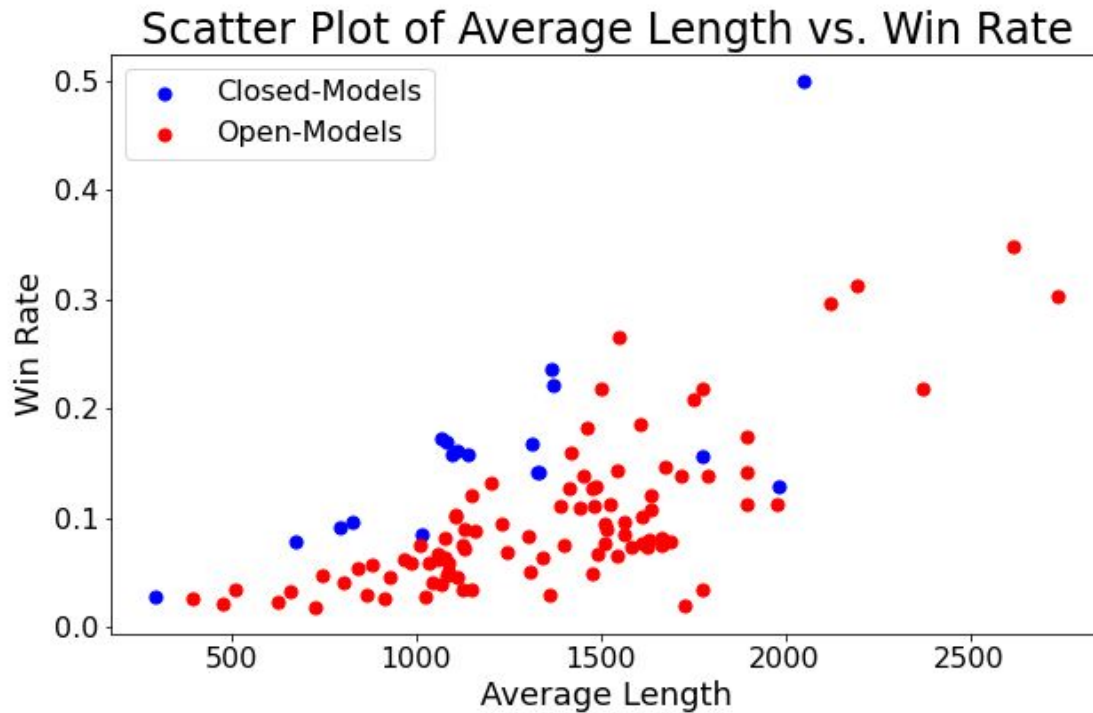
Bin Picking Button Press Door Open Drawer Open Plate Slide Sweep Into

		Bin Picking	Button Press	Door Open	Drawer Open	Plate Slide	Sweep Into	
State	2.5k Dense	SFT	66.9 ± 2.1	21.6 ± 1.6	63.3 ± 1.9	62.6 ± 2.4	41.6 ± 3.5	51.9 ± 2.1
		P-IQL	70.6 ± 4.1	16.2 ± 5.4	69.0 ± 6.2	71.1 ± 2.3	49.6 ± 3.4	60.6 ± 3.6
		CPL	80.0 ± 2.5	24.5 ± 2.1	80.0 ± 6.8	83.6 ± 1.6	61.1 ± 3.0	70.4 ± 3.0
Image	2.5k Dense	SFT	74.7 ± 4.8	20.8 ± 2.4	62.9 ± 2.3	64.5 ± 7.6	44.5 ± 3.2	52.5 ± 2.5
		P-IQL	83.7 ± 0.4	22.1 ± 0.8	68.0 ± 4.6	76.0 ± 4.6	51.2 ± 2.4	67.7 ± 4.4
		CPL	80.0 ± 4.9	27.5 ± 4.2	73.6 ± 6.9	80.3 ± 1.4	57.3 ± 5.9	68.3 ± 4.8
State	20k Sparse	SFT	67.0 ± 4.9	21.4 ± 2.7	63.6 ± 2.4	63.5 ± 0.9	41.9 ± 3.1	50.9 ± 3.2
		P-IQL	75.0 ± 3.3	19.5 ± 1.8	79.0 ± 6.6	76.2 ± 2.8	55.5 ± 4.2	73.4 ± 4.2
		CPL	83.2 ± 3.5	29.8 ± 1.8	77.9 ± 9.3	79.1 ± 5.0	56.4 ± 3.9	81.2 ± 1.6
Image	20k Sparse	SFT	71.5 ± 1.9	22.3 ± 2.9	65.2 ± 2.2	67.5 ± 1.1	41.3 ± 2.8	55.8 ± 2.9
		P-IQL	80.0 ± 2.3	27.2 ± 4.1	74.8 ± 5.8	80.3 ± 1.2	54.8 ± 5.8	72.5 ± 2.0
		CPL	78.5 ± 3.1	31.3 ± 1.6	70.2 ± 2.1	79.5 ± 1.4	61.0 ± 4.2	72.0 ± 1.8
Oracle	% BC	10%	62.6 ± 2.6	18.9 ± 1.7	57.5 ± 3.0	61.5 ± 3.7	39.1 ± 2.5	49.3 ± 2.1
		5%	64.6 ± 4.1	18.2 ± 0.6	59.8 ± 1.6	61.3 ± 1.8	38.6 ± 2.5	49.2 ± 1.9

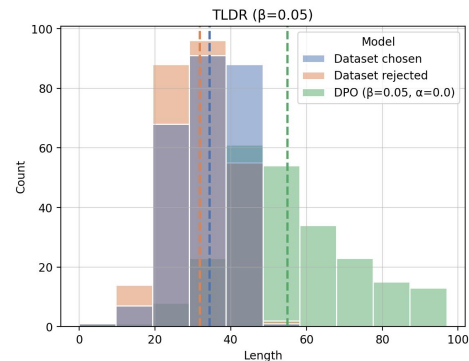
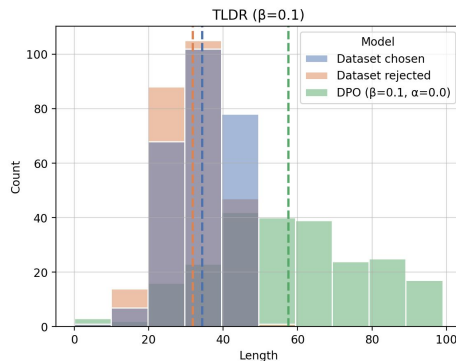
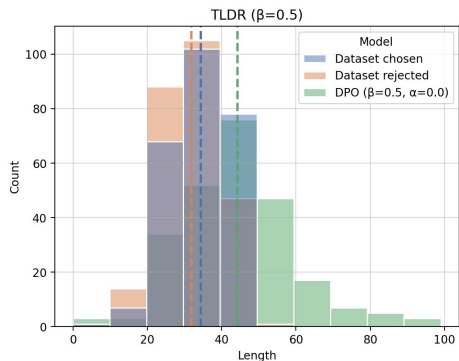
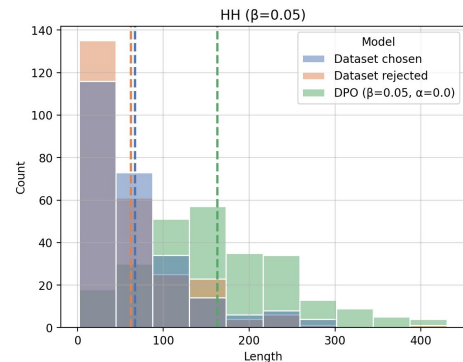
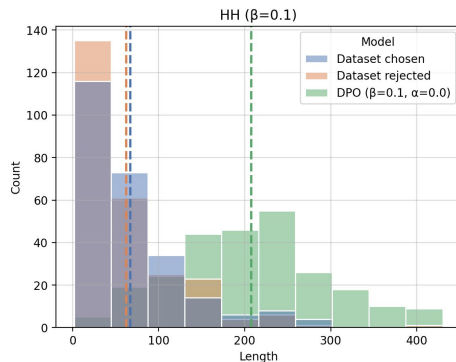
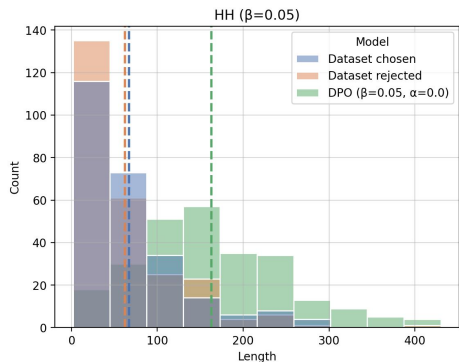
CPL

Contrastive Preference Learning: Learning from Human Feedback without RL, Hejna et. al.

Where do things go wrong?

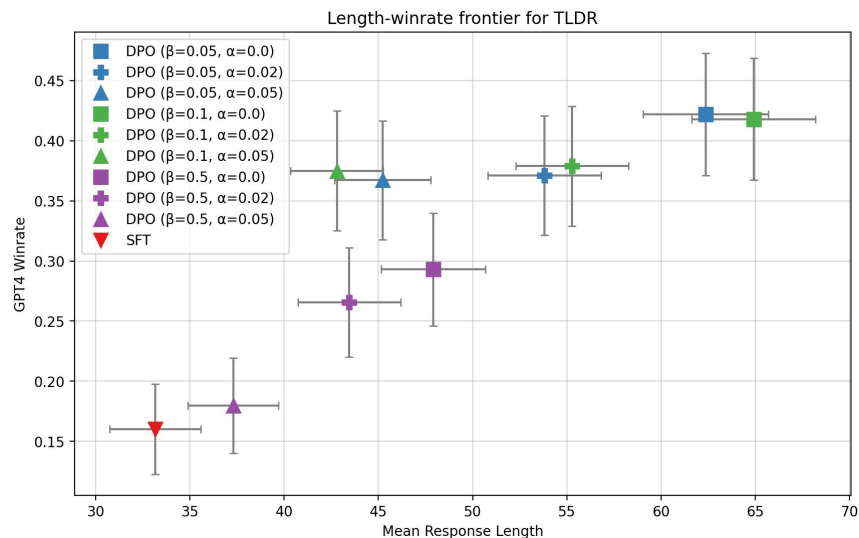
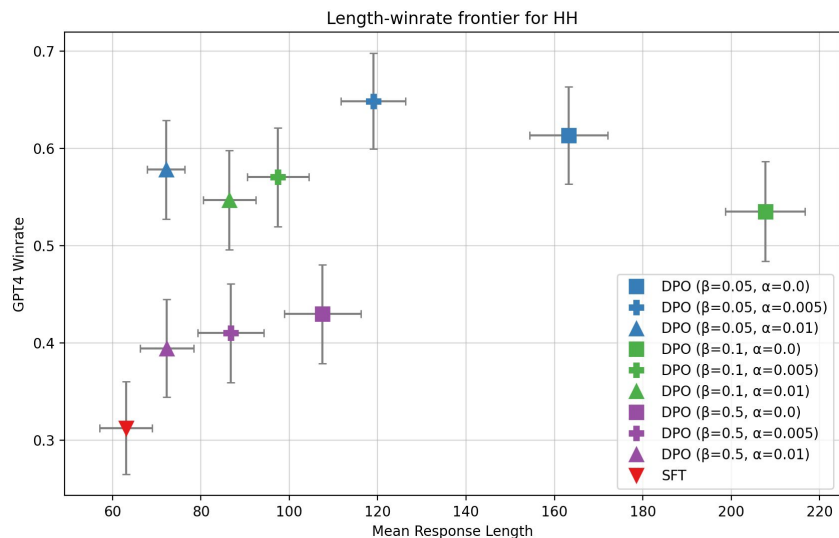


Where do things go wrong?



Disentangling Length from Quality in Direct Preference Optimization, Park et. al.

Where do things go wrong: Regularization



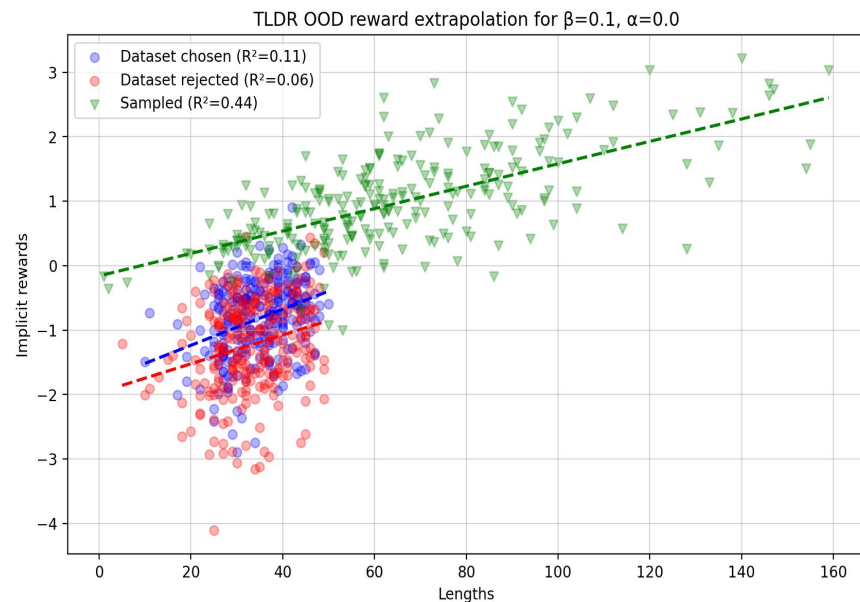
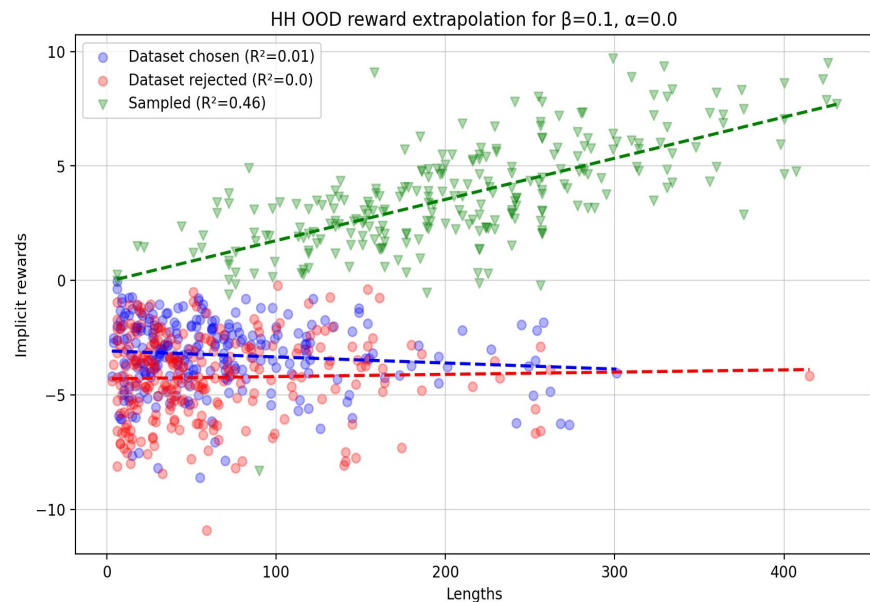
Disentangling Length from Quality in Direct Preference Optimization, Park et. al.

Stanford University

Where do things go wrong?

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

Where do things go wrong: OOD Robustness



Disentangling Length from Quality in Direct Preference Optimization, Park et. al.

Stanford University