Exploration of Direct Preference Optimization

ZHIHAO QIAN

Major Premise In this study, all machine learning processes are centered around a pre-trained large language model (LLM).

Key words direct preference optimization, RL-free alternative to RLHF, implicit reward modeling, different levels of feedback

1 Introduction

Explore an easy understanding of DPO through Q&A.

1.1 What is the task that DPO do?

Direct preference optimization (DPO) is a new method that helps large, unsupervised language models better match human preferences. DPO fine-tunes language models directly using human feedback with a simpler mechanism.

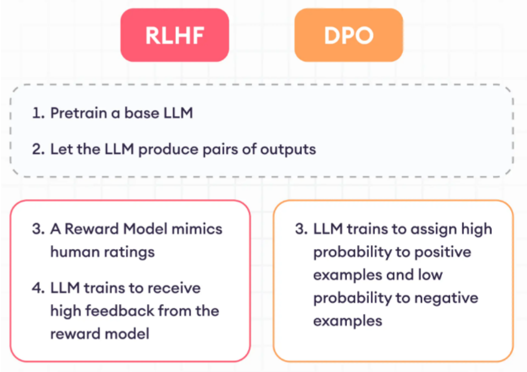
RLHF
DPO
1. Pretrain a base LLM
2. Let the LLM produce pairs of outputs
3. A Reward Model mimics human ratings
4. LLM trains to receive high feedback from the reward model
3. LLM trains to assign high probability to positive examples and low probability to negative examples

Fig. 1. RHLF and DPO, what's the difference?

1.2 Some Q&A about RHLF.

Previously, we adopted RHLF to address this task. The typical RLHF pipeline for LLMs generally consists of three key phases: (1) supervised fine-tuning (SFT), (2) preference sampling and reward model training, and (3)RL optimization.

1.2.1 How does RHLF align machine behavior with human preferences?

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(y|x)} \left[r_\phi(x,y)\right] - \beta\, \mathrm{D}_{\mathrm{KL}} \left[\pi_\theta(y|x) \,||\, \pi_{\mathrm{ref}}(y|x)\right] \tag{1}$$

There are two hierarchical levels of alignment: (1) **Inner level:** A reward model is trained to mimic human ratings–reward function r_ϕ. (2) **Outer level:** The Reward Function r_ϕ is used within a second-level Objective Function. The model is encouraged to generate responses that maximize r_1, while being penalized if its policy deviates excessively from the reference policy.

Author's Contact Information: Zhihao Qian, zhqian_1@stu.xidian.edu.cn.

1.2.2 How to train a model for reward function?

$$\mathcal{D} = \left\{ \left(x^{(i)}, y_u^{(i)}, y_l^{(i)} \right) \right\}_{i=1}^{N} \tag{2}$$

Given y_u the preferred response, and y_l the less preferred response, trainable reward model $r_\phi(y, x)$ is optimized via:

$$\mathcal{L}_R(r_\phi, \mathcal{D}) = -\mathbb{E}_{(x, y_u, y_l) \sim \mathcal{D}} \left[\log \sigma \left(r_\phi(x, y_u) - r_\phi(x, y_l) \right) \right] \tag{3}$$

where σ is the sigmoid function, which maps the reward difference to a probability in the range $[0, 1]$. Minimizing $\mathcal{L}_R$ maximizes the likelihood of correctly ranking responses per human preference.

1.2.3 What is the origin and technical meaning of the reference strategy ($\pi_{ref}(y \mid x)$)? The reference policy, π_{ref}, denotes the output probability distribution generated by a pre-trained or supervised fine-tuned model, and serves as a behavioral baseline during Reinforcement Learning from Human Feedback (RLHF).

1.2.4 Why use KL divergence? (1) the output is a probability distribution. (2) the KL divergence term penalizes excessive deviations from this baseline. The model improves its reward while maintaining safe and stable outputs.

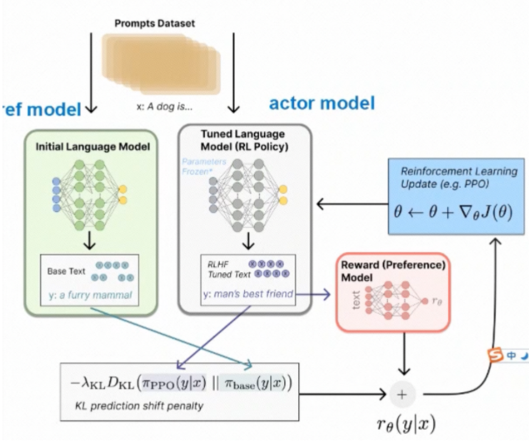
Prompts Dataset
ef model
x: A dog is...
actor model
Initial Language Model
Tuned Language Model (RL Policy)
Parameters Frozen*
Reinforcement Learning Update (e.g. PPO)
$\theta \leftarrow \theta + \nabla_\theta J(\theta)$
Base Text
y: a furry mammal
RLHF Tuned Text
y: man's best friend
Reward (Preference) Model
text
r_θ
$-\lambda_{\text{KL}} D_{\text{KL}}(\pi_{\text{PPO}}(y|x) \,||\, \pi_{\text{base}}(y|x))$
KL prediction shift penalty
+
$r_\theta(y|x)$
中

Fig. 2. Step3: Reinforcement Learning reward model using the PPO reinforcement learning algorithm.

1.2.5 Why is the RLHF process classified as Reinforcement Learning? In the workflow shown in Fig. 2, Reinforcement Learning from Human Feedback (RLHF) consists of key steps that closely align with the canonical reinforcement learning (RL) paradigm:

1. **Policy:** The "Tuned Language Model (RL Policy)" acts as the policy π_θ, which determines how to generate output y given an input x.
2. **Reward:** The "Reward Model" functions as the environment's feedback mechanism, assigning a scalar reward $r(y \mid x)$ to each model output.
3. **Policy Update:** Algorithms such as Proximal Policy Optimization (PPO) are used to adjust and optimize the policy based on the received rewards, enabling the model to generate outputs that obtain higher rewards. The parameter update rule $\theta \longleftarrow \theta + \nabla_\theta J(\theta)$ is a standard policy gradient step in RL.

(4) **Environment:** The environment in this context consists of the prompt combined with the reward model. After each model output, the reward model provides a score, serving as the environment's feedback.

1.2.6 What is reward hacking? Where is reward hacking used in RLHF? Why is reward hacking problematic? (1) Reward hacking is a long-standing problem in RL where the policy achieves a high reward but fails to meet the actual objective (e.g. exploiting potential shortcuts like response length and style to develop specific response patterns to hack the reward model). (2) In RLHF, it requires maintaining a reward model that mimics human judgment—a process prone to errors like reward hacking (3) Not only does it not fully satisfy human needs, it also increases arithmetic requirements.

1.3 Direct Preference Optimization (DPO) Loss Function:

Let x be the input, y_u the preferred response, and y_l the less preferred response.

1. $\pi_\theta(y_u \mid x),\ \pi_\theta(y_l \mid x):$ Probabilities assigned by the model with parameters θ.

2. $\pi_{\text{ref}}(y_u \mid x),\ \pi_{\text{ref}}(y_l \mid x):$ Probabilities from the reference (baseline) policy.

3. $\beta \log \frac{\pi_\theta(y_u \mid x)}{\pi_{\text{ref}}(y_u \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}:$
Optimizes the model to prefer y_u over y_l compared to the baseline; β controls baseline adherence.

4. $\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = \mathbb{E}_{x,y_u,y_l}\left[\beta \log \frac{\pi_\theta(y_u \mid x)}{\pi_{\text{ref}}(y_u \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right]:$
Expected value across all input and response pairs for overall optimization.

1.4 Can DPO scale to a real preference dataset?

1.4.1 What is the 'real preference dataset'? A real preference dataset refers to data collected from genuine human judgments or choices, as opposed to synthetic or automatically generated preferences. Automatic metrics like ROUGE often misalign with human judgments in summarization. Models fine-tuned via human preferences (e.g., PPO) yield more relevant summaries[14]. To compare, DPO was evaluated with PPO and Preferred-FT[16] on the TL;DR dataset.

1.4.2 How is temperature generally implemented efficiently in large language models (LLMs)? The temperature parameter is not predetermined. During inference, the model outputs a logits vector; temperature is applied by dividing each logit by the temperature value before softmax normalization, thereby modifying the resulting probability distribution.

1.4.3 How does it perform? DPO achieved a 61% win rate at temperature 0.0, slightly outperforming PPO's 57% at its best setting. DPO also showed more stable performance across temperatures, unlike PPO's variability.

1.5 Other about DPO itself

1.5.1 What is Single-turn dialogue? A single turn refers to one exchange between a user and a model, typically consisting of one prompt and one response, without ongoing conversation history.

1.5.2 Single-turn dialogue performance. DPO outperforms or matches other methods in single-turn dialogue tasks from the Anthropic HH dataset, including strong baselines like Preferred-FT and Best-of-128. It uniquely enhances preferred responses and converges rapidly, highlighting its efficiency.

2 Mathematical Understanding of DPO-Driven Advantages

2.1 What is human's preference?

The BT formula proposes that the human preference distribution p^* for a given response pair can be expressed as the following formula:

$$p^*(y_1 \succ y_2 \mid x) = \frac{\exp\left(r^*(x, y_1)\right)}{\exp\left(r^*(x, y_1)\right) + \exp\left(r^*(x, y_2)\right)} \tag{4}$$

2.2 Difference in Reward Function

In RHLF, the reward function is trained separately via loss function (2). In contrast, DPO simplifies the process as follows:

$$Z(x) = \sum_y \pi_{\text{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right) \tag{5}$$

The partition function $Z(x)$ normalizes the policy distribution $\pi_r(y \mid x)$:

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

Applying identity transformation and substituting r into the original reward function r^*:

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

Substituting into Eq.4:

$$p^*(y_1 \succ y_2 \mid x) = \frac{1}{1 + \exp\left(\beta \log \frac{\pi_r^*(y_2 \mid x)}{\pi_{\text{ref}}(y_2 \mid x)} - \beta \log \frac{\pi_r^*(y_1 \mid x)}{\pi_{\text{ref}}(y_1 \mid x)}\right)}$$

Approximated by:

$$p^*(y_1 \succ y_2 \mid x) = \sigma\left(r^*(x, y_1) - r^*(x, y_2)\right) \tag{6}$$

Yielding the loss function:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_u, y_l) \sim \mathcal{D}}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_u \mid x)}{\pi_{\text{ref}}(y_u \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right] \tag{7}$$

3 Overview of Research Trends

Broadly speaking, there is a central goal shared by the community: to help large, unsupervised language models better align with human preferences. A variety of research themes have emerged around this objective, including Reward Models (explicit/implicit, pointwise/preference, response-level/token-level, negative/positive), Feedback Mechanisms (binary/preference, human/AI, pair-wise/listwise), and Reinforcement Learning approaches (reference-free/reverse, length control/reverse, different divergences, on-policy/off-policy). These themes have given rise to a range of methodologies. Recently, one approach has stood out as particularly promising—Direct Preference Optimization (DPO). Many variants and related explorations have emerged from DPO itself.

The survey [19] provides a well-designed overall perspective on DPO. I extract its key research perspectives and discuss specific studies in a reduced form.

4 Research Questions and Variants

4.1 Effect of Implicit Reward Modeling

Examining the generalization capabilities of the implicit reward modeling employed in DPO.

4.1.1 A General Theoretical Paradigm to Understand Learning from Human Preferences[6].

- *Problem:* DPO tends to overfit the reward model.
- *Method:* Introduced a novel loss function to mitigate reward overfitting.
- *Limitation:* Focused on overfitting, not generalization under distribution shifts.

4.1.2 On the limited generalization capability of the implicit reward model induced by direct preference optimization[9].

- *Problem:* Implicit reward modeling in DPO (DPORM) underperforms in generalization compared to explicit reward modeling (EXRM), especially under distribution shifts.
- *Method:* Empirical evaluation across five out-of-distribution (OOD) settings comparing DPORM and EXRM.
- *Discovery:* DPORM suffers a 3% average and up to 7% accuracy drop under OOD conditions.

4.1.3 Policy optimization in rlhf: The impact of out-of-preference data[8].

- *Problem:* Policy optimization methods (PPO, DPO, IPO) incur errors when aligning with user preferences from limited distributions.
- *Contribution:* Theoretical and empirical analysis emphasizing the importance of OOD preference data; showed that EXRM improves policy performance.
- *Limitation:* Did not address how to enhance DPORM's generalization.

4.1.4 Generalizing reward modeling for out-of-distribution preference learning[7].

- *Problem:* Reward models often fail to generalize across distributions.
- *Contribution:* Meta-learned a general reward model via bilevel optimization to guide policy learning under varying distributions.
- *Limitation:* Focused on EXRM; DPO generalization remains unaddressed.

4.1.5 Regularizing hidden states enables learning generalizable reward model for llms[20].

- *Problem:* Reward models lack robustness to distribution shifts.
- *Method:* Regularized hidden states by preserving language model capabilities with auxiliary text-generation losses, while learning a reward.
- *Limitation:* Focused on EXRM; DPO generalization remains unaddressed.

4.2 Effect of Different Feedback

Obtaining high-quality pair-wise preference data is both costly and time-consuming, posing challenges for scalability. Additionally, instance-level optimization may not fully leverage the potential of preference data. It should be emphasized that while reward is a critical component of feedback, DPO implicitly models reward. Consequently, feedback is more directly derived from (or understood through) the final objective function.

4.2.1 What are point-wise,pair-wise and other rewards? As shown in Tab.1, Point-wise rewards assign a numerical score to each model output individually. Pair-wise preference data indicate which response is preferred from a pair of model-generated outputs for the same input, capturing relative preference rather than absolute scores.

Table 1. Comparison of Different Feedback Granularities for Model Optimization

Feedback Type	Evaluation Target	Feedback Format
Point-wise	Single, complete model output (e.g., a full response)	Absolute scalar score or categorical rating (e.g., 1-5 scale, "good"/"bad")
Pair-wise	Two distinct model outputs (for the same input prompt)	Relative preference between the pair (e.g., Output A is better than Output B)
List-wise	A set or list of multiple model outputs (for the same input)	Ordinal ranking or partial ordering of the outputs in the list
Binary	Single, complete model output	Dichotomous judgment (e.g., acceptable/unacceptable, correct/incorrect)
Step-wise	Intermediate steps or states within a sequential generation process	Evaluation of individual steps or decision points (e.g., correctness of a reasoning step)
Token-wise	Specific tokens or spans of text within a model output	Annotation or fine-grained evaluation at the token/span level (e.g., identifying erroneous tokens)

4.2.2 *Raft: Reward ranked finetuning for generative foundation model alignment[4].*

- *Feedback:* A reward ranked fine-tuning method to explore the list-wise feedback.
- *Core idea:* The model iteratively learns from the induced best-of-K policy [3, 11], which samples K responses and selects the one with the highest reward as the final output. Then the model is fine-tuned on the optimal responses.

4.2.3 *Rrhf: Rank responses to align language models with human feedback[21].*

- *Feedback:* Exploiting rank from human annotators or reward models by combining a modified rank loss with SFT loss.
- *Trick:* To avoid explicit reward model, they take length-normalized conditional log probability of responses under policy model π_θ as reward score.
- *Core idea:* Letting the policy model π_θ give larger probabilities for better responses and give smaller probabilities for worse responses

4.2.4 *Advancing llm reasoning generalists with preference trees[22].*

- *Feedback:* A data collection method named ULTRAINTERACT for tree-structured preference data, especially in the reasoning domain.
- *Contribution:* (1) Decomposing complex tasks into multiple steps to obtain multi-turn model actions. (2) Modeling correct and incorrect actions organized in binary tree structures. (3) A critique model to refine the solution while the actor interact with the Python environment. (4) By training an explicit reward model, they enhanced the Bradley-Terry objective (Eq.4) with a term to directly boost the rewards of chosen actions while decreasing the rewards of rejected ones.

4.2.5 *Model alignment as prospect theoretic optimization[5].*

- *Feedback:* Maximizing the utility(*usability* or *satisfaction*) of LLM generations directly rather than maximizing the log-likelihood of preferences inspired from prospect theory[17].
- *Core idea:* It focuses on discerning whether a preference is desirable or undesirable which eliminates the need for two preferences for the same input.

4.2.6 *From r to q^*: Your language model is secretly a q-function[13].*

- *RLHF (PPO stage):* Frames text generation as a token-level MDP, a multi-step sequential decision process where tokens are selected at each step, and reward is based on the entire sequence's quality.
- *DPO (original perspective):* Frames text generation as a bandit problem, a single-step decision where an entire response constitutes a single action, and preference is derived from comparing complete outputs, disregarding token-level choices.
- *Contribution:* Despite its initial conceptualization as a simple bandit problem, DPO can be re-derived as a specific token-level MDP reinforcement learning algorithm (inverse Q-learning). This implies DPO satisfies the Bellman equation, theoretically linking it to more complex, sequential RLHF methods.

4.2.7 *Token-level direct preference optimization[24].*

- *Feedback:* Token-level Direct Preference Optimization (TDPO), an approach to align LLMs with human preferences by optimizing policy at the token level.
- *Method:* (1) Incorporating forward KL divergence constraints for each token. (2) Utilizing the Bradley-Terry model 4 for a token-based reward system.
- *Contribution:* Method(1) improving alignment and diversity. Method(2) enhancing the regulation of KL divergence, while preserving simplicity without the need for explicit reward modeling.

4.2.8 *A complex relation: Nash-Learning&Point-wise&pair-wise&DPO.*

(1) **Two Main Deficiencies of Deriving Pairwise Preferences from Pointwise Rewards via the BT Model:**
   - **Suboptimal Performance Compared to Direct Methods:**
     - *Deficiency:* The approach of first assigning pointwise rewards to individual responses and then using the Bradley-Terry (BT) model to infer pairwise preferences (e.g., response A is preferred over B) was found to be less effective or "not comparable" to methods that directly model preferences from explicit pairwise comparisons.
     - *Intuitive Understanding:* This indirect inference step can lead to a loss of information or a less accurate representation of true preferences compared to directly learning from data explicitly stating "A is better than B". It's an approximation that might not capture the nuances of direct human comparative judgment.
   - **Failure to Address Inconsistencies within Pairwise Preferences:**
     - *Deficiency:* This method inherently struggles with, or rather masks, inconsistencies often present in human preference data, such as cyclical preferences (e.g., $A > B$, $B > C$, but $C > A$).
     - *Intuitive Understanding:* Pointwise rewards typically impose a transitive, linear ordering (if score(A) > score(B) and score(B) > score(C), then score(A) > score(C)). When the BT model derives preferences from these scores, it inherits this enforced consistency. Consequently, the model doesn't learn to handle or represent the underlying non-transitive nature or other inconsistencies that might exist in the raw preference judgments, as these are effectively "ironed out" by the initial pointwise scoring.

(2) **How Nash Learning Methodologies Overcome These Limitations:**

- *Addressing Comparability:* Nash learning frameworks inherently operate on direct pairwise comparisons. Each LLM (player) aims to maximize its probability of being preferred over its opponent. This directly optimizes for relative superiority in a pairwise context, aligning with the strengths of direct pairwise preference modeling.
- *Addressing Inconsistencies:* In a game-theoretic setting, players (LLMs) learn equilibrium strategies. If the underlying preference landscape is inconsistent (e.g., non-transitive), the system doesn't force a single, globally consistent ranking. Instead, models learn to navigate this complex space, potentially leading to mixed strategies or cyclical dynamics that reflect the inconsistencies rather than ignoring them. The goal shifts from finding a universally "best" response to finding a strategy that performs optimally given the opponent's strategy and the potentially inconsistent preference structure.

(3) **How DPO Overcomes These Limitations and Its Distinction from Nash Learning:**
- **How DPO Overcomes Limitations:**
  - *Addressing Comparability:* DPO directly optimizes a policy using (chosen, rejected) preference pairs, bypassing the need for an explicit pointwise reward model. Its objective function is formulated to directly increase the likelihood of preferred responses and decrease the likelihood of dispreferred ones relative to each other.
  - *Addressing Inconsistencies:* DPO learns from the provided dataset of preference pairs. If this dataset contains inconsistencies (e.g., the same input yields $A > B$ sometimes and $B > A$ other times, or implies $A > B, B > C, C > A$ across different examples), DPO's optimization process attempts to find a policy that best fits this (potentially noisy or inconsistent) data distribution. It doesn't explicitly model the inconsistency but learns a policy robust to it or one that reflects the aggregate signal from the data. It does not enforce transitivity a priori like the pointwise reward approach.
- **Distinction from Nash Learning:**
  - *Learning Paradigm:* DPO is typically a single-agent optimization problem where a policy is trained against a static dataset of preferences. Nash learning involves multiple (at least two) agents learning dynamically and competitively, where each agent's optimal strategy depends on the strategies of other agents.
  - *Objective:* DPO aims to maximize the log-likelihood of the observed human preferences in the dataset. Nash learning aims to find a Nash equilibrium, where no player can unilaterally improve its outcome (probability of being preferred over its opponent) by changing its strategy.
  - *Reward Signal Dynamism:* In DPO, the "reward" is implicitly defined by the fixed preference pairs. In Nash learning, the "reward" (being preferred) is dynamic and depends on the opponent's current response and the preference evaluator (which could be a reward model or human feedback within the competitive loop).
  - *Modeling of Inconsistencies:* While DPO learns a policy that is implicitly robust to inconsistencies in the data, Nash learning can, in principle, model and even exhibit behaviors (like cyclical strategies) that explicitly reflect non-transitive preference structures if they lead to an equilibrium.
  - *Relevant works:* Nash learning from human feedback[10], Self-play preference optimization for language model alignment[18], Direct nash optimization: Teaching language models to self-improve with general preferences[15].

4.2.9 Self-Play fIne-tuNing [2, 18].

- *Method:* Self-Play fIne-tuNing (SPIN) considered a two-player game, where the main player distinguishes the generated responses are from model or human, while the opponent player generates responses indistinguishable from human.
- *Contribution:* Eliminating the need for a reward model and derived a objective in a similar form to DPO.

4.2.10 Negative Log-likelihood.

- *What is it? :* A negative log-likelihood (NLL) loss [23] is similar to SFT (supervised fine-tuning) to rank loss.

$$\begin{aligned}\mathcal{L}_{\text{DPO+NLL}} =& \mathcal{L}_{\text{DPO}}(c_i^w, y_i^w, c_i^l, y_i^l|x_i) + \alpha\mathcal{L}_{\text{NLL}}(c_i^w, y_i^w|x_i) \\ =& -\log\sigma\left(\beta\log\frac{M_\theta(c_i^w, y_i^w|x_i)}{M_t(c_i^w, y_i^w|x_i)} - \beta\log\frac{M_\theta(c_i^l, y_i^l|x_i)}{M_t(c_i^l, y_i^l|x_i)}\right) \\ & -\alpha\frac{\log M_\theta(c_i^w, y_i^w|x_i)}{|c_i^w| + |y_i^w|}\end{aligned} \tag{8}$$

$\mathcal{L}_{\text{DPO+NLL}}$ Total loss function, combining DPO and NLL losses.
$\mathcal{L}_{\text{DPO}}$ Direct Preference Optimization (DPO) loss term.
$\mathcal{L}_{\text{NLL}}$ Negative Log-Likelihood (NLL) loss term.
c_i^w, y_i^w Preferred Chain-of-Thought (CoT) reasoning c and answer y for input x_i.
c_i^l, y_i^l Dispreferred CoT reasoning c and answer y for input x_i.
x_i Input question.
$M_\theta(\cdot|x_i)$ Probability of generating a sequence under the current model (with parameters θ) given input x_i.
$M_t(\cdot|x_i)$ Probability of generating a sequence under the reference model (from iteration t) given input x_i.
$\sigma(\cdot)$ Sigmoid function.
β Temperature hyperparameter for the DPO loss.
α Weighting hyperparameter for the NLL loss.
$|c_i^w| + |y_i^w|$ Total length of the preferred CoT reasoning c_i^w and answer y_i^w.

- *Problem:* DPO is intuitively expected to increase chosen and decrease rejected response likelihoods. However, this conflicts with the observed decrease in chosen response likelihood over time [12, 13].
- *Usage:* Forcing the model to learn the response with the highest reward.
- *Contribution:* When training with DPO without negative log-likelihood (NLL) loss, the log probabilities of chosen sequences barely increase over training; when training with DPO with NLL loss normalized by the total response length, the log probabilities increase noticeably. Thus, it believed that NLL enhances learning over the winning response from each pair.

4.3 Effect of KL Penalty Coefficient and Reference Model

Investigating the impact of the KL penalty coefficient, which constrains the policy model to remain within a specified proximity to the reference model, and the choice of the reference model.

4.3.1 What is instance-level optimization? Instance-level optimization refers to updating model parameters using feedback from individual training examples (e.g., a single response or a response pair), rather than leveraging feedback aggregated over multiple samples or lists.

4.4 Online DPO

Exploring iterative and online variants of DPO, as well as strategies for efficiently collecting new. preference datasets

4.5 Reward Hacking

Overcoming the limitations caused by reward hacking arising from both explicit and implicit reward models. Alignment Tax Investigating the alignment tax and proposed methods to reduce its effect.

4.5.1 What is alignment tax? A term describing the cost in performance, efficiency, or capability incurred when training LLMs to align better with human preferences, such as sacrificing some original abilities or performance.

5 Elements to be explored

5.1 Datasets

5.1.1 Human labeled .

5.1.2 Human labeled .

5.2 Applications

5.2.1 Application on Large Language Models.

5.2.2 Application on Multi-modal Understanding and Generation.

5.2.3 More Applications.

5.3 proximal policy optimization (PPO).

Proximal policy optimization (PPO) is a reinforcement learning (RL) algorithm for training an intelligent agent. Specifically, it is a policy gradient method, often used for deep RL when the policy network is very large.

5.4 Research Themes

Reward Models (explicit/implicit, pointwise/preference, response-level/token-level, negative/positive), Feedback Mechanisms (binary/preference, human/AI, pairwise/listwise), and Reinforcement Learning approaches (reference-free/reverse, length control/reverse, different divergences, on-policy/off-policy).

5.5 Implicit Reward Function

Implicit reward function that is parameterized by the policy model itself.[1]

References

[1] Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952.

[2] Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024.

[3] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[4] Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023.

[5] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Model alignment as prospect theoretic optimization. In *Forty-first International Conference on Machine Learning*, 2024.

[6] Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences. *arXiv e-prints*, pages arXiv–2310, 2023.

[7] Chen Jia. Generalizing reward modeling for out-of-distribution preference learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 107–124. Springer, 2024.

[8] Ziniu Li, Tian Xu, and Yang Yu. Policy optimization in rlhf: The impact of out-of-preference data. *arXiv preprint arXiv:2312.10584*, 2023.

[9] Yong Lin, Skyler Seto, Maartje Ter Hoeve, Katherine Metcalf, Barry-John Theobald, Xuan Wang, Yizhe Zhang, Chen Huang, and Tong Zhang. On the limited generalization capability of the implicit reward model induced by direct preference optimization. *arXiv preprint arXiv:2409.03650*, 2024.

[10] Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Andrea Michi, et al. Nash learning from human feedback. *arXiv preprint arXiv:2312.00886*, 18, 2023.

[11] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback, 2022. *URL https://arxiv. org/abs/2112.09332*, 2022.

[12] Arka Pal, Deep Karkhanis, Samuel Dooley, Manley Roberts, Siddartha Naidu, and Colin White. Smaug: Fixing failure modes of preference optimisation with dpo-positive. *arXiv preprint arXiv:2402.13228*, 2024.

[13] Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From r to Q^*: Your language model is secretly a q-function. *arXiv preprint arXiv:2404.12358*, 2024.

[14] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

[15] Corby Rosset, Ching-An Cheng, Arindam Mitra, Michael Santacroce, Ahmed Awadallah, and Tengyang Xie. Direct nash optimization: Teaching language models to self-improve with general preferences. *arXiv preprint arXiv:2404.03715*, 2024.

[16] Fahim Tajwar, Anikait Singh, Archit Sharma, Rafael Rafailov, Jeff Schneider, Tengyang Xie, Stefano Ermon, Chelsea Finn, and Aviral Kumar. Preference fine-tuning of llms should leverage suboptimal, on-policy data. *arXiv preprint arXiv:2404.14367*, 2024.

[17] Amos Tversky and Daniel Kahneman. Advances in prospect theory: Cumulative representation of uncertainty. *Journal of Risk and uncertainty*, 5:297–323, 1992.

[18] Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. Self-play preference optimization for language model alignment. *arXiv preprint arXiv:2405.00675*, 2024.

[19] Wenyi Xiao, Zechuan Wang, Leilei Gan, Shuai Zhao, Wanggui He, Luu Anh Tuan, Long Chen, Hao Jiang, Zhou Zhao, and Fei Wu. A comprehensive survey of direct preference optimization: Datasets, theories, variants, and applications. *arXiv preprint arXiv:2410.15595*, 2024.

[20] Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. Regularizing hidden states enables learning generalizable reward model for llms. *arXiv preprint arXiv:2406.10216*, 2024.

[21] Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback. *Advances in Neural Information Processing Systems*, 36:10935–10950, 2023.

[22] Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, et al. Advancing llm reasoning generalists with preference trees. *arXiv preprint arXiv:2404.02078*, 2024.

[23] Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. Iterative reasoning preference optimization. *arXiv e-prints*, pages arXiv–2404, 2024.

[24] Yongcheng Zeng, Guoqing Liu, Weiyu Ma, Ning Yang, Haifeng Zhang, and Jun Wang. Token-level direct preference optimization. *arXiv preprint arXiv:2404.11999*, 2024.