Common Pitfalls of Margin-based Preference Optimization in Language Model Alignment

Anonymous authors
Paper under double-blind review

Abstract

Reinforcement Learning from Human Feedback (RLHF) has become the predominant approach for aligning language models (LMs) to be more helpful and less harmful. At its core, RLHF uses a margin-based loss for preference optimization, which specifies the ideal LM behavior only in terms of the difference between preferred and dispreferred responses. This *under-specification* of ideal behavior for each response individually leads to two unintended consequences as the margin increases: (1) The probability of dispreferred (e.g., unsafe) responses may increase, resulting in potential safety alignment failures. (2) When the probability of dispreferred responses is reduced, this often coincides with a decrease in the probability of preferred responses, even when these responses are ideal. In this paper, we identify the fundamental issue: margin-based preference optimization loss under-specifies ideal LM behaviors. We derive key conditions under which the probabilities of both preferred and dispreferred responses increase or decrease together. These conditions occur when the inner products between the gradients of the log-probabilities of preferred and dispreferred responses are large. We theoretically analyze when such inner products are large and empirically validate our findings. Our framework also reveals important differences in the training dynamics of various preference optimization algorithms and suggests new directions for developing better algorithms for language model alignment.

1 Introduction

Reinforcement Learning from Human Feedback (RLHF) has become a primary approach for aligning Language Models (LMs) to improve their helpfulness and mitigate harmfulness (Stiennon et al., 2020; Bai et al., 2022; Ouyang et al., 2022). This pipeline typically consists of two stages: supervised fine-tuning (SFT), where demonstration data is used to directly teach the model desirable behaviors, and the reinforcement learning (RL) stage, which uses preference data—comparisons between different responses to the same prompt—to highlight the contrast between chosen and rejected responses, with the goal of helping the model learn distinctions between good and bad behaviors.

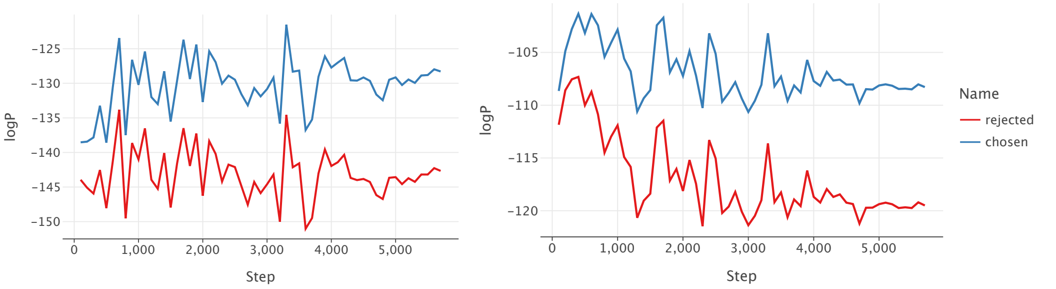
In its vanilla form, the RL stage first employs a contrastive loss—based on the margin between the scores of the chosen and rejected responses—to train a reward model, followed by policy optimization methods to fine-tune the LM. Leveraging the structure of the problem, a recent line of work has combined these two steps by directly optimizing the language model using a margin-based preference optimization loss of the following general form (Rafailov et al., 2024; Azar et al., 2024; Xu et al., 2024; Ethayarajh et al., 2024; Hong et al., 2024; Pal et al., 2024; Park et al., 2024; Yuan et al., 2024; Meng et al., 2024; Zhao et al., 2023; Wu et al., 2024):[1]

$$\ell(x, y_w, y_l; \theta) = m(h_w(\log \pi_\theta(y_w|x)) - h_l(\log \pi_\theta(y_l|x))), \tag{1}$$

for a language model $\pi_\theta(y|x)$ that specifies the probability of response y given prompts x, a dataset consisting of pairs of chosen responses y_w and rejected responses y_l for the same prompt x. These preference optimization losses can be interpreted as varying the scalar functions m, h_w, h_l (Section 3.2 and Table 2). At the core, they all rely on the *margin* between a transformation of the chosen log-probability $\log \pi_\theta(y_w|x)$ and a transformation of the rejected log-probability $\log \pi_\theta(y_l|x)$.

[1] The reward modeling loss in vanilla RLHF is also an example of this general form.

The training dynamics of these margin-based preference optimization is quite intriguing—the log probabilities of the chosen and rejected responses often show a synchronized increase and decrease (Figure 1). It is worth noting that, by the end of the training, even though the margin increases (resulting in the minimization of the margin-based preference optimization loss), the log probability of both the chosen and rejected responses may increase (Figure 1a), or both may decrease (Figure 1b).



(a) Mistral 7B

(b) Llama-3 8B

Figure 1: Training dynamics of the chosen and rejected log probabilities on the TL;DR dataset (Stiennon et al., 2020). As the margin between the two increases, the chosen and rejected log probabilities exhibit synchronized increases and decreases. In Figure 1a, both the chosen and rejected log probabilities increase in the end, whereas in Figure 1b, both decrease in the end.

This synchronized log-probability change exposes a fundamental issue with using margin-based loss for preference optimization in language model alignment: it only specifies the ideal behavior of the margin but not the ideal behavior of individual terms. This under-specification may have two problematic consequences:

- First, when the primary goal is to reduce the probability of generating rejected responses (e.g., in safety-related alignment tasks where certain undesirable responses should not be generated), merely increasing the margin (i.e., ensuring that the chosen response is preferred over the rejected one) does not guarantee that the log-probability of the rejected response is actually decreasing (Figure 1a).
- Second, even when the log-probability of the rejected response does decrease, the current margin-based losses often imply a simultaneous reduction in the log-probability of the chosen response (Figure 1b). This becomes particularly concerning under some of the current fine-tuning practices for LMs, where we want to retain or even increase the probability of generating the preferred responses. In the original procedure of RLHF, both chosen and rejected samples are drawn from models that require further training (Stiennon et al., 2020). In such cases, the ideal behavior of the model on the chosen samples is less clear—aside from being preferred over the rejected ones. However, in more recent work, the chosen and rejected samples are often synthetic data generated by strong language models and are used to distill these strong models into smaller ones (Dubey et al., 2024; Chiang et al., 2023; Tunstall et al., 2024; Taori et al., 2023). In some other cases, chosen samples may come from demonstration data collected during the SFT phase (Chen et al., 2024). In both scenarios, where the chosen responses are ideal, we want the probability of the chosen response to increase—or at least not decrease—to ensure the model retains a high probability of generating these ideal responses.

In this work, we dig into this phenomenon, identifying conditions under which the chosen and rejected log-probability $\log \pi_\theta(y_w|x), \log \pi_\theta(y_l|x)$ exhibits synchronized increase and decrease. Our first key finding is that this synchronized change happens when the gradient inner product $\langle \nabla_\theta \log \pi_\theta(y_w|x), \nabla_\theta \log \pi_\theta(y_l|x) \rangle$ is "large" relative to their individual norms (Section 3.1). The precise definition of "large" varies for different algorithms (Section 3.2). The gradient inner product conditions we derived enable us to characterize existing margin-based preference optimization methods, explain their differing training dynamics, and identify the appropriate conditions for using these algorithms. Our theoretical findings are also validated through empirical observations (Section 3.3).

We further investigate when these gradient inner product conditions may fail. In synthetic settings, we theoretically show that (1) as the chosen and rejected responses share more similar tokens, their gradient inner product will increase, and (2) while the sentence-level gradient inner product may be large and positive, individual token-level inner products can be small and negative (Section 4.1). We validate these theoretical insights empirically, and our findings suggest the potential for more fine-grained preference optimization methods that leverage token-level information (Section 4.2).

To summarize, our contributions are as follows:

- We identify a key issue with margin-based preference optimization for LM alignment: it underspecifies the ideal behavior of the LM on chosen and rejected responses individually (Section 1);
- We provide a gradient inner product condition that captures when the synchronized movement of chosen and rejected log probabilities occurs for various margin-based losses (Section 3);
- We explore when the gradient conditions may fail theoretically and experimentally (Section 4);
- Using our framework, we categorize existing RLHF variants (Section 3.2) and outline future directions for language model alignment (Section 5).

2 Background and Related Work

2.1 Problem Setup

We consider auto-regressive language models $\pi(y^t|x, y^{<t})$ that specify the distribution of the next token y^t at index t on a finite vocabulary set $\mathcal{V}$, given the prefix tokens including the prompt x and the partially generated responses $y^{<t}$. In the context of LM alignment, there is a reference policy π_{ref}, usually obtained by large-scale pre-training and supervised fine-tuning, and serves as the sampling policy and start point of further alignment algorithms.

2.2 Preference Optimization

There have been plenty of works on the design of preference optimization losses. These loss designs are motivated by various assumptions or considerations. Here we briefly categorize them:

Rafailov et al. (2024) derive the DPO loss from the KL-constrained reward maximization problem:

$$\max_{\theta} \mathbb{E}_{x\sim\mathcal{X}, y\sim\pi_\theta(\cdot|x)}[r(y;x)] - \beta\mathbb{E}_{x\sim\mathcal{X}}[\mathrm{KL}(\pi_\theta(\cdot|x)\|\pi_{\text{ref}}(\cdot|x))]. \tag{2}$$

They further derive the DPO loss for any triplet (x, y_w, y_l) where the y_w, y_l are the chosen and rejected response, respectively:

$$\ell_{\text{DPO}}(x, y_w, y_l; \theta; \pi_{\text{ref}}) := -\log\sigma\left(\beta\left[\log\left(\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)}\right) - \log\left(\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right]\right).$$

Motivated by non-transitive human preference and language model calibration respectively, Azar et al. (2024) and Zhao et al. (2023) proposed the IPO and SlicHF loss with similar form that solely depend on the **margin** $\log\pi_\theta(y_w|x) - \log\pi_\theta(y_l|x)$.

Due to the length bias observed in practice, Park et al. (2024) propose to add a length penalty term in the BT preference model, but the gradient still relies on the margin $\log\pi_\theta(y_w|x) - \log\pi_\theta(y_l|x)$. Meng et al. (2024) and Yuan et al. (2024) consider the setting of average rewards and derive a loss dependent on the **length-normalized margin** $\frac{1}{|y_w|}\log\pi_\theta(y_w|x) - \frac{1}{|y_l|}\log\pi_\theta(y_l|x)$.

Unlike prior work, Ethayarajh et al. (2024) and Wu et al. (2024) do not consider the difference between the likelihood, but deal with the chosen and rejected response separately. These works typically assign a positive reward signal to the chosen response and a negative reward signal to the loser, according to the logistic loss (Ethayarajh et al., 2024) or the square loss (Wu et al., 2024).

Given that the decreasing log-probability of the chosen response is a well-observed phenomenon (Pal et al., 2024), it is natural to add explicit regularization to the loss objective to *force* the increase of the chosen response's log-probability. In particular, (Pal et al., 2024) proposed the DPOP loss that behaves the same as DPO when the chosen response's log-ratio $\log\left(\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)}\right)$ is above 0, and add explicit regularization when it is below 0 to push it up. Similarly, Xu et al. (2024) and Zhao et al. (2023) also add explicit regularization to maximize the chosen response's log-probability.

Among these works, the most relevant to ours is Pal et al. (2024), which touches upon the similar failure mode of DPO. The main difference is that they focus on how to mitigate the decrease of the chosen response's probability and propose new loss designs. In contrast, we dig deeper to obtain a broader view. We rigorously analyze the training dynamics and extract general success/failure conditions for different losses: the gradient similarity.

3 When will margin-based preference optimization be problematic?

The fundamental issue with margin-based preference optimization objectives is that they only specify the behavior of the difference between the terms depending on the log-probabilities of the chosen and rejected samples, but does not specify the behavior of these two terms individually (Section 1). In many cases where the chosen responses are ideal, we want to ensure that the chosen log-probability does not decrease as the rejected log-probability decreases. In this section, we identify the condition under which this occurs. We start with the condition for DPO (Section 3.1) and then move to the general case (Section 3.2).

3.1 Gradient Inner Product Conditions for DPO

At a high level, the change in the log-probability of the chosen and rejected responses is influenced by the inner product of their gradients. When this inner product becomes large relative to the squared ℓ_2 norm of the chosen log-probability gradient, the log probability of the chosen will decrease. The key factor here is the relative magnitude of the inner product compared to the gradient norms.

To see this, let us focus on the gradient of the DPO loss:

$$\nabla_\theta \ell_{\mathrm{DPO}}(\theta) = -\beta\sigma\left(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w)\right)\left[\nabla_\theta \log \pi_\theta(y_w \mid x) - \nabla_\theta \log \pi_\theta(y_l \mid x)\right], \quad (3)$$

where the implicit reward $\hat{r}_\theta(x, y) = \beta \log \frac{\pi_\theta(y|x)}{\pi_{\mathrm{ref}}(y|x)}$ is a scalar. We simplify the gradient by introducing notations:

$$\log \pi_w(\theta) := \log \pi_\theta(y_w|x),\ \log \pi_l(\theta) := \log \pi_\theta(y_l|x),\ c(\theta) := \sigma\left(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w)\right) > 0.$$

Considering a single sample (x, y_w, y_l), the DPO gradient (3) can be rewritten as[2]

$$\nabla_\theta \ell_{\mathrm{DPO}} = -\beta c(\theta) \cdot (\nabla_\theta \log \pi_w - \nabla_\theta \log \pi_l). \quad (4)$$

After one step of gradient descend with step size $\eta > 0$ for decreasing the loss ℓ_{DPO}, the changes $\Delta \log \pi_w$ in the log-probability of the chosen response $\log \pi_w$ and the changes $\Delta \log \pi_l$ in the log-probability of the rejected response $\log \pi_l$ can be approximated by the first-order Talyor expansion:

$$\begin{aligned}\Delta \log \pi_w &\approx \langle \nabla_\theta \log \pi_w, \eta \nabla_\theta \ell_{\mathrm{DPO}} \rangle = \eta\beta c(\theta) \cdot \left(\|\nabla \log \pi_w\|^2 - \langle \nabla \log \pi_w, \nabla \log \pi_l \rangle\right), \\ \Delta \log \pi_l &\approx \langle \nabla_\theta \log \pi_l, \eta \nabla_\theta \ell_{\mathrm{DPO}} \rangle = \eta\beta c(\theta) \cdot \left(\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle - \|\nabla \log \pi_l\|^2\right). \end{aligned} \quad (5)$$

If we measure the change in the margin $\log \pi_w - \log \pi_l$, i.e., $\Delta(\log \pi_w - \log \pi_l)$, then the Cauchy–Schwarz inequality ensures:

$$\Delta(\log \pi_w - \log \pi_l) \approx \eta\beta c(\theta) \cdot (\|\nabla \log \pi_w\|^2 - 2\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle + \|\nabla \log \pi_l\|^2) \geq 0, \quad (6)$$

which fulfills the contrastive goal of the DPO loss: enlarging the difference between the chosen log-probability $\log \pi_w$ and rejected log-probability $\log \pi_l$.

However, this does not ensure anything about the increment or decrement of chosen and rejected log-probability $\log \pi_w, \log \pi_l$ individually. There are three possible cases for the margin to increase:

- **Case 1 (Ideal):** $\log \pi_w$ **will increase and** $\log \pi_l$ **will decrease**;
- **Case 2:** $\log \pi_w$ **and** $\log \pi_l$ **are decreasing at the same time but** $\log \pi_l$ **decreases more**;
- **Case 3:** $\log \pi_w$ **and** $\log \pi_l$ **are increasing at the same time but** $\log \pi_l$ **increases more**.

As our derivation in (5) suggests, for DPO, we have the following conditions:

$$\begin{aligned}\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle \leq \|\nabla \log \pi_w\|^2 &\iff \Delta \log \pi_w \geq 0, \log \pi_w \uparrow \\ \langle \nabla \log \pi_w, \nabla \log \pi_l \rangle \leq \|\nabla \log \pi_l\|^2 &\iff \Delta \log \pi_l \leq 0, \log \pi_l \downarrow \end{aligned} \quad (7)$$

Case	$\Delta \log \pi_w, \Delta \log \pi_l$	$\log \pi_w, \log \pi_l$	Condition
1	$\Delta \log \pi_w \geq 0 \geq \Delta \log \pi_l$	$\log \pi_w \uparrow \log \pi_l \downarrow$	$\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle \leq \min(\lVert \nabla \log \pi_w \rVert^2, \lVert \nabla \log \pi_l \rVert^2)$
2	$0 \geq \Delta \log \pi_w \geq \Delta \log \pi_l$	$\log \pi_w \downarrow \log \pi_l \downarrow$	$\lVert \nabla \log \pi_w \rVert^2 \leq \langle \nabla \log \pi_w, \nabla \log \pi_l \rangle \leq \lVert \nabla \log \pi_l \rVert^2$
3	$\Delta \log \pi_w \geq \Delta \log \pi_l \geq 0$	$\log \pi_w \uparrow \log \pi_l \uparrow$	$\lVert \nabla \log \pi_l \rVert^2 \leq \langle \nabla \log \pi_w, \nabla \log \pi_l \rangle \leq \lVert \nabla \log \pi_w \rVert^2$

Table 1: Three possible cases of the changes on chosen and rejected log-probabilities in DPO.

	$m(a)$	$h_w(a)$	$h_l(a)$	$\Lambda(a)$
DPO (Rafailov et al.)	$\log \sigma(a - c_{\text{ref}})$	βa	βa	—
R-DPO (Park et al.)	$\log \sigma(a - (c_{\text{ref}} + \alpha(\lvert y_w \rvert - \lvert y_l \rvert)))$	βa	βa	—
SimPO (Meng et al.)	$\log \sigma(a - \gamma)$	$\frac{\beta}{\lvert y_w \rvert} a$	$\frac{\beta}{\lvert y_l \rvert} a$	—
IPO (Azar et al.)	$(a - (c_{\text{ref}} + \frac{1}{2\beta}))^2$	a	a	—
RRHF (Yuan et al.)	$\min(0, a)$	$\frac{1}{\lvert y_w \rvert} a$	$\frac{1}{\lvert y_l \rvert} a$	λa
SlicHF (Zhao et al.)	$\min(0, a - \delta)$	a	a	λa
CPO (Xu et al.)	$\log \sigma(a)$	βa	βa	λa
DPOP (Pal et al.)	$\log \sigma(a - c_{\text{ref}})$	$\beta a - \lambda \max(0, \log c^w_{\text{ref}} - a)$	βa	—
KTO (Ethayarajh et al.)	a	$\lambda_w \sigma(\beta a - (\log c^w_{\text{ref}} + z_{\text{ref}}))$	$\lambda_l \sigma((\log c^l_{\text{ref}} + z_{\text{ref}}) - a)$	—
SPPO (Wu et al.)	a	$(a - \beta^{-1})^2$	$(a + \beta^{-1})^2$	—

Table 2: Instantiation of margin-based preference optimization objectives. The constants in these objectives satisfy $\beta, \gamma, \delta, \lambda_w, \lambda_l > 0$.

To summarize, the conditions corresponding to the three cases are listed in Table 1.

We will see that in the general case with other margin-based objectives, a similar condition can be derived on the inner product $\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle$, but the condition, in some settings, is more lenient than the (7) above, explaining why some methods may mitigate the problem with DPO (Section 3.2).

3.2 Gradient Inner Product Conditions for General Margin-based Preference Optimization

Moving on to the more general case, as discuss in (1), except that sometimes, we have regularizers

$$\ell(\theta) = -\left(m(h_w(\log \pi_w) - h_l(\log \pi_l)) + \Lambda(\log \pi_w)\right), \tag{8}$$

where $\Lambda(\log \pi_\theta(y_w|x))$ is a scalar regularizer depending on the chosen log-probability. We instantiate this general form for popular preference optimization in Table 2, where we denote $c^w_{\text{ref}} := \log \pi_{\text{ref}}(y_w|x), c^l_{\text{ref}} := \log \pi_{\text{ref}}(y_l|x), c_{\text{ref}} := c^w_{\text{ref}} - c^l_{\text{ref}}$. Note that ℓ is a function of θ, thus $\pi_{\text{ref}}(y|x)$ shall be viewed as constant.

Using this general form, we analyze the gradient similar to the DPO case and identify criteria for increasing chosen log-probability and decreasing rejected ones. First, the gradient can be written as

$$\nabla_\theta \ell = d_w \nabla_\theta \log \pi_w - d_l \nabla_\theta \log \pi_l,$$

where the constants that do not depend on θ are

$$\begin{aligned} d_w &:= m'(h_w(\log \pi_w) - h_l(\log \pi_l)) h'_w(\log \pi_w) + \Lambda'(\log \pi_w) \\ d_l &:= m'(h_w(\log \pi_w) - h_l(\log \pi_l)) h'_l(\log \pi_l). \end{aligned}$$

After one step of gradient descend with step size $\eta > 0$ for decreasing the loss ℓ, the changes in the log-probability can be approximated by the first-order Taylor expansion:

$$\begin{aligned} \Delta \log \pi_w &\approx \langle \nabla_\theta \log \pi_w, \eta \nabla_\theta \ell \rangle = \eta \left(d_w \|\nabla_\theta \log \pi_w\|^2 - d_l \langle \nabla_\theta \log \pi_w, \nabla_\theta \log \pi_l \rangle \right) \\ \Delta \log \pi_l &\approx \langle \nabla_\theta \log \pi_l, \eta \nabla_\theta \ell \rangle = \eta \left(d_w \langle \nabla_\theta \log \pi_w, \nabla_\theta \log \pi_l \rangle - d_l \|\nabla_\theta \log \pi_l\|^2 \right). \end{aligned}$$

In our ideal setting, we want the margin to increase while increasing chosen log-probability and decreasing rejected log-probability:

$$\Delta \log \pi_w \geq 0 \geq \Delta \log \pi_l \implies \Delta(\log \pi_w - \log \pi_l) \geq 0.$$

[2] When the context is clear, we omit θ and just use $\log \pi_w$, $\log \pi_l$ and ∇.

This implies the following general condition:

$$\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle \leq \frac{d_w}{d_l} \|\nabla \log \pi_w\|^2 \iff \Delta \log \pi_w \geq 0, \log \pi_w \uparrow \tag{9}$$

$$\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle \leq \frac{d_l}{d_w} \|\nabla \log \pi_l\|^2 \iff \Delta \log \pi_l \leq 0, \log \pi_l \downarrow \tag{10}$$

Note that for one condition to be more lenient (e.g., $d_w/d_l > 1$ for winner), the other condition becomes more strict ($d_l/d_w < 1$ for loser). Accordingly, we can instantiate these conditions for different algorithms using their m, h_w, h_l, Λ. Here, we provide a brief discussion of some of the algorithms and explain why under certain settings, these algorithms may work differently from DPO using these conditions.

- DPO: $\frac{d_w}{d_l} = \frac{d_l}{d_w} = 1$.
- SPPO: $\frac{d_w}{d_l} = \frac{\beta^{-1} - \log \pi_w}{\beta^{-1} + \log \pi_l} > 1$[3], where β^{-1} is a large constant. Compared with DPO, SPPO loss ensures that it is easier for $\log \pi_w$ to go up (9) and harder for $\log \pi_l$ to go down (10). Since $\|\nabla_\theta \log \pi_l\|_2^2 > \|\nabla_\theta \log \pi_w\|_2^2$ in practice, The general condition for winner (10) is more likely to be satisfied.
- KTO: $\frac{d_w}{d_l} \propto \frac{\lambda_w}{\lambda_l}$, the ratio is determined by two hyperparameters in KTO, fine-tuned according to different tasks and datasets.
- Explicit Regularization to maximizing winner's log-likelihood (e.g., CPO): We always have $\frac{d_w}{d_l} > 1$ since $\Lambda' \geq 0$, i.e., it will increase d_w compared with DPO, and d_l is the same as in DPO. It is worth noting that this type of regularization always helps to make $\Delta \log \pi_w$ large. Again, if we have $\|\nabla_\theta \log \pi_l\|_2^2 > \|\nabla_\theta \log \pi_w\|_2^2$ in practice, CPO or the NLL term can help the winner's conditions (9) to hold while (10) remains true.
- SimPO: $\frac{d_w}{d_l} = \frac{|y_l|}{|y_w|}$. SimPO is partly motivated by the *length bias*: human (or LLM) labelers prefer longer but not necessarily more helpful responses. We have two cases:
  - $|y_w| > |y_l| \Rightarrow \frac{d_w}{d_l} < 1$. A longer winner will have its probability increase less, and the shorter loser will have its probability decrease more.
  - $|y_w| < |y_l| \Rightarrow \frac{d_w}{d_l} > 1$. A shorter winner will have its probability increase more, and the longer loser will have its probability decrease less.

  Compared to DPO, we can see that SimPO rewards a shorter winner more significantly and a longer winner less so. For a dataset that is not heavily length-biased, SimPO might behave similar to DPO in terms of the winner and loser's log-likelihood. The same reasoning also applies to Yuan et al. (2024) and Azar et al. (2024), which uses averaging in practice.

3.3 Empirical Observations

We conduct experiments on the TL;DR dataset (Stiennon et al., 2020) to showcase the phenomena. Figure 1 depicts how different margin-based preference optimization algorithms affect the log-likelihood of chosen and rejected responses.

For algorithms with explicit regularization on the winner's log-likelihood, such as **CPO**, **DPOP**, **RRHF**, and **Slic-HF**, we observe a consistent increase in the log-likelihood of the chosen (winner) responses. This behavior is expected based on the formulation of these methods, where explicit regularization ensures that the winner's log-likelihood is directly increased, aligning with the conditions discussed in Section 3.2.

For **DPO** and **R-DPO**, both the chosen and rejected log-likelihoods tend to decrease simultaneously. This behavior aligns with the analysis that shows how these methods, purely dependent on the margin, might result in both terms decreasing, with the rejected log-likelihood decreasing more significantly. This leads to an increase in the margin, which is the objective, but not necessarily an increase in the chosen log-probability.

SimPO and **IPO**[4] in Figure 1 report the *average* log-likelihood of responses. The simultaneous decrease in both the (average) chosen and rejected log-likelihoods is expected, because the loss only

[3] See Section A.1 for the derivation.

[4] In their original paper, Azar et al. (2024) proposed the IPO loss without average log-likelihood. The authors later claimed using average log-likelihood with IPO yields improved performance.

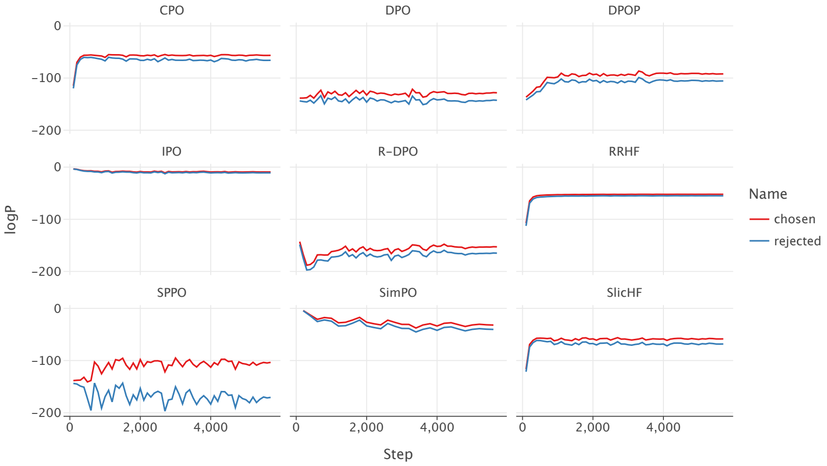


Figure 2: Training dynamics of the chosen and rejected log probabilities on the TL;DR dataset for different algorithms trained on Mistral 7B. The corresponding plot for Llama3 8B is in Figure 5 (Appendix C.5). All algorithms exhibit synchronized increases and decreases in the chosen and rejected log probabilities. We note that for SimPO and IPO, the log probabilities are normalized by the response length, while in the other plots, they are the original log probabilities. We also provide the cosine similarity between $\nabla_\theta \log \pi_w$ and $\nabla_\theta \log \pi_l$ for these cases (Figure 7, Appendix C.5).

depends on the length-normalized margin, $\frac{1}{|y_w|} \log \pi_\theta(y_w|x) - \frac{1}{|y_l|} \log \pi_\theta(y_l|x)$. Again, an increase in the margin is guaranteed, but not necessarily an increase in the average chosen log-probability.

SPPO demonstrates a distinct trend where the log-likelihood of the chosen responses increases slightly, while the log-likelihood of the rejected responses decreases. This matches the theoretical predictions from Section 3.2, where SPPO encourages a favorable increase in the chosen log-likelihood and a decrease in the rejected log-likelihood.

Overall, these experimental results closely align with the gradient-based conditions outlined in Section 3.2, demonstrating how explicit regularization, loss structures, and specific design choices influence the dynamics of preference optimization.

4 When will the Gradient Inner Product Conditions be Violated?

To ensure that the chosen log-probability increases while the rejected log-probability decreases, we need the inner product $\langle \nabla_\theta \log \pi_w, \nabla_\theta \log \pi_l \rangle$ to be relatively small compared to $\|\nabla_\theta \log \pi_w\|^2$ and $\|\nabla_\theta \log \pi_l\|^2$, respectively (Section 3). Focus on DPO, in this section, we study when this condition (7) may be violated and what causes the violation.

We use toy synthetic settings to analyze this problem and build up our general intuition on the gradient inner product. In these synthetic settings, we observe that (1) the gradient inner product increases as the chosen and rejected responses share more similar tokens; and (2) while the sentence-level gradient inner product can be large, individual token-level inner products may be small (Section 4.1). We then empirically verify our intuition (Section 4.2). All proofs are in Appendix B.

4.1 Theoretical Results

4.1.1 Positive result on when the condition holds

We first provide a positive result when condition (7) holds and DPO has the ideal behavior that pushes up the log-probability of chosen and pushes down the log-probability of rejected. We begin with set-ups for the LM and preference data.

LM with learnable last linear layer. We analyze DPO for optimizing an LM with a learnable last linear layer. We assume for prompt x and response y, at any index $i \in [L]$, the LM outputs:

$$\pi_\theta(y^i \mid x, y^{<i}) = s(h_i^\top \theta)[y^i],$$

where $L = |y|$, $\theta \in \mathbb{R}^{d\times V}$ is the learnable parameter, $h_i \in \mathbb{R}^d$ is the hidden state for the i-th token in response and $s : \mathbb{R}^V \to \Delta_{\mathcal{V}}$ denotes the softmax function.[5] The hidden states are assumed as frozen during DPO.

Data setup 1. Under the prompt x, the chosen and rejected responses both have only one token, that is, $y_w, y_l \in \mathcal{V}^1$ and $y_w[1] \neq y_l[1]$.[6]

The following theorem shows in this task, $\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle < 0$ so that gradient descent steps of DPO make sure $\log \pi_w$ increases and $\log \pi_l$ decreases.

Theorem 1. *Under the above data and model set-ups, assume after SFT stage, given prompt x the model prediction on the first token in response is uniformly concentrated on M tokens in the vocabulary $\mathcal{V}$, then we have*

$$\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle = -\frac{1}{M}\|h\|^2, \quad \|\nabla \log \pi_w\|^2 = \|\nabla \log \pi_l\|^2 = \frac{M-1}{M}\|h\|^2,$$
$$\log \pi_w \uparrow \textit{ and } \log \pi_l \downarrow,$$

with h being the hidden state of the token right after the prompt x.

Data setup 2. Under the prompt x, the chosen and rejected responses are of arbitrary same length L and only differ at the last token: i.e., $y_w[1 : L-1] = y_l[1 : L-1]$, $y_w[L] \neq y_l[L]$.

Results in Theorem 1 can be easily extended to the preference data in this setup, because up to the L-th token where chosen and rejected differ, the hidden states are the same for the two responses: Given $y_w[1 : L-1] = y_l[1 : L-1]$, we have that $h_i = h_{i,w} = h_{i,l}$ for $i \in [L]$.

Corollary 2. *In the case where y_w and y_l differ at the last token, assume after SFT the model prediction on L-th token in response is uniformly concentrated on M tokens in vocabulary, we have*

$$\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle \leq \|\nabla \log \pi_w\|^2 = \|\nabla \log \pi_l\|^2, \quad \log \pi_w \uparrow \textit{ and } \log \pi_l \downarrow,$$

with h_i being the hidden state of the i-th token in both responses.

4.1.2 Negative result on when the condition is violated

From the previous results, we can see the gradient inner product condition is not violated and DPO has the ideal behavior when the chosen and rejected responses differ only at the last token. However, as observed in Section 3.3, often time DPO is triggered to not behave in the ideal way, which suggests the underlying condition (7) is violated. To gain theoretical insights on what causes the violation in (7), we level up our previous data set-up.

Data setup 3. Chosen and rejected responses have an edit distance 1 and the difference appears in the middle of a response, i.e., the chosen and rejected responses $y_w \in \mathcal{V}^L$ and $y_l \in \mathcal{V}^L$ satisfy: $y_w[1 : m-1] = y_l[1 : m-1]$, $y_w[m] \neq y_l[m]$, $y_w[m+1 : L] = y_l[m+1 : L]$ for $1 \leq m < L$.

To analyze the optimization steps of DPO under this data setup, we first adopt a simpler setting for parameterizing the LM, where the LM has learnable logits.

LM with learnable logits. Let $V = |\mathcal{V}|$ be the vocabulary size. We first consider the setting where the LM output follows the structure: For index $i \in [L]$,

$$\pi_\theta(\cdot|x, y_w^{<i}) = s_{w,i}, \quad \pi_\theta(\cdot|x, y_l^{<i}) = s_{l,i},$$

where $s_{w,i}, s_{l,i} \in \Delta_{\mathcal{V}}$ are the probability distributions of the chosen and rejected response at token i, respectively. $s_{w,i}$ and $s_{l,i}$ are configured as variables to optimize in the model and to which we take derivative of chosen and rejected log probability.

Because $y_w[1 : m-1] = y_l[1 : m-1]$, we have that $s_i = s_{w,i} = s_{l,i}$ for $i \in [m]$. Since $s_{w,i}$ and $s_{l,i}$ are predicted by a shared model, they are not independent and one may impose assumptions to characterize the relationship between them. We denote for $i \in [m+1 : L]$, j_i^* to be the vocabulary index of token appearing at $y_w[i]$ and $y_l[i]$. As in Pal et al. (2024), we assume that $s_{w,i}[j_i^*] \geq s_{l,i}[j_i^*]$ and $s_{w,i}[j] \leq s_{l,i}[j]$ for $j \neq j_i^*$. Under this assumption, Theorem 3 shows that in this case the log-probability of the chosen and rejected will likely both decrease after one DPO step.

[5] Here, Δ denote the probability simplex.

[6] For a vector y, we use $y[i]$ to denote its i-th entry and use $y[i_1 : i_2]$ to denote its entry from i_1 to i_2.

Theorem 3. *When chosen and rejected responses have edit distance* 1 *and the differing token is the m-th token in the responses s.t $1 \leq m < L$, then after one DPO step, the per-token log-probability change in chosen response y_w can be characterized with first-order Taylor expansion. For $i \in [1 : m-1]$, the per-token log-probability before the difference stays unchanged:*

$$\Delta \log \pi(y_w^i \mid x, y_w^{<i}) \approx 0. \tag{11}$$

For $i = m$, the log-probability of chosen at the differing position will increase: suppose j^* and k^* are the indices of $y_w[m]$ and $y_l[m]$ in the vocabulary $\mathcal{V}$,

$$\Delta \log \pi(y_w^m \mid x, y_w^{<m}) \approx 1 + (s_{w,m}[j^*] - s_{w,m}[k^*]) \geq 0. \tag{12}$$

For $i \in [m+1 : L]$, the log-probability of chosen at these positions will decrease:

$$\Delta \log \pi(y_w^i \mid x, y_w^{<i}) \approx (1 - s_{w,i}[j_i^*])(s_{l,i}[j_i^*] - s_{w,i}[j_i^*]) - \sum_{j \neq j_i^*} s_{w,i}[j](s_{l,i}[j] - s_{w,i}[j]) \leq 0, \tag{13}$$

since $s_{l,i}[j_i^*] - s_{w,i}[j_i^*] \leq 0$ and $s_{l,i}[j] - s_{w,i}[j] \geq 0$. Given the change in sentence-wise log-probability of chosen is the summation of the per-token changes specified in (11), (12) *and* (13)*, as the same suffix following the differing tokens gets longer,* $\log \pi_w$ *decreases more.*

Remark 4. *It is worth mentioning that Theorem 3 explicitly presents the amount of probability changes, the same prediction on the change direction can also be derived with a per-token gradient inner product condition similar to* (7)*, see Appendix B.2. The decrease of* (12) *follows the same intuition obtained in Theorem 1 that if two contrast tokens are picked by chosen and rejected responses under a similar context, then the chosen token probability will increase. An intuitive explanation of what causes the decrease in* (13) *could be: the gradient of chosen and rejected are highly correlated as they pick the same token under a similar context. Mathematically, the assumption we adopted actually implies the gradient inner product between chosen and rejected is lower bounded.*

While Theorem 3 adopts the same assumptions made in Pal et al. (2024), we precisely characterizes the per-token log-probability changes based on the first-order approximation, and explicitly break down the sentence-wise probability change in chosen into 3 parts: before/at/after the differing position. Therefore, the analysis in Theorem 3 captures the varying probability change directions at different positions, uncovering the underlying dynamic behind the overall decreased probability observed in experiments (Figure 3).

Combining our insights gained in Section 4.1.1 and 4.1.2, we find that the gradient inner product increases as the chosen and rejected responses share more similar tokens. Additionally, the sentence-wise gradient inner product and their change in log probability may not necessarily reflect the individual token-wise gradient inner product and their probability changes.[7] Below we verify our theoretical findings empirically.

4.2 Empirical observations

We verify our intuition regarding when the gradient inner product condition may be held or violated using a sentiment classification task trained on GPT-2, where the prompt x is a statement, e.g., "Happy mothers day mumm xoxo." The chosen response y_w specifies the correct sentiment, while the rejected response y_l gives the wrong one. We consider three styles of responses:

- **Single token**: y_w: positive. y_l: negative.
- **Short suffix**: y_w: It has a positive sentiment. y_l: It has a negative sentiment.
- **Long suffix**: y_w: It has a positive sentiment based on my judgement. y_l: It has a negative sentiment based on my judgement.

Our theoretical results suggest that: (1) In the **single token** case, DPO would have a small gradient inner product, thus allowing the chosen log-probability to increase while the rejected to decrease (Theorem 1). (2) Between the **short suffix** and **long suffix** cases, we expect DPO to reduce the chosen log probability more for the latter, as it contains more tokens following the differing token between the chosen and rejected responses, leading to more chosen tokens with decreasing log

[7] To be specific, by token-wise gradient, we mean $\nabla_\theta \pi_\theta(y^i|x, y^{<i})$.

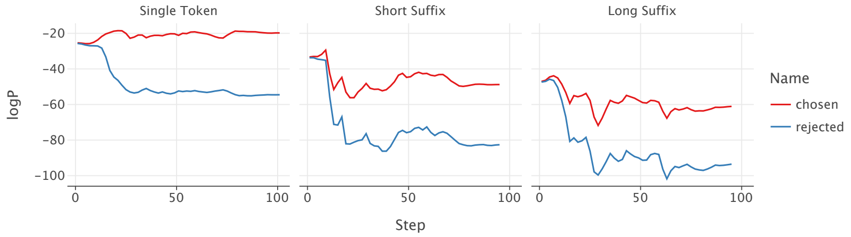
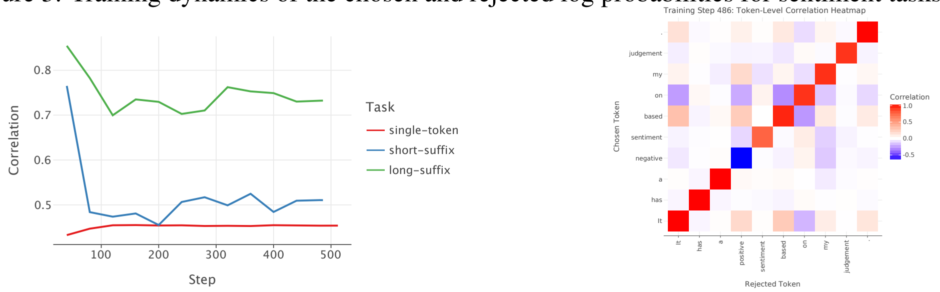


Figure 3: Training dynamics of the chosen and rejected log probabilities for sentiment tasks.



(a) Cosine similarity between $\nabla_\theta \log \pi_w$ and $\nabla_\theta \log \pi_l$ for DPO on three sentiment datasets.

(b) Token-wise gradient correlation for an instance in the **long suffix** task.

Figure 4: Gradient correlation behaviors on the sentence-level and token-level for sentiment tasks.

probability (Theorem 3). Additionally, our theoretical results suggest that for the differing token (e.g., "positive" vs. "negative"), the token-wise gradient inner product would be negative, while for identical tokens, the token-wise gradient inner product would be positive.

Empirically, we have the following observations, validating our theoretical intuition. First, the chosen log probability increases only in the **single token** case, and the **short suffix** chosen log probability decreases less than that of the **long suffix**, aligning with our theoretical results. (Figure 3). Second, the gradient cosine similarity increases as the suffix (i.e., the number of identical tokens in the response) grows, with the single token case being the lowest (Figure 4a). This aligns with our gradient condition (7), where the drop in chosen log probability depends on the magnitude of the gradient inner product. Finally, we inspect the token-wise gradient inner product heatmap for the **long suffix** case (Figure 4b). We observe from the diagonal of the heatmap that the inner product between the gradients on the tokens "positive" and "negative" is below 0, whereas for identical tokens in the two responses, the gradient cosine similarity is high. Our investigation into the token-level gradient inner product raises new questions about the role of token-level information in preference optimization and how we can leverage this fine-grained information to develop new algorithms.

5 Implications

In this paper, we touch upon a common pitfall of margin-based preference optimization methods in language alignment: it underspecifies the ideal behavior of the LM on the chosen and rejected responses individually. Our gradient inner product condition suggests that when the chosen and rejected gradients are similar, their log probabilities will exhibit synchronized increases and decreases. Using this gradient condition, we can categorize existing RLHF variants into two types: (1) those that modify the criterion for the size of the inner product, as seen in the works listed in Table 2, which rely on the same gradient inner product but apply different size criteria; and (2) those that change the inner product of interest directly. As discussed in Section 4, while the sentence-level gradient inner product may be large, the token-level inner product can be small. A line of research, such as advantage-based methods(Mudgal et al., 2023; Setlur et al., 2024), focuses on leveraging token-level information to improve RLHF and falls under the second category.

Finally, at a high level, our work highlights the need to reconsider the current margin-based preference optimization paradigm in language model alignment. While this approach may enable language models to effectively learn contrasts between good and bad responses, it may not be well-suited for settings where the focus is on the behavior of either the rejected or chosen samples—such as in safety-critical alignment tasks or when distilling from a strong model.

References

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pp. 4447–4455. PMLR, 2024.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

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.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. In *International Conference on Machine Learning*, 2024.

Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*, 2(4):5, 2024.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *arXiv preprint arXiv:2405.14734*, 2024.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, Yaguang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, Jilin Chen, Alex Beutel, and Ahmad Beirami. Controlled decoding from language models. *arXiv [cs.LG]*, October 2023.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

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.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization. *Findings of the Association for Computational Linguistics*, 2024.

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, 2024.

Amrith Setlur, Saurabh Garg, Xinyang Geng, Naman Garg, Virginia Smith, and Aviral Kumar. RL on incorrect synthetic data scales the efficiency of LLM math reasoning by eight-fold. *arXiv [cs.LG]*, June 2024.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Alpaca: A strong, replicable instruction-following model. *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, 3(6):7, 2023.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. Zephyr: Direct distillation of lm alignment. *Conference on Language Modeling*, 2024.

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.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. In *International Conference on Machine Learning*, 2024.

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, 2024.

Yao Zhao, Rishabh Joshi, Tianqi Liu, Misha Khalman, Mohammad Saleh, and Peter J Liu. Slic-hf: Sequence likelihood calibration with human feedback. *arXiv preprint arXiv:2305.10425*, 2023.

A Additional discussion on the gradient inner product condition in Section 3

A.1 Derivation for SPPO

Denote $\mathbf{a} = \nabla_\theta \log \pi(w)$ and $\mathbf{b} = \nabla_\theta \log \pi(l)$. For DPO, we see that the direction of winner and loser is decided by $\langle \mathbf{a}, \mathbf{a} - \mathbf{b} \rangle$ and $\langle \mathbf{b}, \mathbf{a} - \mathbf{b} \rangle$.

Similarly, for any pairwise loss $\ell(\log \pi(w) - \log \pi(l))$, the above statement still holds. Now we take a look at non-pairwise loss $\ell_{\text{SPPO}} = (\log \pi(w) - \beta^{-1})^2 + (\log \pi(l) + \beta^{-1})^2$. We have

$$\frac{d\theta}{dt} = -\nabla_\theta \ell_{\text{SPPO}} = -(\log \pi(w) - \beta^{-1}) \nabla_\theta \log \pi(w) - (\log \pi(l) + \beta^{-1}) \nabla_\theta \log \pi(l).$$

Then

$$\begin{aligned}\frac{d}{dt} \log \pi(i) &= \left\langle \nabla_\theta \log \pi(i), \frac{d\theta}{dt} \right\rangle \\ &= -(\log \pi(w) - \beta^{-1}) \big\langle \nabla_\theta \log \pi(i), \nabla_\theta \log \pi(w) \big\rangle - (\log \pi(l) + \beta^{-1}) \big\langle \nabla_\theta \log \pi(i), \nabla_\theta \log \pi(l) \big\rangle.\end{aligned}$$

We have

$$\frac{d}{dt} \log \pi(w) \approx -(\log \pi(w) - \beta^{-1}) \langle \mathbf{a}, \mathbf{a} \rangle - (\log \pi(l) + \beta^{-1}) \langle \mathbf{a}, \mathbf{b} \rangle$$

which means if we want $\log \pi(w)$ to increase, we need

$$\frac{\langle \mathbf{a}, \mathbf{b} \rangle}{\langle \mathbf{a}, \mathbf{a} \rangle} < \frac{\beta^{-1} - \log \pi(w)}{\beta^{-1} + \log \pi(l)} =: \alpha.$$

Note that the inequality above implicitly assume that $\beta^{-1} + \log \pi(l) > 0$. This is true in practice as we set β^{-1} to be extremely large. Similarly, if we want $\log \pi(l)$ to decrease, we need

$$\frac{\langle \mathbf{a}, \mathbf{b} \rangle}{\langle \mathbf{b}, \mathbf{b} \rangle} < \frac{\beta^{-1} + \log \pi(l)}{\beta^{-1} - \log \pi(w)} =: \alpha^{-1}.$$

We have $\alpha > 1$. It seems SPPO can make sure that $\log \pi(w)$ goes up more easily but also make $\log \pi(l)$ goes up more easily, compared to DPO.

B Derivation for the Gradient Inner products in Section 4

B.1 LM with learnable last linear layer: Single Token Case

We prove Theorem 1 below. WLOG, assume $T_w = T_l = L$,

$$\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle = \big\langle \nabla_\theta \log \pi(y_w^L \mid x, y_w^{<L}),\ \nabla_\theta \log \pi(y_l^L \mid x, y_l^{<L}) \big\rangle$$

$\theta \in \mathbb{R}^{d \times V}$, $h_L \in \mathbb{R}^d$ is the hidden state for the L-th token, $s(\cdot)$ is the softmax function.

$$\nabla_\theta \log \pi(y_w^L \mid x, y_w^{<L}) = \nabla_\theta \left(\log s(h_L^\top \theta)[y_w^L] \right) \tag{14}$$

$$\nabla_\theta \log \pi(y_l^L \mid x, y_l^{<L}) = \nabla_\theta \left(\log s(h_L^\top \theta)[y_l^L] \right) \tag{15}$$

Compute the gradient with chain rule,

$$\nabla_\theta \log \pi_w^L = [-s(1)h_L, \cdots, (1 - s(i_w))h_L, \cdots, -s(i_l)h_L, \cdots, -s(V)h_L] \tag{16}$$

$$\nabla_\theta \log \pi_l^L = [-s(1)h_L, \cdots, -s(i_w)h_L, \cdots, (1 - s(i_l))h_L, \cdots, -s(V)h_L], \tag{17}$$

i_w, i_l are the index of token y_w^L and y_l^L in vocabulary, respectively. For any index i, $s(i_w)$ denote LLM's output logit for the i-th token in vocabulary.

Suppose at the initialization of θ, $s(1) = \cdots = s(i_w) = \cdots = s(i_l) = s(v) = \frac{1}{M}$ for M entries and the rest $V - M$ entries have $s(j) = 0$. We note that the exact indices j of which $s(j) = 1/M$ does not matter as it would be the same index for both the chosen and rejected gradients.

$$\nabla \log \pi_w^L = [-\frac{1}{M} h_L, \ldots, \underbrace{\left(1 - \frac{1}{M}\right) h_L}_{i_w - th}, \cdots \underbrace{-\frac{1}{M} h_L}_{i_l - th}, \cdots, -\frac{1}{M} h_L] \tag{18}$$

$$\nabla \log \pi_l^L = [-\frac{1}{M} h_L, \cdots, \underbrace{-\frac{1}{M} h_L}_{i_w - th}, \cdots \underbrace{\left(1 - \frac{1}{M}\right) h_L}_{i_l - th}, \cdots - \frac{1}{M} h_L] \tag{19}$$

$$\left\langle \nabla \log \pi_w^L, \nabla \log \pi_l^L \right\rangle = \frac{M-2}{M^2} \|h_L\|^2 - 2 \cdot \frac{1}{M} \cdot \frac{M-1}{M} \|h_L\|^2 = -\frac{1}{M} \|h_L\|^2. \tag{20}$$

$\left\langle \nabla \log \pi_w^L, \nabla \log \pi_l^L \right\rangle$ is negative. While in comparison, the norm of $\nabla \log \pi_w^L$ and $\nabla \log \pi_l^L$ is large:

$$\|\nabla \log \pi_w^L\|^2 = \|\nabla \log \pi_l^L\|^2 = \frac{M-1}{M^2} \|h_L\|^2 + \left(1 - \frac{1}{M}\right)^2 \|h_L\|^2 = \frac{M-1}{M} \|h_L\|^2.$$

Therefore, based on our condition:

$$\begin{aligned}
&\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle = -\frac{1}{M} \|h_L\|^2, \\
&\|\nabla \log \pi_w\|^2 = \|\nabla \log \pi_l\|^2 = \frac{M-1}{M} \|h_L\|^2, \\
&\log \pi_w \uparrow \text{ and } \log \pi_l \downarrow .
\end{aligned}$$

B.2 LM WITH LEARNABLE LOGITS SETTING

We prove Theorem 3 below. We will set up some new notations first. First, we work with the case where $T_w = T_l = L$ is sentence length, V is the vocab size, $y_w[1 : m-1] = y_l[1 : m-1]$, $y_w[m] \neq y_l[m]$, and $y_w[m+1 : L] = y_l[m+1 : L]$. Note that for all $i \in [L]$, the token $y[i] \in [V]$ is an index.

Each row of the following matrix is $\pi_\theta(\cdot|x, y^{<i}) \in \Delta_{[V]}$ where i is the row index. (Here, there is a slight abuse of notation: Δ is the probability simplex.) $s : \mathbb{R}^V \to \Delta_V$ is the softmax function.

$$[0,1]^{L \times V} \ni \pi_\theta(x, y_w) = s(\overline{\theta}_w) = \begin{bmatrix} s(\overline{\theta}_w[1,:]) \\ \vdots \\ s(\overline{\theta}_w[m,:]) \\ s(\overline{\theta}_w[m+1,:]) \\ \vdots \\ s(\overline{\theta}_w[L,:]) \end{bmatrix}, \quad \pi_\theta(x, y_l) = s(\overline{\theta}_l) = \begin{bmatrix} s(\overline{\theta}_l[1,:]) \\ \vdots \\ s(\overline{\theta}_l[m,:]) \\ s(\overline{\theta}_l[m+1,:]) \\ \vdots \\ s(\overline{\theta}_l[L,:]) \end{bmatrix} = \begin{bmatrix} s(\overline{\theta}_w[1,:]) \\ \vdots \\ s(\overline{\theta}_w[m,:]) \\ s(\overline{\theta}_l[m+1,:]) \\ \vdots \\ s(\overline{\theta}_l[L,:]) \end{bmatrix}$$

Each row $s(\overline{\theta}[i,:]) \in \Delta_V$. The first m rows are the same for $\overline{\theta}_w$ and $\overline{\theta}_l$ because the tokens up to row m are the same between y_w and y_l. The index at row i corresponding to the selected token will be denoted as j_i^*, a generic vocab index is j. Note that, $j_i^* = j_{i,w}^* = j_{i,l}^*$ for $i \neq m$, and $j_{i,w}^* \neq j_{i,l}^*$ for $i = m$.

Next, the corresponding gradient matrices $\nabla \log s(\overline{\theta}_w), \nabla \log s(\overline{\theta}_l)$ can be specified by:

$$\mathbb{R}^{L \times V} \ni \nabla_\theta \log s(\overline{\theta}_w[i, j_{i+1}^*]) = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \nabla_{\overline{\theta}_w[i,:]} \log s(\overline{\theta}_w[i, j_i^*]) \\ \vdots \\ \mathbf{0} \end{bmatrix}, \quad \nabla_\theta \log s(\overline{\theta}_l) = \begin{bmatrix} \mathbf{0} \\ \vdots \\ \nabla_{\overline{\theta}_l[i,:]} \log s(\overline{\theta}_l[i, j_i^*]) \\ \vdots \\ \mathbf{0} \end{bmatrix}.$$

where

$$\nabla_{\overline{\theta}[i,:]} \log s(\overline{\theta}[i, j_i^*]) \in \mathbb{R}^V, \quad \text{and for } j \in [V], \nabla_{\overline{\theta}[i,:]} \log s(\overline{\theta}[i, j_i^*])[j] = \begin{cases} -s[i,j] & \text{if } j \neq j_i^* \\ 1 - s[i,j] & \text{if } j = j_i^* \end{cases}$$

where $s[i,j] = s(\overline{\theta}[i,:])[j]$, $\log s(\overline{\theta}[i, j_i^*])$ is j_i^*-th entry of $\log s(\overline{\theta}[i,:])$, and $\nabla \log s(\overline{\theta}[i, j_i^*])[j]$ is the j-th entry of the gradient of $\log s(\overline{\theta}[i, j_i^*])$.

The sentence-wise gradient is

$$\mathbb{R}^{L \times V} \ni \nabla_\theta \mathcal{L} \propto \begin{bmatrix} \nabla \log s(\overline{\theta}_w[1, j_1^*]) - \nabla \log s(\overline{\theta}_w[1, j_1^*]) \\ \vdots \\ \nabla \log s(\overline{\theta}_w[m, j_{m,w}^*]) - \nabla \log s(\overline{\theta}_w[m, j_{m,l}^*]) \\ \nabla \log s(\overline{\theta}_w[m+1, j_{m+1}^*]) - \nabla \log s(\overline{\theta}_l[m+1, j_{m+1}^*]) \\ \vdots \\ \nabla \log s(\overline{\theta}_w[L, j_L^*]) - \nabla \log s(\overline{\theta}_l[L, j_L^*]) \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{0} \\ \vdots \\ \nabla \log s(\overline{\theta}_w[m, j_{m,w}^*]) - \nabla \log s(\overline{\theta}_w[m, j_{m,l}^*]) \\ \nabla \log s(\overline{\theta}_w[m+1, j_{m+1}^*]) - \nabla \log s(\overline{\theta}_[m+1, j_{m+1}^*]) \\ \vdots \\ \nabla \log s(\overline{\theta}_w[L, j_L^*]) - \nabla \log s(\overline{\theta}_l[L, j_L^*]) \end{bmatrix}$$

Now, let's first derive the token-wise condition for the selected token (learning rate $\eta = 1$):
Chosen response: if $i = m$, we have

$$\begin{aligned} &\Delta \log s(\overline{\theta}_w[i, j_{i,w}^*]) \approx \sum_{i'=1}^{L} \langle \nabla \log s(\overline{\theta}_w[m, j_{m,w}^*]), \nabla \mathcal{L}[i', :] \rangle = \langle \nabla \log s(\overline{\theta}_w[m, j_{m,w}^*]), \nabla \mathcal{L}[m, :] \rangle \\ =& \langle \nabla \log s(\overline{\theta}_w[m, j_{m,w}^*]), \nabla \log s(\overline{\theta}_w[m, j_{m,w}^*]) - \nabla \log s(\overline{\theta}_w[m, j_{m,l}^*]) \rangle \\ =& \left(\sum_{j' \neq j_{m,w}^*} s_w[m, j']^2 \right) + (1 - s_w[m, j_{m,w}^*])^2 \\ &- \left(\sum_{j' \neq j_{m,w}^*, j' \neq j_{m,l}^*} s_w[m, j']^2 \right) + s_w[m, j_{m,w}^*](1 - s_w[m, j_{m,w}^*]) + s_w[m, j_{m,l}^*](1 - s_w[m, j_{m,l}^*]) \\ =& 1 + (s_w[m, j_{m,l}^*] - s_w[m, j_{m,w}^*]) \geq 0, \end{aligned} \tag{21}$$

where the last inequality is true because $s \in [0,1]$. Here, basically, this margin loss will just encourage increase the chosen logP (and reduce the rejected one) for the selected token.

Chosen response: if $i \neq m$, we have

$$\begin{aligned} &\Delta \log s(\overline{\theta}_w[i, j_{i,w}^*]) \approx \sum_{i'=1}^{L} \langle \nabla \log s(\overline{\theta}_w[i, j_i^*]), \nabla \mathcal{L}[i', :] \rangle = \langle \nabla \log s(\overline{\theta}_w[i, j_i^*]), \nabla \mathcal{L}[i, :] \rangle \\ =& \langle \nabla \log s(\overline{\theta}_w[i, j_i^*]), \nabla \log s(\overline{\theta}_w[i, j_i^*]) - \nabla \log s(\overline{\theta}_l[i, j_i^*]) \rangle \\ =& (1 - s_w[i, j_i^*])(s_l[i, j_i^*] - s_w[i, j_i^*]) - \sum_{j' \neq j_i^*} s_w[i, j'](s_l[i, j'] - s_w[i, j']) \end{aligned} \tag{22}$$

Here, basically, the loss can only pick one direction to change both chosen and rejected entry.

Connection to derivation in Pal et al. (2024). The assumption in Pal et al. (2024) mainly ensures the sign of (22). Basically, smaug's assumption ensures that for $i \in [m+1, L]$, $s_w[i, j_i^*] \geq s_l[i, j_i^*]$

and $s_w[i,j] \leq s_l[i,j]$ for $j \neq j_i^*$.

$$\nabla \log s(\overline{\theta}_w[i,j_i^*]) - \nabla \log s(\overline{\theta}_l[i,j_i^*]) = \begin{bmatrix} s_l[i,1] - s_w[i,1] \\ \vdots \\ s_l[i,j_i^*] - s_w[i,j_i^*] \\ \vdots \\ s_l[i',V] - s_w[i',V] \end{bmatrix} = \begin{bmatrix} \geq 0 \\ \vdots \\ \leq 0 \\ \vdots \\ \geq 0 \end{bmatrix}$$

For (22), we have

$$(1 - s_w[i,j_i^*])(s_l[i,j_i^*] - s_w[i,j_i^*]) - \sum_{j' \neq j_i^*} s_w[i,j'](s_l[i,j'] - s_w[i,j']) \leq 0.$$

This ensures the chosen token will have reduced logP.

Condition on chosen tokens increasing and rejected token decreasing at m, and on chosen and rejected tokens decreasing after $m+1$:

$$\begin{aligned} &(21) \geq 0 \text{ always holds}, \\ &\forall i \in [m+1, L],\ s_w[i,j_i^*] \geq s_l[i,j_i^*],\ \forall j \neq j_i^*, s_w[i,j] \leq s_l[i,j] \implies (22) \leq 0 \end{aligned}$$

C Experiment details

C.1 Hardware and Software Setup

Our experiments were implemented using TRL version 0.11.0. The training was performed on a hardware setup consisting of two NVIDIA H100 GPUs, providing substantial computational power for the training process.

C.2 TL;DR Task Setup

For the TL;DR summarization task, we utilized the CarperAI/openai_summarize_comparisons dataset. We employed two LLMs for this task:

- mistralai/Mistral-7B-Instruct-v0.3 (referred to as Mistral 7B)
- meta-llama/Meta-Llama-3-8B-Instruct (referred to as Llama-3 8B)

We did not perform any supervised fine-tuning step prior to the RLHF training for these models.

To optimize the training process, we applied Low-Rank Adaptation (LoRA) with a rank of 64 to both models. The learning rate was set at 5×10^{-6} for all RLHF training.

C.3 RLHF Algorithm Configurations

We implemented several RLHF algorithms, each with its own specific configurations:

- Direct Preference Optimization (DPO): $\beta = 0.1$
- Chosen NLL term (used in CPO, RRHF, and SLiC-HF): $\lambda = 1$
- SLiC-HF: $\delta = 1$
- SimPO: $\gamma = 0.5$
- R-DPO: $\alpha = 0.2$
- DPOP: $\lambda = 50$

C.4 Sentiment Analysis Task Setup

For the sentiment analysis task, we used a specially curated sentiment dataset. Unlike the TL;DR task, we performed supervised fine-tuning on the GPT-2 model before proceeding with the RLHF training. The learning rate for this RLHF training was also set to 5×10^{-6}.

C.5 Additional empirical results

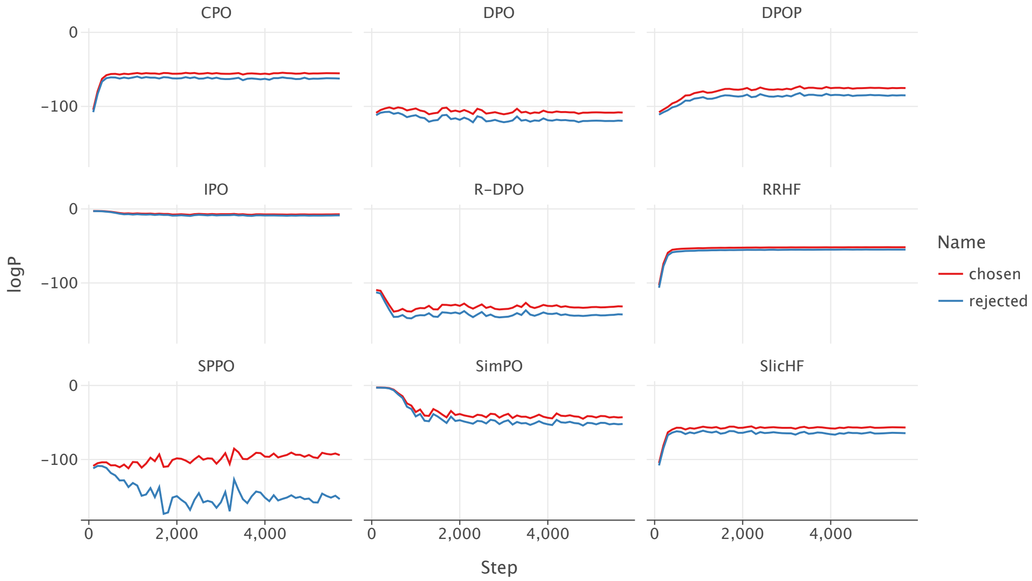


Figure 5: Training dynamics of the chosen and rejected log probabilities on the TL;DR dataset for different preference optimization algorithms trained on Llama-3 8B. All algorithms exhibit synchronized increases and decreases in the chosen and rejected log probabilities. Note: For SimPO and IPO, the log probabilities are normalized, while in the other plots, they are the original log probabilities.

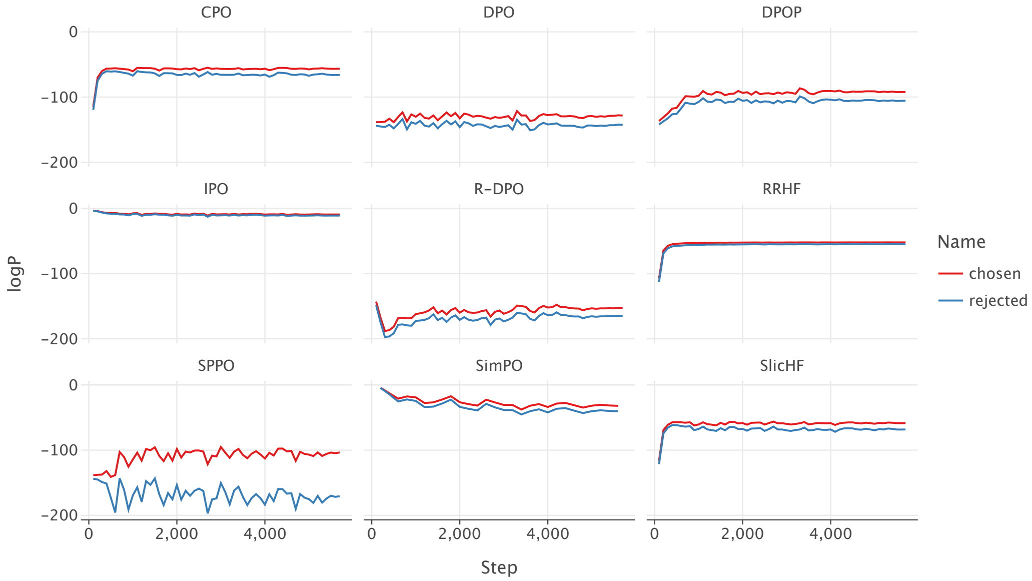


Figure 6: Training dynamics of the chosen and rejected log probabilities on the TL;DR dataset for different preference optimization algorithms trained on Mistral 7B. All algorithms exhibit synchronized increases and decreases in the chosen and rejected log probabilities. Note: For SimPO and IPO, the log probabilities are normalized, while in the other plots, they are the original log probabilities.

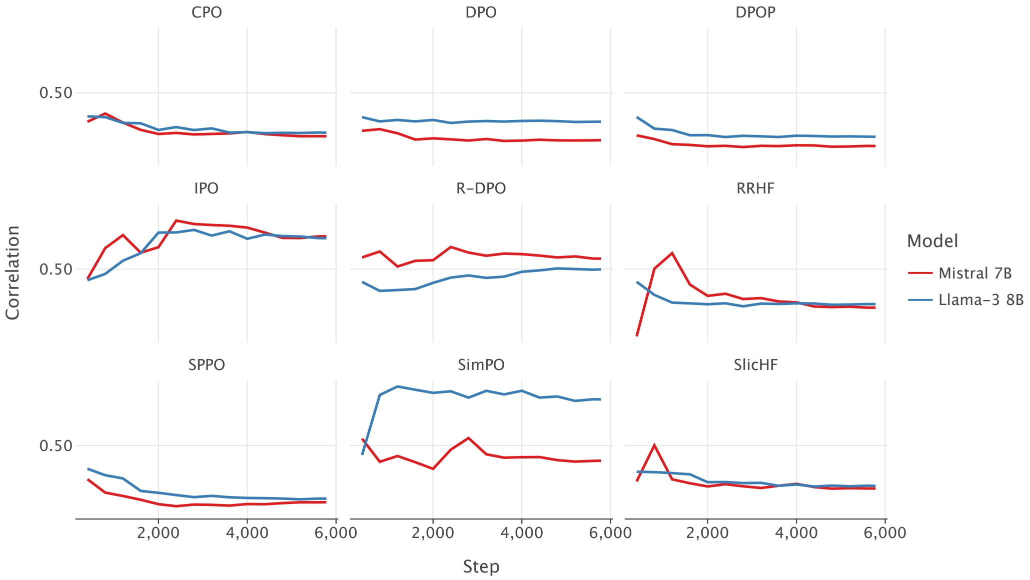


Figure 7: Cosine similarity between $\nabla_\theta \log \pi_w$ and $\nabla_\theta \log \pi_l$ on the TL;DR dataset for different preference optimization algorithms trained on Llama-3 8B and Mistral 7B.