UCLA Samueli School of Engineering

A Brief and Partial Summary of RLHF Algorithms

UCLA ScAi Lab Reading Group

Presenter: Yihe Deng

Why RLHF?

LLMs pre-trained on large text corpus express unintended behaviors such as hallucinations, bias/toxicity, or failure to follow instructions.

- Misaligned: language modeling objective (next token prediction) is different from the objective of human values.
  - Helpful, honest, harmless.

RLHF is proposed to align a model trained on general corpus to complex human values.

- Use human feedback for generated text as a measure of performance and use that feedback as a loss to optimize the model.
- Use methods from RL to directly optimize a language model with human feedback.

Training language models to follow instructions with human feedback. OpenAI.

Learning from Preference Feedback

Preference Reward Modeling with RL

- Requires building a reward model based on user preferences, optimized with RL, typically using the **PPO** (Proximal Policy Optimization) algorithm.
- Computationally expensive and sensitive to hyper-parameter tuning.

Direct Preference Optimization from Data

- Views preference optimization as offline RL, with implicit reward model optimization.
- Starting with **DPO** (Direct Preference Optimization), the evolution of variations of DPO aims in adjusting its loss function, with ongoing fixes that make it more RL-like.

https://zhuanlan.zhihu.com/p/1686790674?utm_psn=1833144248435879936

RLHF Pipeline

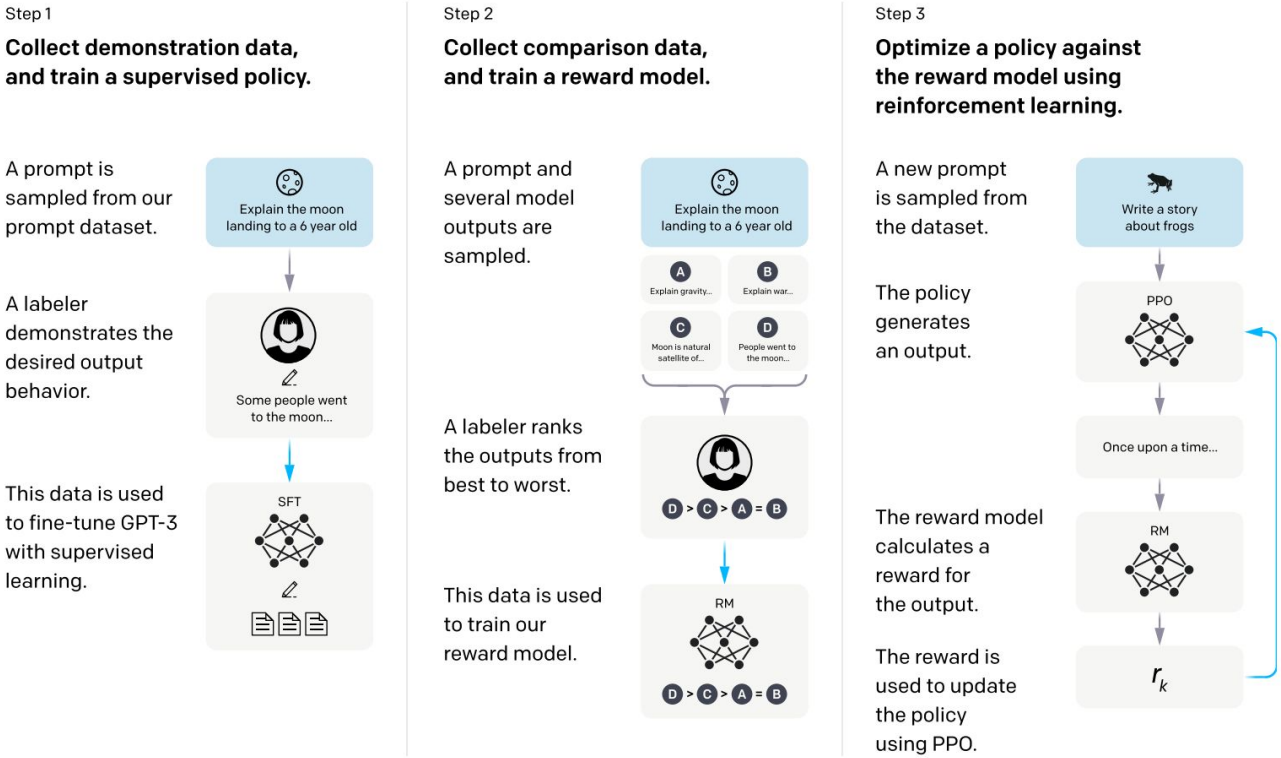
1. Fine-tuning (SFT) a language model (the initial model for RLHF).
   - Core to starting the RLHF process is having a model that responds well to diverse instructions.

2. Gathering data with human feedback and training a reward model.

3. Fine-tuning the LM with reinforcement learning.

Illustrating Reinforcement Learning from Human Feedback (RLHF). HuggingFace Blog.

RLHF Pipeline



UCLA Samueli School of Engineering

[Training language models to follow instructions with human feedback.](#) OpenAI.

RLHF Pipeline

2. Gathering data with human feedback and training a reward model.

Get a model that takes in a sequence of text, and returns a **scalar reward** which numerically represent the human preference.

Assumption: the preference signal can be modeled using the reward-based Bradley-Terry model.

Definition 2 (Bradley-Terry Model). *There exists a ground-truth reward function r^* and the preference model satisfies:*

$$\mathbb{P}(a^1 \succ a^2 | x, a^1, a^2) = \frac{\exp(r^*(x, a^1))}{\exp(r^*(x, a^1)) + \exp(r^*(x, a^2))} = \sigma\big(r^*(x, a^1) - r^*(x, a^2)\big), \quad (1)$$

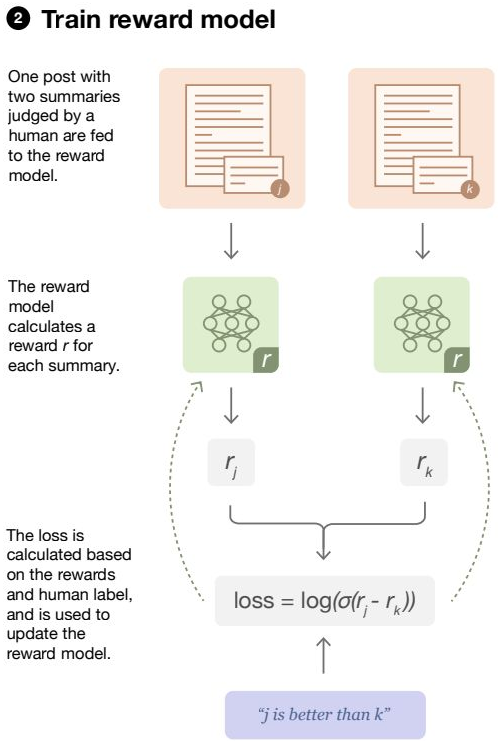
where $\sigma(z) = 1/(1 + \exp(-z))$ is the sigmoid function.

preferred response

dispreferred response

RLHF Workflow: From Reward Modeling to Online RLHF. Dong et al.

RLHF Pipeline



Bradley-Terry Model

$$\mathbb{P}(a^1 \succ a^2 | x, a^1, a^2) = \frac{\exp(r^*(x, a^1))}{\exp(r^*(x, a^1)) + \exp(r^*(x, a^2))}$$

Preference dataset: $\mathcal{D}_{\text{off}} = \{(x, a^w, a^l)\}$

(Maximize) log likelihood of the BT Model:

$$\ell_{\mathcal{D}_{\text{off}}}(\theta) = \sum_{(x,a^w,a^l,y)\in\mathcal{D}_{\text{off}}} \log\Big(\sigma\big(r_\theta(x, a^w) - r_\theta(x, a^l)\big)\Big).$$

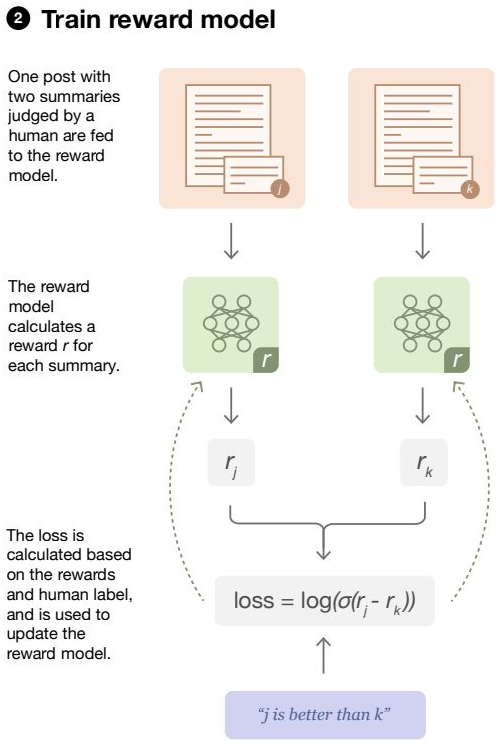
⇒ the maximum likelihood estimator (MLE)

+ Regularization with KL divergence:

$$\hat{r}(x, a) = r_{\text{MLE}}(x, a) - \eta \log \frac{\pi(a|x)}{\pi_0(a|x)}.$$

Learning to summarize from human feedback. OpenAI.
RLHF Workflow: From Reward Modeling to Online RLHF. Dong et al.

RLHF Pipeline



+ Regularization with KL divergence:

$$\hat{r}(x,a) = r_{\text{MLE}}(x,a) - \eta \log \frac{\pi(a|x)}{\pi_0(a|x)}.$$

Per-token probability distributions from the RL policy are compared to the ones from the initial model

- penalizes the RL policy from moving substantially away from the initial pretrained model with each training batch.

UCLA Samueli School of Engineering

Learning to summarize from human feedback. OpenAI.
RLHF Workflow: From Reward Modeling to Online RLHF. Dong et al.

Fine-tuning LLM with RL

We consider human interaction as the “environment”.

At each timestep t, the agent (LLM) receives a **state** *s_t* from the environment (i.e., the dialogue history)

- Th **environment** consists of all the dialogue text up to this point, both by the assistant and the human.
- Based on its **policy** π, the agent’s action at is to generate the next token.
- The environment returns a **reward** *r(s_t, a_t)*, which is calculated from a reward function *r* trained from human preference data.
- The agent then transitions to the next **state** *s_{t+1}*, which includes the next dialogue history.

The aim of RL is to find an optimal behavior strategy for the agent to maximize the cumulative reward (i.e., **return**) over a trajectory $\tau = \{s_1, a_1, \ldots, s_T, a_T\}$.

$$R(\tau) = \sum_{t=1}^{T'} r(s_t, a_t)$$

Fine-tuning LLM with RL: Policy Gradient

The policy π is typically parameterized by θ, we denote it as *π(a|s, θ)*, which is the probability of taking action a in state s.

$$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta),$$

J(θ) represents the expected return when following policy πθ

- ∇J(θ) is the policy gradient

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \Phi_t \right],$$

where Φ_t could be any of the following that leads to the same expected value for the policy gradient

1. $\Phi_t = R(\tau)$
2. $\Phi_t = \sum_{t'=t}^{T} R(s_{t'}, a_{t'})$
3. $\Phi_t = \sum_{t'=t}^{T} R(s_{t'}, a_{t'}) - b(s_t)$ ← subtract by baseline to reduce variance

Fine-tuning LLM with RL: Policy Gradient

If the return is favorable, all actions are “reinforced” by increasing their probability of being selected.

To reduce variance, a common strategy is to use advantage function estimates in place of raw returns in the policy gradient update rule.

- The advantage function *A(s_t, a_t)* represents how much better it is to take a specific action at at state *s_t*, compared to the average quality of actions at that state under the same policy.

$$\Phi_t = A(s_t, a_t). \qquad A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$$

where

- *Q(s_t, a_t)* is the action-value function, representing the expected return after taking action at at state s,
- *V(s_t)* is the value function, representing the average expected return at state st.

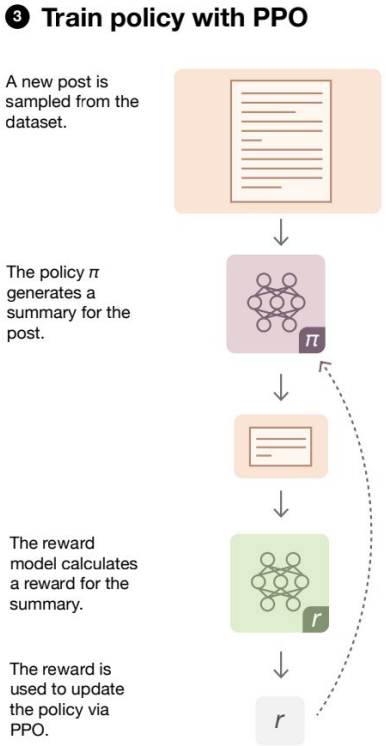
Fine-tuning LLM with RL: Policy Gradient

More content:

- Generalized Advantage Estimation https://arxiv.org/pdf/1506.02438
- [Blog] Policy Gradient Algorithms https://lilianweng.github.io/posts/2018-04-08-policy-gradient/

(TRPO and) PPO: pivotal techniques in RL, aimed at effectively training a policy without jeopardizing its stability.

Proximal Policy Optimization (PPO)



PPO wants to improve the training stability of the policy by limiting the change made to the policy at each training epoch.

- Measure how much the current policy changed compared to the former one using a ratio between the current and former policy.

$$r(\theta) = \frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)}$$

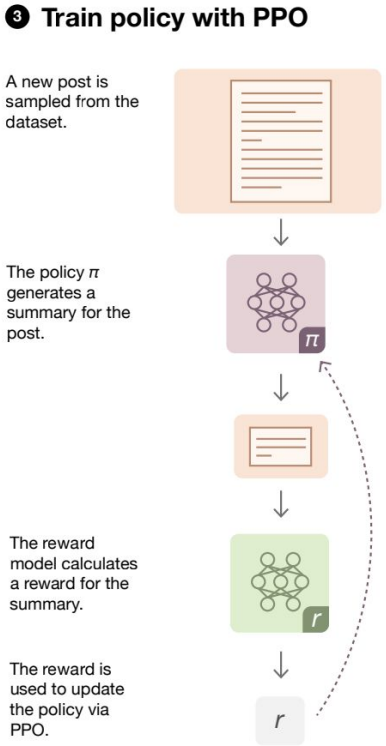
the probability ratio between old and new policies.

- Clip this ratio in a range.

Smaller policy updates during training are more likely to converge to an optimal solution

Proximal Policy Optimization (PPO). HuggingFace Blog.

Proximal Policy Optimization (PPO)



Trust Region Policy Optimization (TRPO):

- Hard constraint on the KL divergence.

$$\text{maximize}_\theta \ \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \hat{A}_t \right],$$
$$\text{subject to } \hat{\mathbb{E}}_t \left[\text{KL}(\pi_{\theta_{\text{old}}}(\cdot|s_t), \pi_\theta(\cdot|s_t)) \right] \leq \delta,$$

PPO: unconstrained optimization problem (a penalty-based approach)

$$\mathcal{L}_{\text{ppo-penalty}}(\theta) = \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \hat{A}_t \right] - \beta \text{KL}(\pi_{\theta_{\text{old}}}(\cdot|s_t), \pi_\theta(\cdot|s_t)),$$

Secrets of RLHF in Large Language Models Part I: PPO. Zheng et al.

Proximal Policy Optimization (PPO)

PPO: unconstrained optimization problem (a penalty-based approach)

+ ***Clipped Surrogate Objective.***

  (a clipped version of the policy ratio in its objective)

$$\mathcal{L}_{\text{ppo-clip}}(\theta) = \hat{\mathbb{E}}_t \left[\min \left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \hat{A}_t, \text{clip} \left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}, 1-\epsilon, 1+\epsilon \right) \hat{A}_t \right) \right],$$

Take the minimum of the clipped and non-clipped objective

- The final objective is a lower bound (pessimistic bound) of the unclipped objective.

Secrets of RLHF in Large Language Models Part I: PPO. Zheng et al.

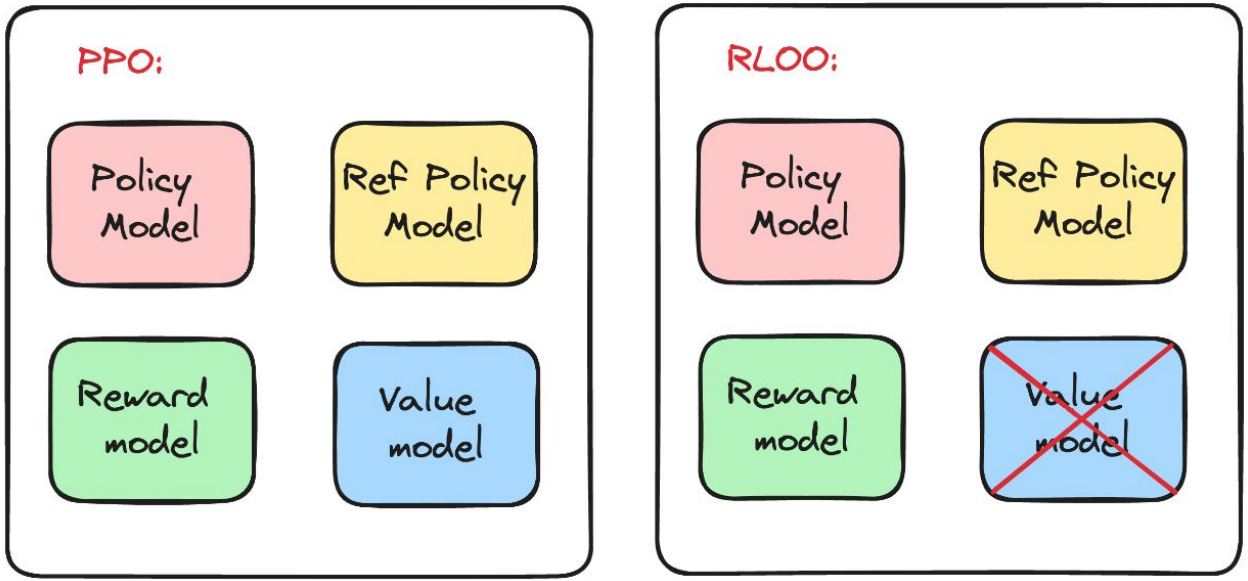
Challenges of PPO

Computationally intensive and sensitive hyperparameter tuning.

- PPO requires loading up to 4 models simultaneously: policy model, reference model, reward model, value (critic) model.
  - Removing value model: RLOO, GRPO, ect.
- Reward model training is interleaved with the training of the policy model (LLM).
  - Offline RL (removing explicit reward model training): DPO and its variants.
- The unstable and sensitive nature of online RL optimization, and the relative algorithmic complexity of PPO requires expertise to tune it well.

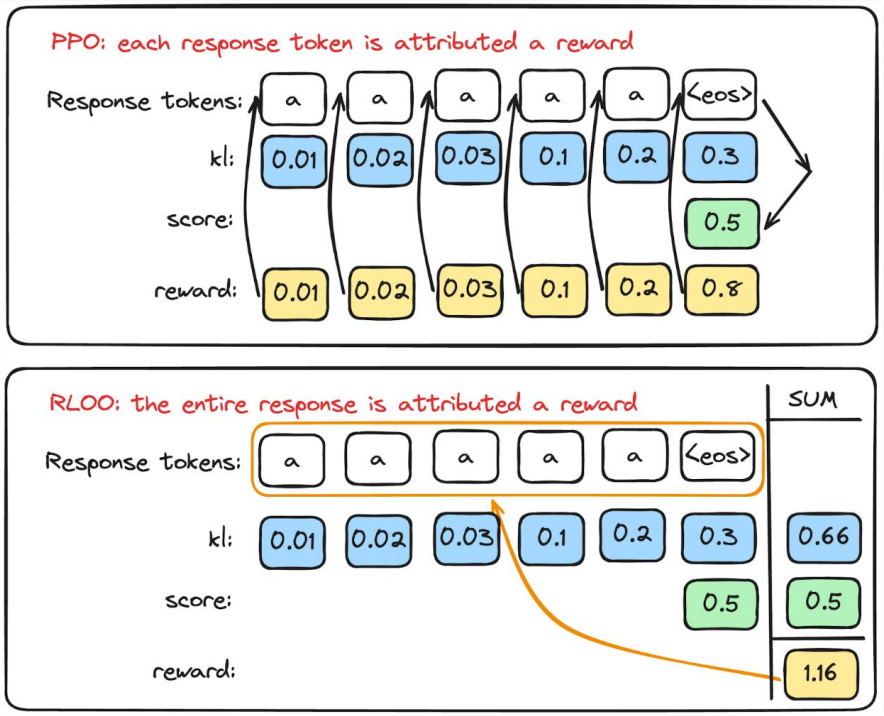
Back to Basics: Revisiting REINFORCE Style Optimization for Learning from Human Feedback in LLMs. Cohere For AI. 2024.

RLOO (REINFORCE Leave One-Out)



Back to Basics: Revisiting REINFORCE Style Optimization for Learning from Human Feedback in LLMs. Cohere For AI. 2024.

RLOO (REINFORCE Leave One-Out)



PPO:

- Initial state is determined by the prompt
- Each generated token is modeled as an action
- Partial sequences are seen as states.

RLOO:

- Only generating the <EOS> token carries a reward as output by the reward model which is combined with KL penalty.

Back to Basics: Revisiting REINFORCE Style Optimization for Learning from Human Feedback in LLMs. Cohere For AI. 2024.

RLOO (REINFORCE Leave One-Out)

Policy gradient

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t) \Phi_t \right],$$

The choice of Φ_t:

$$\frac{1}{k} \sum_{i=1}^{k} \left[R(y_{(i)}, x) - \frac{1}{k-1} \sum_{j \neq i} R(y_{(j)}, x) \right] \nabla \log \pi(y_{(i)}|x) \text{ for } y_{(1)}, ..., y_{(k)} \overset{i.i.d}{\sim} \pi_\theta(.|x)$$

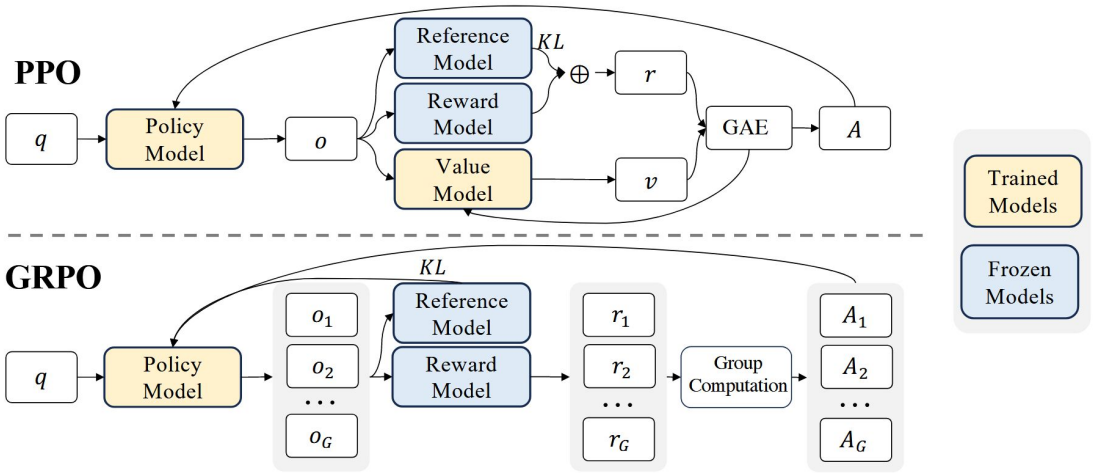
k refers to the number of online samples generated,

- For each prompt x, RLOO considers each y(i) individually and uses the remaining k – 1 samples to create an unbiased estimate of the expected return
- Similar to a parameter-free value-function, but estimated at each training step

GRPO (Group Relative Policy Optimization)

Similarly, GRPO removes the value/critic model

- Instead estimate the baseline from group scores, significantly reducing training resources compared to PPO.



DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. DeepSeek-AI. 2024.

GRPO (Group Relative Policy Optimization)

Recall PPO

$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{old}}(o_t|q,o_{<t})} A_t, \text{clip}\left(\frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{old}}(o_t|q,o_{<t})}, 1-\varepsilon, 1+\varepsilon \right) A_t \right], \quad (1)$$

Instead of the value function approximation (advantage A_t computed by applying GAE), GRPO uses the average reward of multiple sampled outputs, produced in response to the same question, as the baseline A_{i, t}

$$\hat{A}_{i,t} = \tilde{r}_i = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}$$

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)]$$

$$\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left\{ \min \left[\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})} \hat{A}_{i,t}, \text{clip}\left(\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}, 1-\varepsilon, 1+\varepsilon \right) \hat{A}_{i,t} \right] - \beta \mathbb{D}_{KL}\left[\pi_\theta || \pi_{ref}\right] \right\},$$

Instead of adding KL in the reward, GRPO directly adds KL between trained policy and reference policy to the loss.

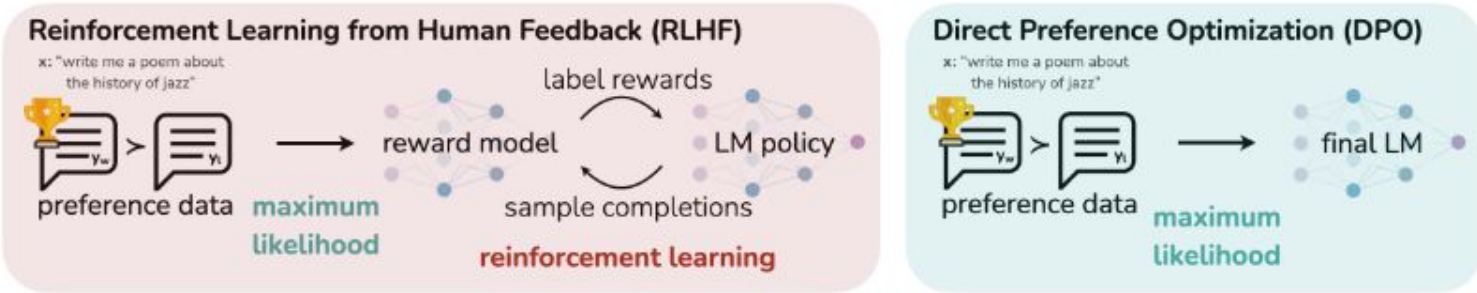
DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. DeepSeek-AI. 2024.

Challenges of Online RL

Computationally intensive

- Need to sample from the policy model in the loop of training. (LLM generation during training is very expensive)

⇒ Offline RL (DPO)



Direct Preference Optimization: Your Language Model is Secretly a Reward Model. Rafailov et al. 2022.

DPO: Direct Preference Optimization

DPO loss:

$$\log \sigma\big(r_\theta(x, a^w) - r_\theta(x, a^l)\big), \quad \hat{r}_\theta(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$$

Bradley-Terry Model

Reward implicitly defined by the language model and reference model

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

Intuitively, the gradient of DPO loss

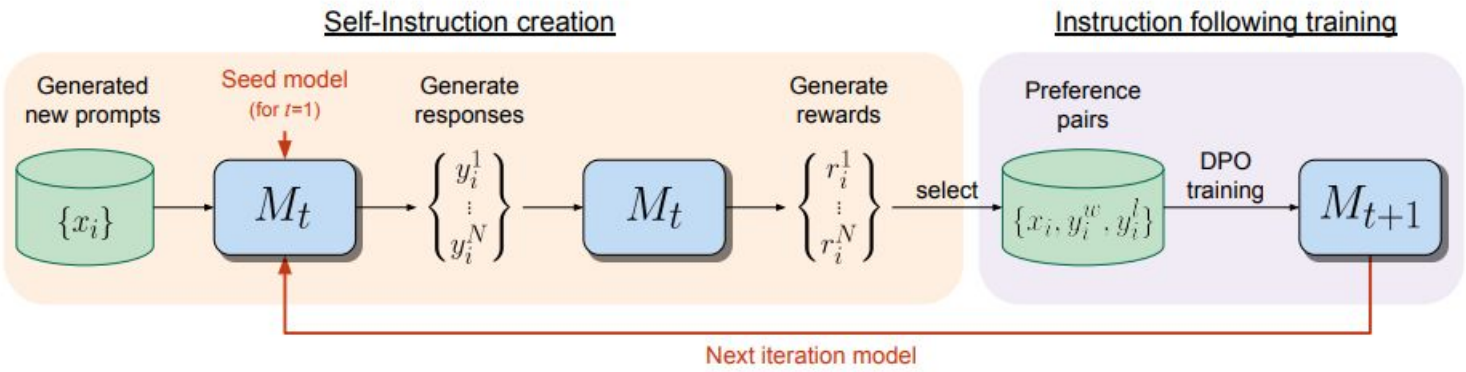
- increases the likelihood of the preferred completions y_w
- decreases the likelihood of dispreferred completions y_l.

Iterative DPO

A version of DPO in between online and offline RL.

⇒ What current self-training methods are doing: iterative generation of preference data.

- Self-Reward
- SPIN
- Iterative Reasoning Preference Optimization
- ReST-MCTS*



Iterative DPO

A version of DPO in between online and offline RL.

⇒ Update the reference model during DPO training across iterations

- SPIN
- Iterative Reasoning Preference Optimization
- Building Math Agents with Multi-Turn Iterative Preference Learning.

$$\text{Update } \boldsymbol{\theta}_{t+1} = \operatorname{argmin}_{\boldsymbol{\theta}\in\boldsymbol{\Theta}} \sum_{i\in[N]} \ell\left(\lambda \log \frac{p_{\boldsymbol{\theta}}(\mathbf{y}_i|\mathbf{x}_i)}{p_{\boldsymbol{\theta}_t}(\mathbf{y}_i|\mathbf{x}_i)} - \lambda \log \frac{p_{\boldsymbol{\theta}}(\mathbf{y}'_i|\mathbf{x}_i)}{p_{\boldsymbol{\theta}_t}(\mathbf{y}'_i|\mathbf{x}_i)}\right).$$

Iteratively update the reference model.

Iterative DPO

A version of DPO in between online and offline RL.

⇒ Update the reference model during DPO training

- SPIN
- Iterative Reasoning Preference Optimization
- Building Math Agents with Multi-Turn Iterative Preference Learning.

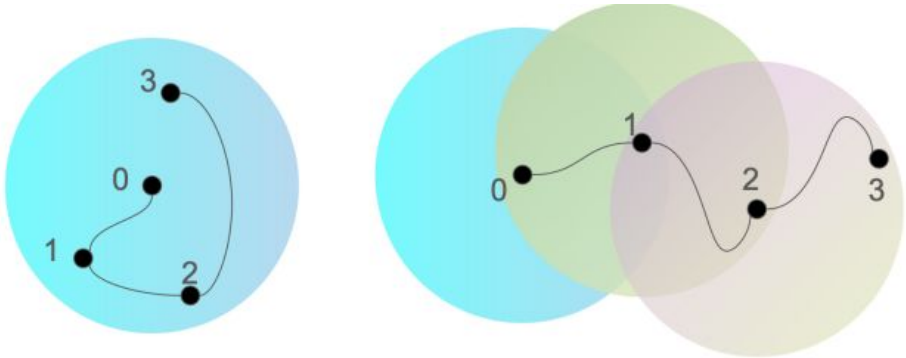


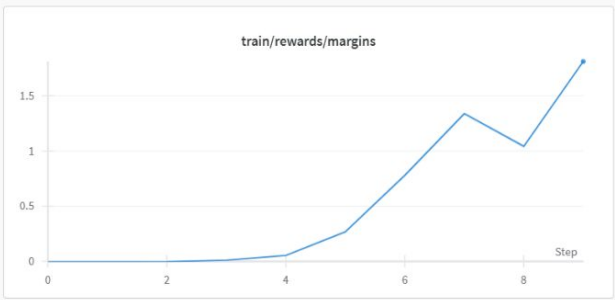
Figure 2 | Illustration of the difference between the two learning objectives. The left-hand figure corresponds to the KL-regularized target where we do not update the reference model. The right-hand figure corresponds to the non-regularized target where we always update the reference model as the last-iteration one.

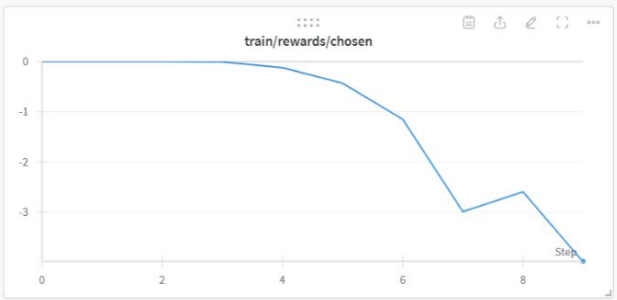
RPO (DPO + NLL loss)

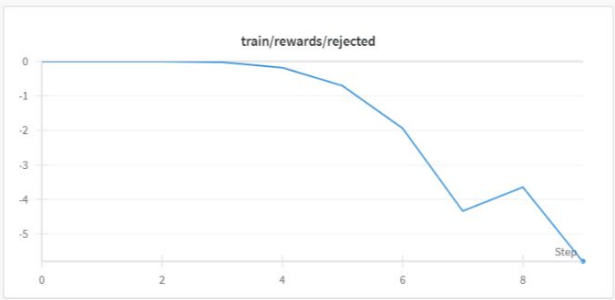
A common phenomenon when training with DPO loss:

- It maximizes the margin of the rewards
- But the chosen/preferred reward decreases to negative as well.

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







RPO (DPO + NLL loss)

A strategy to resolve is to add an additional NLL loss to the chosen/preferred sample.

$$\mathcal{L}_{\text{DPO+NLL}} = \mathcal{L}_{\text{NLL}}(x_i, c_i^w, y_i^w) + \alpha \mathcal{L}_{\text{DPO}}(c_i^w, y_i^w, c_i^l, y_i^l | x_i)$$
$$= \boxed{-\frac{\log M_\theta(x_i, c_i^w, y_i^w)}{|x_i| + |c_i^w| + |y_i^w|}} - \alpha \log \sigma \left(\beta \frac{\log M_\theta(c_i^w, y_i^w | x_i)}{\log M_{\text{t}}(c_i^w, y_i^w | x_i)} - \beta \frac{\log M_\theta(c_i^l, y_i^l | x_i)}{\log M_{\text{t}}(c_i^l, y_i^l | x_i)} \right).$$

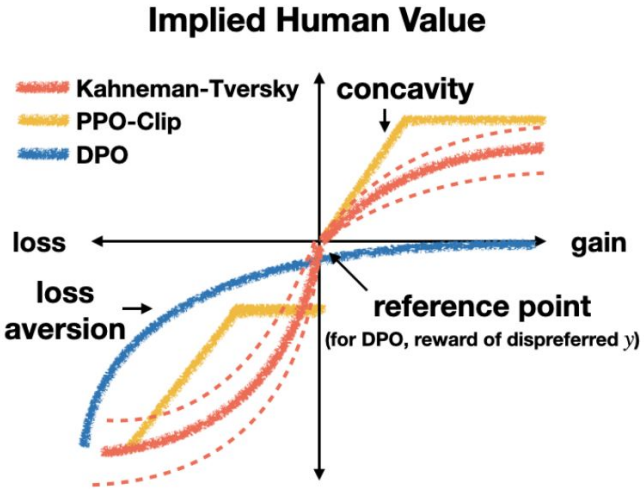
A mixed objective between SFT and DPO.

Iterative Reasoning Preference Optimization. Pang et al. 2024.
Enhancing Large Vision Language Models with Self-Training on Image Comprehension. Deng et al. 2024.

KTO: Beyond Bradley-Terry Model

Humans make decisions about uncertain events that do not maximize their expected value

- Relative to some reference point, humans are more sensitive to losses than gains, a property called **loss aversion**



UCLA Samueli School of Engineering

KTO: Model Alignment as Prospect Theoretic Optimization. Ethayarajh et al. 2024.

KTO: Beyond Bradley-Terry Model

BT model assumed that human preference can be captured with the specific function class that outputs a numerical value.

Kahneman-Tversky model of human utility

- Human-aware losses (HALOs) that directly maximizes the utility of generations instead of maximizing the log-likelihood of preferences.

Where λ_y denotes $\lambda_D(\lambda_U)$ when y is desirable(undesirable) respectively, the default KTO loss is:[6]

$$L_{\text{KTO}}(\pi_\theta, \pi_{\text{ref}}) = \mathbb{E}_{x,y\sim D}[\lambda_y - v(x,y)] \tag{8}$$

where

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

$$z_0 = \text{KL}(\pi_\theta(y'|x) \| \pi_{\text{ref}}(y'|x))$$

$$v(x,y) = \begin{cases} \lambda_D \sigma(\beta(r_\theta(x,y) - z_0)) \text{ if } y \sim y_{\text{desirable}}|x \\ \lambda_U \sigma(\beta(z_0 - r_\theta(x,y))) \text{ if } y \sim y_{\text{undesirable}}|x \end{cases}$$

UCLA Samueli School of Engineering

KTO: Model Alignment as Prospect Theoretic Optimization. Ethayarajh et al. 2024.

DNO: Beyond Bradley-Terry Model

Direct Nash Optimization

For Bradley-Terry model, the reward maximization approach is limited by the nature of "*point-wise*" rewards (scalar score for a single response to input x), which fails to express complex intransitive or cyclic preference relations.

- Re-frame RLHF as finding a Nash equilibrium of a two-player game from a general preference function.

$$\pi_{t+1} \leftarrow \underset{\pi \in \Pi}{\operatorname{argmax}}\, \mathbb{E}_{(x,y_1,y_2)\sim\mathcal{D}_t} \Bigg\{ \sigma\left(r_t(x,y_1) - r_t(x,y_2)\right) \log \left[\sigma\left(\eta \log \frac{\pi(y_1 \mid x)}{\pi_t(y_1 \mid x)} - \eta \log \frac{\pi(y_2 \mid x)}{\pi_t(y_2 \mid x)} \right) \right] + \sigma\left(r_t(x,y_2) - r_t(x,y_1)\right) \log \left[\sigma\left(\eta \log \frac{\pi(y_2 \mid x)}{\pi_t(y_2 \mid x)} - \eta \log \frac{\pi(y_1 \mid x)}{\pi_t(y_1 \mid x)} \right) \right] \Bigg\},$$

$$r_t(x,y) \leftarrow \mathbb{E}_{y' \sim \pi_t(\cdot|x)} \left[\mathcal{P}(y \succ y' \mid x) \right]$$

UCLA Samueli School of Engineering

Direct Nash Optimization: Teaching Language Models to Self-Improve with General Preferences. Rosset et al. 2024.

Step-level DPO and Process Reward

PPO: token-level; DPO: response-level

⇒ More granular reward for DPO? (Reasoning) Step-level

Let's Verify Step-by-Step (OpenAI): process supervision/reward at each reasoning step of the model's generation is better than outcome (response) supervision.

Step-level DPO:

- Step-DPO: Step-wise Preference Optimization for Long-chain Reasoning of LLMs. Lai et al. 2024.
- ReST-MCTS*: LLM Self-Training via Process Reward Guided Tree Search. Zhang et al. 2024.
- Flow-DPO: Improving LLM Mathematical Reasoning through Online Multi-Agent Learning. Deng et al. 2024.
- Monte Carlo Tree Search Boosts Reasoning via Iterative Preference Learning. Xie et al. 2024.

Step-DPO and Process Reward

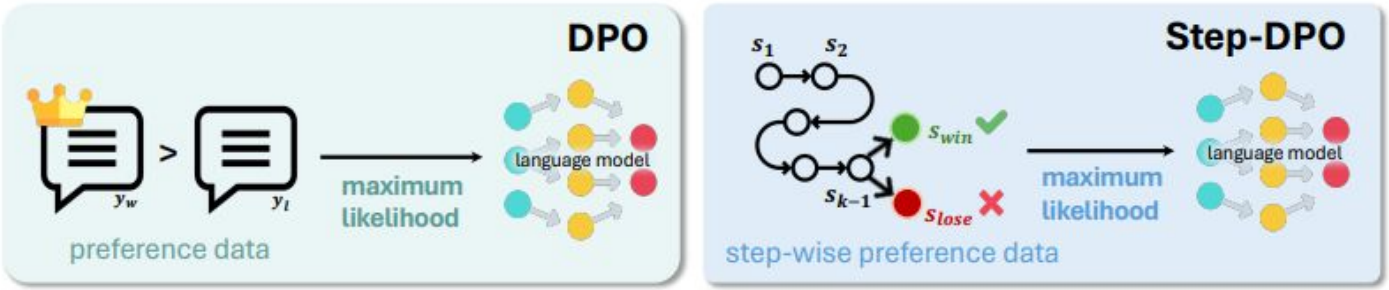
PPO: token-level

DPO: response-level

⇒ More granular reward for DPO? (Reasoning) Step-level

Let's Verify Step-by-Step (OpenAI): process supervision/reward at each reasoning step of the model's generation is better than outcome (response) supervision.

Step-level DPO:



Step-DPO and Process Reward

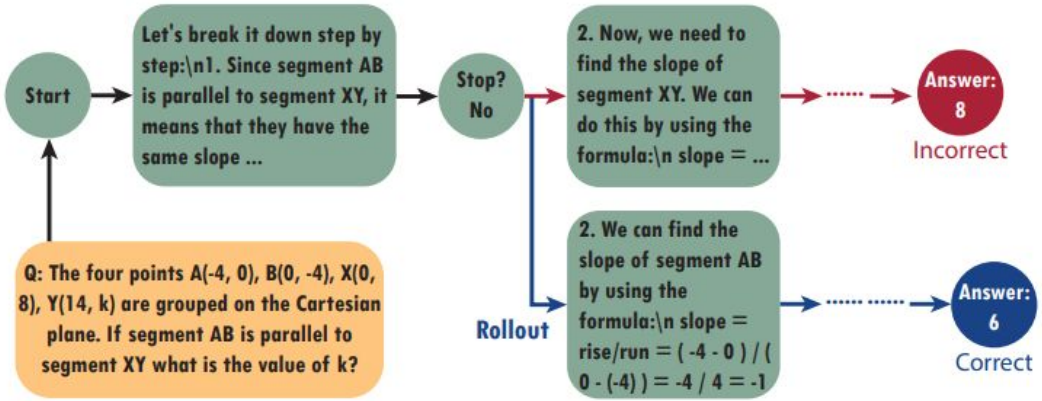
PPO: token-level

DPO: response-level

⇒ More granular reward for DPO? (Reasoning) Step-level

Let's Verify Step-by-Step (OpenAI): process supervision/reward at each reasoning step of the model's generation is better than outcome (response) supervision.

Step-level DPO:



R-DPO and SimPO: Bias of Length

Human generally prefer more detailed response.

- DPO tends to encourage the model responses to be longer as compared to better if such bias is in the preference data.

Dataset	Preferred Length			Dispreferred Length		
	Mean	Median	Std.	Mean	Median	Std.
Anthropic RLHF HH	**79.6**	**57.0**	**74.0**	75.7	51.0	73.3
Reddit TL;DR	**37.9**	**36.0**	**13.9**	35.2	34.0	13.4

Disentangling Length from Quality in Direct Preference Optimization. Park et al. 2024.

R-DPO

$$\mathcal{L}_{\mathrm{R-DPO}}\left(\pi_\theta; \pi_{\mathrm{ref}}\right) =$$

$$-\mathbb{E}_{(\mathbf{x},\mathbf{y}_w,\mathbf{y}_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta\left(\mathbf{y}_w \mid \mathbf{y}\right)}{\pi_{\mathrm{ref}}\left(\mathbf{y}_w \mid \mathbf{x}\right)} - \beta\log\frac{\pi_\theta\left(\mathbf{y}_l \mid \mathbf{x}\right)}{\pi_{\mathrm{ref}}\left(\mathbf{y}_l \mid \mathbf{x}\right)} + \boxed{\left(\alpha|\mathbf{y}_w| - \alpha|\mathbf{y}_l|\right)}\right)\right]$$

An additional per-example learning rate, which up-weighs the pairs where preferred response is shorter and down-weights the reverse.

Disentangling Length from Quality in Direct Preference Optimization. Park et al. 2024.

SimPO

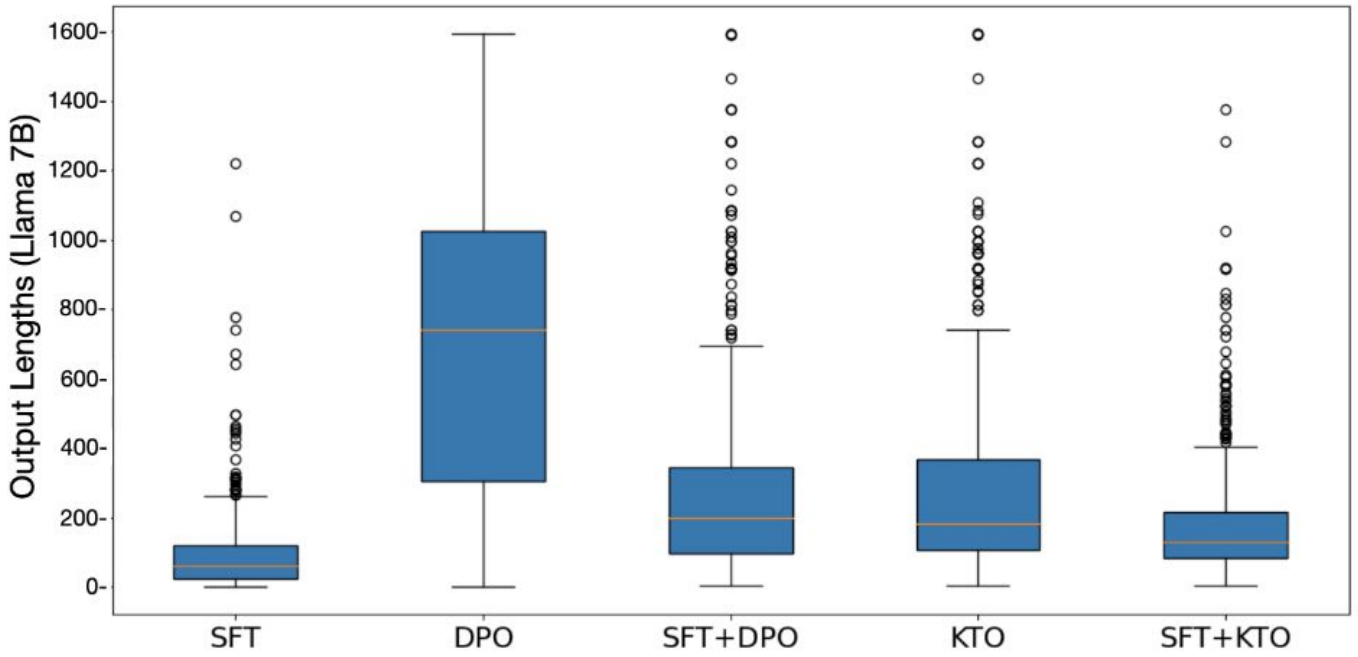
$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right]$$

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

Consider using the average log-likelihood as the implicit reward.
⇒ Regularization on preferred and dispreferred examples.

SimPO: Simple Preference Optimization with a Reference-Free Reward. Ment et al. 2024.

KTO on Output Length

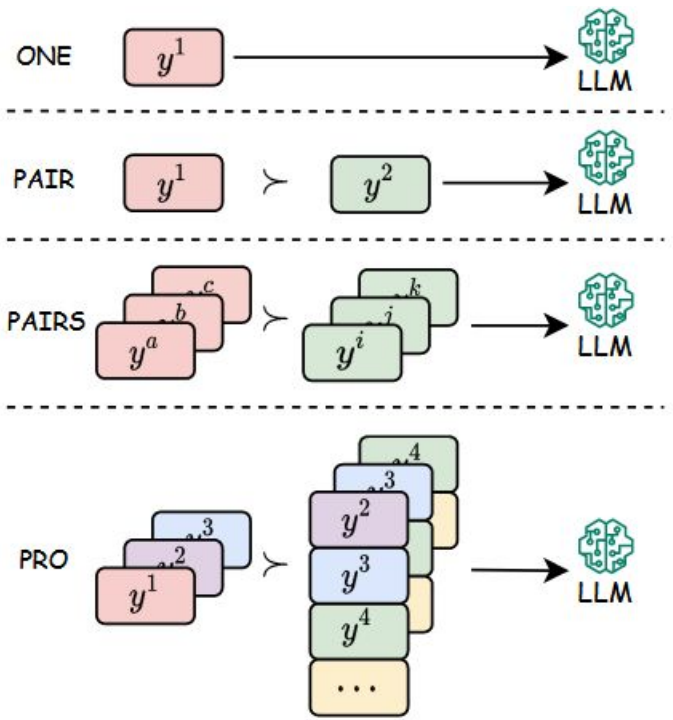


PRO: Beyond Pairwise Preference

Preference Ranking Optimization (PRO)

- An efficient SFT algorithm to directly fine-tune LLMs for human alignment.
- Extends the pairwise contrast to accommodate preference rankings.

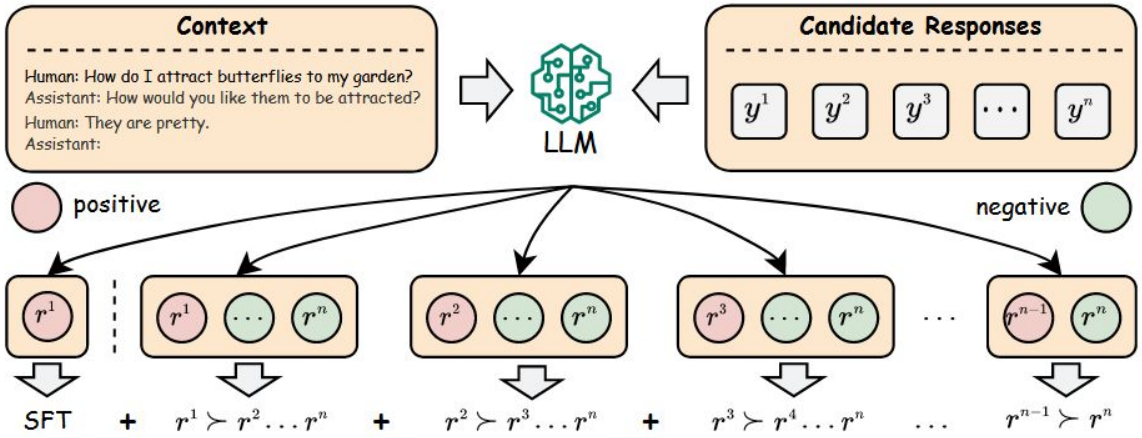
By iteratively contrasting candidates, PRO instructs the LLM to prioritize the best response while progressively ranking the rest responses.



Preference Ranking Optimization for Human Alignment. Song et al. 2023.

PRO: Beyond Pairwise Preference

$$\mathcal{L}_{\text{PRO}}(y^{1,\cdots,n} \mid x) = \mathcal{L} + \beta \mathcal{L}_{\text{SFT}} \qquad \mathcal{L} = -\log \prod_{k=1}^{n-1} \frac{\exp\left(r_\pi(x, y^k)\right)}{\sum_{i=k}^{n} \exp\left(r_\pi(x, y^i)\right)}$$



More: [LiPO: Listwise Preference Optimization through Learning-to-Rank.](#) Liu et al. 2024.

Preference Data Construction

Other than human/AI labeling? (Self-training)

RPO: Iterative Reasoning Preference Optimization

→ Math reasoning: correctness of the model final answer (**Outcome Reward Model**)

Sample K model generations with reasoning to a math question.

- Preferred: reasoning leading to correct answer
- Dispreferred: reasoning leading to incorrect answers

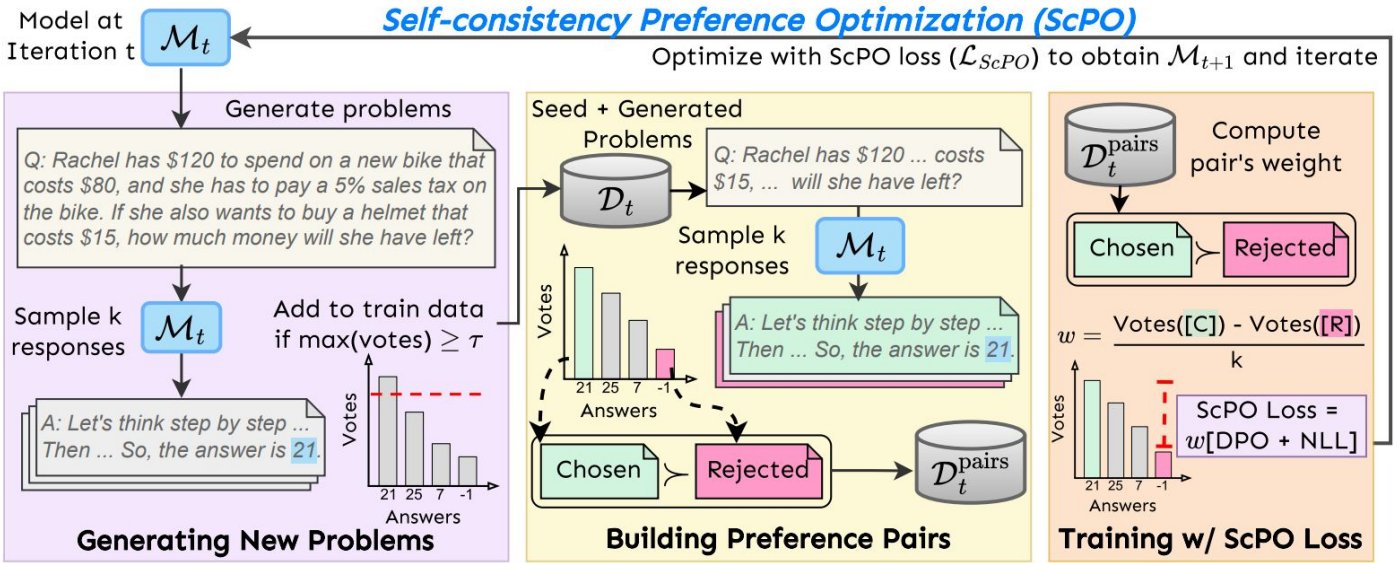
Similar papers (SFT):

- RFT: Scaling Relationship on Learning Mathematical Reasoning with Large Language Models
- ReST-EM: Beyond Human Data: Scaling Self-Training for Problem-Solving with Language Models

Iterative Reasoning Preference Optimization. Pang et al. 2024.

Preference Data Construction

ScPO: Self-Consistency Preference Optimization

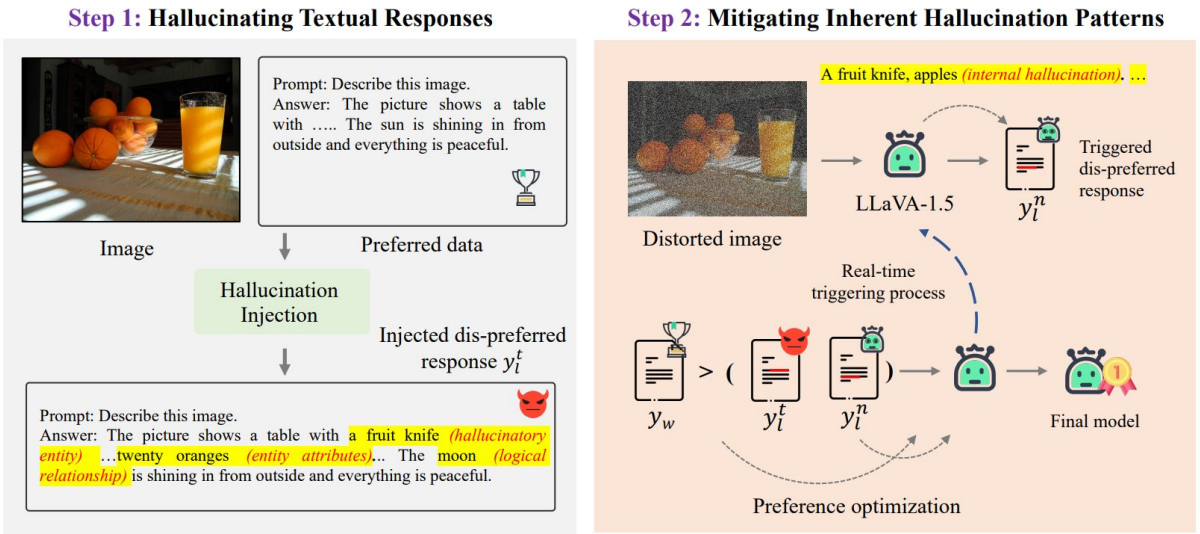


UCLA Samueli School of Engineering

Self-Consistency Preference Optimization. Prasad et al. 2024.

Preference Data Construction

(Multi-modal) **POVID: Preference Optimization in VLLM with AI-Generated Dispreferences**

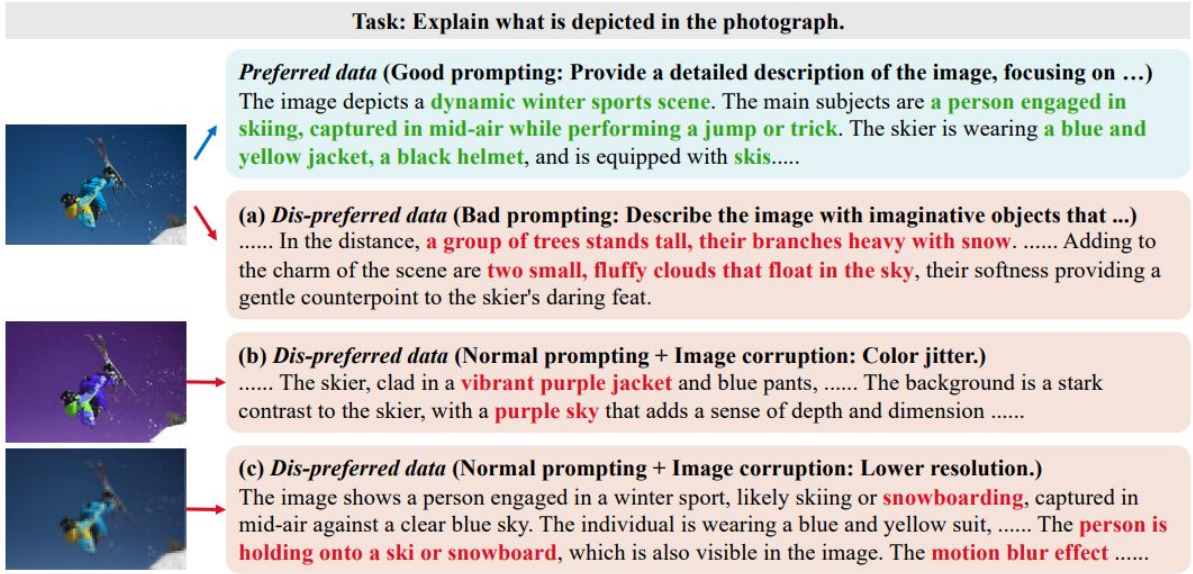


UCLA Samueli School of Engineering

Aligning modalities in vision large language models via preference fine-tuning. Zhou et al. 2024.

Preference Data Construction

(Multi-modal) **STIC: Self-Training on Image Comprehension**



Enhancing Large Vision Language Models with Self-Training on Image Comprehension. Deng et al. 2024.

More on Online/Offline RL

- Is DPO Superior to PPO for LLM Alignment? A Comprehensive Study
- Understanding the performance gap between online and offline alignment algorithms
- From r to Q*: Your Language Model is Secretly a Q-Function
- Agent Q: Advanced Reasoning and Learning for Autonomous AI Agents
- Foundational Autoraters: Taming Large Language Models for Better Automatic Evaluation

References / Useful Blogs

- https://lilianweng.github.io/posts/2018-04-08-policy-gradient/
- https://danieltakeshi.github.io/2017/03/28/going-deeper-into-reinforcement-learning-fundamentals-of-policy-gradients/
- https://huggingface.co/blog/rlhf
- https://huggingface.co/blog/deep-rl-ppo
- https://zhuanlan.zhihu.com/p/1686790674?utm_psn=1833144248435879936
- https://www.notion.so/eb7b2d1891f44b3a84e7396d19d39e6f?v=01bcb084210149488d730064cbabc99f

References / Papers

- Training language models to follow instructions with human feedback.
- RLHF Workflow: From Reward Modeling to Online RLHF.
- Secrets of RLHF in Large Language Models Part I: PPO.
- Back to Basics: Revisiting REINFORCE Style Optimization for Learning from Human Feedback in LLMs.
- DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models.
- Direct Preference Optimization: Your Language Model is Secretly a Reward Model.
- Iterative Reasoning Preference Optimization
- KTO: Model Alignment as Prospect Theoretic Optimization.
- Direct Nash Optimization: Teaching Language Models to Self-Improve with General Preferences.
- Step-DPO: Step-wise Preference Optimization for Long-chain Reasoning of LLMs

References / Papers

- Self-Rewarding Language Models
- Self-Play Fine-Tuning Converts Weak Language Models to Strong Language Models
- ReST-MCTS*: LLM Self-Training via Process Reward Guided Tree Search.
- Flow-DPO: Improving LLM Mathematical Reasoning through Online Multi-Agent Learning
- Monte Carlo Tree Search Boosts Reasoning via Iterative Preference Learning.
- Building Math Agents with Multi-Turn Iterative Preference Learning.
- Disentangling Length from Quality in Direct Preference Optimization.
- SimPO: Simple Preference Optimization with a Reference-Free Reward.
- Preference Ranking Optimization for Human Alignment.
- LiPO: Listwise Preference Optimization through Learning-to-Rank.
- Self-Consistency Preference Optimization.
- Aligning modalities in vision large language models via preference fine-tuning.
- Enhancing Large Vision Language Models with Self-Training on Image Comprehension.