A Common Pitfall of Margin-based Language Model Alignment: Gradient Entanglement

Hui Yuan♠[1], Yifan Zeng♠[2], Yue Wu♠[3], Huazheng Wang[4], Mengdi Wang[5], Liu Leqi♠[6]

[1,3,5]Princeton University
[2,4]Oregon State University
[6]The University of Texas at Austin[*†]

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. In this paper, we identify a common pitfall of margin-based methods—the **under-specification** of ideal LM behavior on preferred and dispreferred responses individually, which results in 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) The probability of preferred responses may decrease, even when those responses are ideal. We demystify the reasons behind these problematic behaviors: margin-based losses couple the change in the preferred probability with the gradient of the dispreferred one, and vice versa, often preventing the preferred probability from increasing while the dispreferred one decreases, and thus causing a synchronized increase or decrease in both probabilities. We term this effect, inherent in margin-based objectives, **gradient entanglement**. Formally, we derive conditions for general margin-based alignment objectives under which gradient entanglement becomes concerning: *the inner product between the gradient of preferred log-probability and the gradient of dispreferred log-probability is large relative to the individual gradient norms*. We theoretically investigate why such inner products can be large when aligning language models and empirically validate our findings. Empirical implications of our framework further extend to explaining important differences in the training dynamics of various preference optimization algorithms, and suggesting potential algorithm designs to mitigate the under-specification issue of margin-based methods and thereby improving language model alignment.[3]

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 [13, 2, 8]. 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

[*]♠: Leading Contributors. Corresponding to: `huiyuan@princeton.edu`, `leqiliu@utexas.edu`.
[†]Work done in part at Princeton Language & Intelligence.
[3]Code for the paper can be found at https://github.com/HumainLab/Understand_MarginPO.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

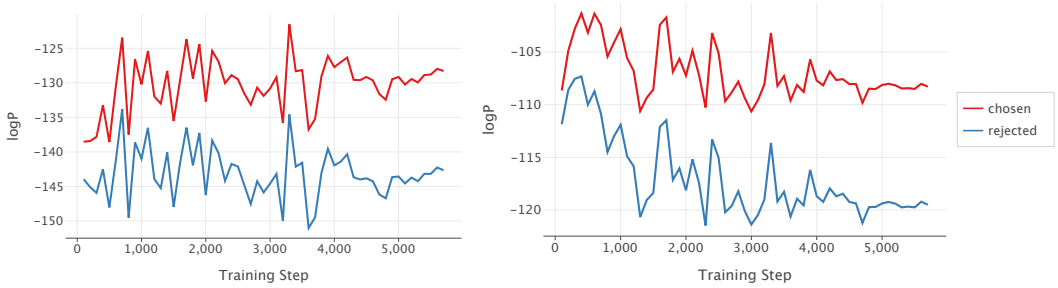
contrast between chosen and rejected responses, with the goal of helping the model learn distinctions between good and bad behaviors.

Leveraging the structure of the RLHF 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 [11, 1, 16, 4, 5, 9, 10, 18, 6, 19, 15]:[4]

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

where for a language model π_θ, $\log \pi_\theta(y_w|x)$ specifies the log-probability of the chosen response y_w[5] and $\log \pi_\theta(y_l|x)$ specifies that of the rejected response y_l, given the same prompt x. Most of the existing preference optimization losses can be interpreted as varying the scalar functions m, h_w, h_l (Section B and Table 2). At the core, they all rely on the *margin* between the chosen log-probability $\log \pi_\theta(y_w|x)$ and the rejected log-probability $\log \pi_\theta(y_l|x)$.

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).



(a) Mistral 7B

(b) Llama-3 8B

Figure 1: Training dynamics of the chosen and rejected log probabilities on the TL;DR dataset [13]. 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 [13]. In such cases, the ideal behavior of the model on the chosen samples is less clear—aside from being preferred over the rejected ones.

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). We further investigate why the gradient inner product can be relative large to individual norms when

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

[5]Subscript w in chosen response y_w stands for "winner", l in y_l stands for "loser."

aligning a model using language data. 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 (Appendix A.1, A.2).

To summarize, our contributions are as follows:

- We identify a fundamental issue with margin-based preference optimization: it under-specifies the ideal behavior of the LM on chosen and rejected responses individually, which often results in synchronized increase/decrease in the chosen and rejected log-probabilities (Section 1);
- We uncover that gradient entanglement is the inherent cause of the pitfalls in margin-based objectives, and provide a general gradient inner product condition that captures when the synchronized movement of chosen and rejected log probabilities occurs (Section 3);
- We investigate the gradient inner product and explore when the condition may fail and the synchronized movement occurs theoretically and experimentally (Appendix A).
- Using our framework, we outline two potential approaches to mitigate issues caused by gradient entanglement: one based on normalized gradients (Section 4.1) and the other leveraging token-level information (Section 4.2).

2 Background and Related Work

2.1 Preference optimization

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 Existing methods

There have been plenty of works on the design of preference optimization losses, motivated by various assumptions or considerations. Here we briefly review them and discuss their connection to the probability **margin**:

Rafailov et al. [11] derive the DPO loss from the KL-constrained reward maximization problem:

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

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:

$$\ell_{\text{DPO}}(x, y_w, y_l; \theta; \pi_{\text{ref}}) := -\log\sigma\left(\beta\left[\log\left(\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)}\right) - \log\left(\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right]\right). \tag{2}$$

Motivated by non-transitive human preference and language model calibration respectively, Azar et al. [1] and Zhao et al. [19] propose IPO and SlicHF loss with similar forms 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. [10] 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. [6] and Yuan et al. [18] consider the setting of average rewards and derive a loss dependent on the **length-normalized margin** $\frac{1}{|y_w|}\log\pi_\theta(y_w|x) - \frac{1}{|y_l|}\log\pi_\theta(y_l|x)$.

3 Gradient Entanglement

Margin-based preference optimization often results in synchronized increase/decrease in chosen and rejected log-probabilities (Section 1). Our key finding is that the synchronized change is caused by an effect we term as gradient entanglement. Starting with a case study on DPO in Section 3.1, we formally define the gradient entanglement effect, from the definition we will see the entanglement is

passed through the inner product between chosen and rejected gradients. We derive conditions on such inner product under which the gradient entanglement causes concerning synchronized change. We validate our findings empirically in Section 3.2.

3.1 Case study: gradient entanglement in DPO

Let us start with deriving the gradient of the DPO objective (2). To simplify the formula of DPO gradient, we define the implicit reward $\hat{r}_\theta(x, y) := \beta \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}$ (which is a scalar) and introduce the notations:

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

Then considering a single sample (x, y_w, y_l), the DPO gradient can be rewritten as[6]

$$\nabla_\theta \ell_{\text{DPO}} = -\beta c(\theta) \cdot (\nabla_\theta \log \pi_w(\theta) - \nabla_\theta \log \pi_l(\theta)). \tag{3}$$

Suppose $\eta > 0$ is the step size for minimizing the DPO objective and let $C = \eta\beta c(\theta)$. After one step gradient descent with (3), a simple analysis of the log-probability change in chosen and rejected responses uncovers the intriguing **gradient entanglement** effect as follows:

Gradient Entanglement (DPO)

The chosen log-probability change $\Delta \log \pi_w$ depends on the rejected gradient $\nabla \log \pi_l$, and similarly, the rejected log-probability change $\Delta \log \pi_l$ depends on the chosen gradient $\nabla \log \pi_w$:

$$\Delta \log \pi_w \approx C \cdot \left(\|\nabla \log \pi_w\|^2 - \langle \nabla \log \pi_w, \nabla \log \pi_l \rangle\right), \tag{4}$$

$$\Delta \log \pi_l \approx C \cdot \left(\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle - \|\nabla \log \pi_l\|^2\right). \tag{5}$$

(4) and (5) are derived by approximating $\Delta \log \pi_w$ and $\Delta \log \pi_l$ with first-order Taylor expansion (Appendix C.1). Beyond the DPO objective, the gradient entanglement effect is an inherent characteristic of margin-based objectives as the chosen and rejected log-probability are coupled in the definition of "margin." In Appendix B, we will formally derive gradient entanglement for general margin-based objectives for preference optimization. From the above definition, we can see that the entanglement effect is passed through the inner product $\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle$ between chosen and rejected gradients. In the absence of $\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle$, the log-probability changes $\Delta \log \pi_w$ and $\Delta \log \pi_l$ will not depend on each other. In the sequel, we will derive conditions on this inner product under which the gradient entanglement will have concerning effects.

3.1.1 When will the gradient entanglement be concerning?

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

$$\Delta(\log \pi_w - \log \pi_l) \approx C \cdot (\|\nabla \log \pi_w\|^2 - 2\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle + \|\nabla \log \pi_l\|^2) \geq 0,$$

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, due to the gradient entanglement effect, to individually ensure the increment of $\log \pi_w$ and the decrement of $\log \pi_l$, the inner product between chosen and rejected gradient should satisfy conditions listed in Condition 1. We will refer to Condition 1 as "gradient condition" as it is imposed on the inner product of gradients.

Condition 1 (Gradient condition for DPO)**.** *In DPO, to increase* $\log \pi_w$ *and decrease* $\log \pi_l$ *individually,* (4) *and* (5) *imply the following conditions:*

$$\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle \leq \|\nabla \log \pi_w\|^2 \iff \Delta \log \pi_w \geq 0, \log \pi_w \textit{ increases};$$

$$\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle \leq \|\nabla \log \pi_l\|^2 \iff \Delta \log \pi_l \leq 0, \log \pi_l \textit{ decreases}.$$

Based on the two conditions above, in Table 1 we summarize three cases that depict all possible changes on the chosen and rejected log-probabilities and are categorized by the value of $\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle$.

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

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. $\uparrow$ and $\downarrow$ indicate increase and decrease. **Case 1 (Ideal)**: $\log \pi_w$ increases and $\log \pi_l$ decreases; **Case 2**: $\log \pi_w$ and $\log \pi_l$ both decreases but $\log \pi_l$ decreases more; **Case 3**: $\log \pi_w$ and $\log \pi_l$ both increases but $\log \pi_w$ increases more.

In Appendix B, we show for other margin-based preference optimization loss, a similar condition on $\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle$ can be derived, but the condition could be more lenient than that of DPO for some specific losses, explaining why the training dynamics of those methods may differ from DPO.

3.2 Empirical observations

We conduct experiments on the TL;DR dataset [13] to showcase the widely-existing phenomenon that the chosen and rejected log-probabilities have synchronized changes during preference optimization. In addition, Figure 1 depicts how different margin-based preference optimization algorithms influence the log-probability of chosen and rejected responses.

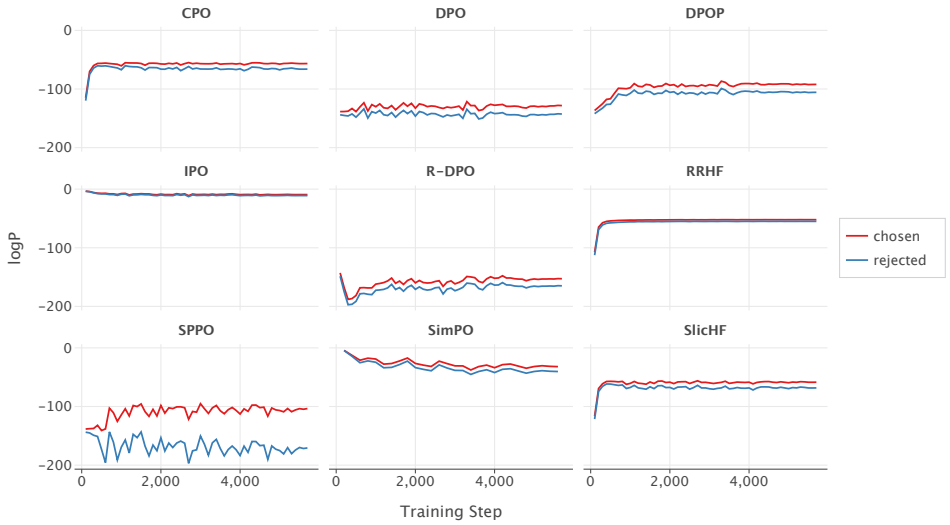


Figure 2: Training dynamics of the chosen and rejected log-probabilities on the TL;DR dataset for different algorithms trained on Mistral 7B. The corresponding plot for Llama3 8B is in Figure 5 (Appendix E.5). For SimPO and IPO, the log-probabilities are normalized by the response length, while in the other plots, the log-probabilities are of entire responses. All algorithms exhibit synchronized increases and decreases in the chosen and rejected log-probabilities. We also provide the cosine similarity plots between $\nabla_\theta \log \pi_w$ and $\nabla_\theta \log \pi_l$ in Appendix E.5 (Figure 6).

For **DPO** and **R-DPO**, both the chosen and rejected log-probabilities tend to decrease simultaneously. This behavior proofs the existence of gradient entanglement, showing that methods purely dependent on the margin might result in both terms decreasing, with the rejected log-probability decreasing more significantly. This leads to an increase in the margin, which is the original learning objective, but not necessarily an increase in the chosen log-probability.

SPPO demonstrates a distinct trend where the log-probability of the chosen responses increases, while the log-probability of the rejected responses decreases. This matches the theoretical intuition obtained from the specialized gradient conditions for SPPO in Appendix B.

For **CPO**, **DPOP**, **RRHF**, and **Slic-HF**, algorithms with explicit regularization on the chosen log-probability, we observe a consistent increase in the log-probability of the chosen responses. This behavior reflects the effect of explicit regularizations in increasing the chosen log-probability, which also aligns with the conditions discussed in Appendix B.

SimPO and **IPO**[7] in Figure 1 report the *average* log-probability of responses. The simultaneous decrease in both the (average) chosen and rejected log-probabilities is expected, because the loss only 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 due to the gradient entanglement effect.

Overall, experimental results on various margin-based losses closely align with our analysis on the gradient entanglement and the gradient conditions outlined in Appendix B, demonstrating how loss structures, explicit regularization, length-normalization and other design choices influence the dynamics of preference optimization.

4 Empirical Implications: Algorithmic Design and More

Using our insights from the gradient inner product conditions (Section 3) and our investigation on when such conditions may be violated (Appendix A), we present two potential ways to mitigate gradient entanglement, thus allowing the chosen and rejected probability to change in different directions simultaneously.

4.1 Design 1: pairwise normalized gradient descent

As discussed in Section 3, to specify an increasing log-probability of the chosen response and a decreasing log-probability of the rejected response, we can set $d_w/d_l = \|\nabla\log\pi_l\|/\|\nabla\log\pi_w\|$ so that (12) and (13) will hold simultaneously. This leads to the following gradient update rule:

$$\nabla_\theta\ell := C\left(\frac{\nabla_\theta\log\pi_w}{\|\nabla_\theta\log\pi_w\|} - \frac{\nabla_\theta\log\pi_l}{\|\nabla_\theta\log\pi_l\|}\right),$$

where C is a quantity relying on the specific preference optimization loss design. This update rule turns out to be the normalized gradient for the chosen and rejected responses respectively. For example, we can modify the gradient update for the DPO loss as:

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

and adjust the learning rate accordingly.

4.2 Design 2: sparsity regularized token masking

An alternative approach to reduce gradient entanglement is by designing a fine-grained margin-based loss that only contrasts **significant tokens**, as suggested in Section A.3. For example, the following loss design could be a potential good candidate for adapting the original DPO objective in this direction:

$$\begin{aligned}\ell(\theta,u_w,u_l) = &-\log\sigma\left(\sum_{i=1}^{L}\mathbb{I}\{u_w^i\geq r\}\log\frac{\pi_\theta(y_w^i|x,y_w^{<i})}{\pi_{\text{ref}}(y_w^i|x,y_w^{<i})} - \mathbb{I}\{u_l^i\geq r\}\log\frac{\pi_\theta(y_l^i|x,y_w^{<i})}{\pi_{\text{ref}}(y_l^i|x,y_l^{<i})}\right)\\ &+\eta\left(\|\mathbb{I}\{u_w\geq r\}\|_1 + \|\mathbb{I}\{u_l\geq r\}\|_1\right),\end{aligned}$$

where $\eta\in\mathbb{R}_+, r\in\mathbb{R}$ are hyper-parameters and $u_w\in\mathbb{R}^L, u_l\in\mathbb{R}^L$ are learnable weights depending on $(x,y_w^{<i}),(x,y_l^{<i})$ respectively, interpreted as the confidence in considering token i significant. In practice, we can approximate the indicator $\mathbb{I}\{u^i\geq r\}$ with the sigmoid function $\sigma(k\cdot(u^i-r))$ for large $k>0$. The loss is inspired by sparsity-related ideas (e.g., LASSO [14]), where the learnable masks $\mathbb{I}\{u^i\geq r\}$ ideally pick out the significant tokens in each response that enlarge the margin. The ℓ_1 regularizer on the token-wise mask imposes sparsity on it. Other variants of preference optimization objectives may also adopt similar sparsity-related adaptations to leverage token-wise information in obtaining the margin.

[7] In their original paper, Azar et al. [1] proposed the IPO loss without average log-probability. The authors later claimed using average log-probability with IPO yields improved performance.

4.3 Further Discussion

In this paper, we touch upon a common pitfall of margin-based preference optimization methods in language alignment: it under-specifies the ideal behavior of the LM on the chosen and rejected responses individually. Due to the gradient entanglement effect, our gradient inner product condition suggests that when the chosen and rejected gradients are highly correlated, their log probabilities will exhibit synchronized increases/decreases. Beyond explaining differences in existing variants of margin-based methods and proposing new algorithmic designs to address gradient entanglement, our framework of gradient entanglement offers a fresh perspective to understand existing avenues of RLHF methods:

1. The first group of methods implicitly adjust the criterion on the maximum size of the gradient inner product under which the synchronized changes do not occur, without directly modifying the gradient inner product, as seen in the works listed in Table 2. Our proposal in Section 4.1 falls under this category.
2. The second group of methods modifies the inner product of interest directly. As discussed in Appendix A, 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 [7, 12, 17], exploiting token-level contrasts to improve RLHF falls under the second category, and so does our proposal in Section 4.2.
3. For the RLHF procedure that involves reward modeling and policy optimization as separate stages, the objectives for reward model learning also suffer from under-specification due to gradient entanglement. However, there is a key difference: LM is not directly updated based on preference samples. Instead, we use on-policy samples from the LM to perform policy optimization, where responses with positive rewards are not necessarily the ones we want to increase or maintain the LM's probability on. A precise characterization of how the under-specification manifests in this procedure will be the subject of future investigation.

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

References

[1] 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*, pages 4447–4455. PMLR, 2024.

[2] 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.

[3] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. Ultrafeedback: Boosting language models with scaled ai feedback. In *Forty-first International Conference on Machine Learning*, 2024.

[4] 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.

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

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

[15] 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.

[16] 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.

[17] Eunseop Yoon, Hee Suk Yoon, SooHwan Eom, Gunsoo Han, Daniel Wontae Nam, Daejin Jo, Kyoung-Woon On, Mark A Hasegawa-Johnson, Sungwoong Kim, and Chang D Yoo. Tlcr: Token-level continuous reward for fine-grained reinforcement learning from human feedback. *arXiv preprint arXiv:2407.16574*, 2024.

[18] 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.

[19] 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 Investigation on Gradient Inner Product

Section 3 reveals that the gradient entanglement effect is driven by the key quantity: the inner product $\langle \nabla_\theta \log \pi_w, \nabla_\theta \log \pi_l \rangle$ between chosen and rejected log-probabilities (Condition 1, 2: gradient condition). As demonstrated in Section 3.2 and widely observed in practice, margin-based objectives are often triggered to not behave in the ideal way, suggesting that the gradient condition is violated due to a large gradient inner product. Therefore, in this section, we investigate into such inner product to understand why it can be large when aligning language models. Our investigation focuses on the representative margin-based objective DPO.

To build our theoretical intuition, we use synthetic toy settings to analyze the gradient inner product and the changes in log-probabilities. Our theory offers explanations from two perspectives: (1) when the gradient condition holds and which factors do not contribute to enlarging the gradient inner product (Theorem 1, Corollary 2) and (2) when the gradient condition is violated and which factors do cause the gradient inner product to grow, leading to a decrease in the chosen log-probability (Theorem 3). All proofs are provided in Appendix D and we empirically verify our theoretical insights in Section A.3.

A.1 Positive result: when the gradient condition holds

We first provide a positive result on when the gradient inner product is small, thus Condition 1 holds and DPO exhibits the ideal behavior that pushes up the log-probability of the chosen response and pushes down the log-probability of the rejected one. In the first synthetic setting, we analyze DPO for optimizing an LM with a learnable last linear layer in a single-token prediction task.

Model Setup 1 (LM with learnable last linear layer). *Let $V = |\mathcal{V}|$ be the vocabulary size. 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 y and $s : \mathbb{R}^V \to \Delta_{\mathcal{V}}$[8] denotes the softmax function. The hidden states are assumed as frozen during DPO.

Data Setup 1. *Both chosen and rejected responses contain only one token under the prompt x. That is, $y_w, y_l \in \mathcal{V}^1$, and $y_w[1] \neq y_l[1]$[9].*

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

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

with h being the hidden state of the last token in prompt x. Thus, both parts of Condition 1 hold, resulting in $\log \pi_w$ increases and $\log \pi_l$ decreases.

Theorem 1 shows that for single-token prediction, $\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle < 0$. This suggests that the gradient descent steps of DPO ensures $\log \pi_w$ increases and $\log \pi_l$ decreases. This result can be easily extended to the data setup where the chosen and rejected responses have multiple tokens but only differ at the last one, i.e., $y_w[1:L-1] = y_l[1:L-1]$, $y_w[L] \neq y_l[L]$ with L being the length of y_w and y_l. In this case, up to the L-th token where chosen and rejected differ, the hidden states are the same for the two responses. This is true because for $y_w[1:L-1] = y_l[1:L-1]$, we have that $h_i = h_{i,w} = h_{i,l}$ for $i \in [L]$.

Corollary 2. *Under Model Setup 1, the chosen and rejected responses only differ at their last token, assume after SFT the model prediction on the L-th token in response is uniformly concentrated on $M \leq V$ tokens in the vocabulary, we have*

$$\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle \leq \|\nabla \log \pi_w\|^2 = \|\nabla \log \pi_l\|^2,$$

and thus $\log \pi_w$ increases and $\log \pi_l$ decreases.

[8] Here, Δ denote the probability simplex.

[9] 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.

A.2 Negative result: when the gradient condition is violated

From the previous results, we can see that 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. To gain theoretical insights on what causes the violation of the condition, we level up our previous data setup to the following.

Data Setup 2. *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 adopt a simpler setting for parameterizing the LM, where the LM has learnable logits.

Model Setup 2 (LM with learnable logits). *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. The vectors* $s_{w,i}$ *and* $s_{l,i}$ *are configured as variables to optimize in the model and to which we take the 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. [9], 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 gradient descent step.

Theorem 3. *Under Model Setup 2 and Data Setup 2, 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 chosen log-probability before the differing token stays unchanged:*

$$\Delta \log \pi(y_w^i \mid x, y_w^{<i}) \approx 0. \tag{6}$$

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

$$\Delta \log \pi(y_w^m \mid x, y_w^{<m}) \approx 1 + (s_{w,m}[j^*] - s_{w,m}[k^*]) \geq 0. \tag{7}$$

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

$$\Delta \log \pi(y_w^i \mid x, y_w^{<i}) \approx (1 - s_{w,i}[j_i^*])(s_{l,i}[j_i^*] - s_{w,i}[j_i^*]) - \sum_{j \neq j_i^*} s_{w,i}[j](s_{l,i}[j] - s_{w,i}[j]) \leq 0, \tag{8}$$

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* (6)*,* (7) *and* (8)*, as the same suffix following the differing tokens gets longer,* $\log \pi_w$ *decreases more.*

Remark 4. *While Theorem 3 adopts the same assumptions made in Pal et al. [9], we precisely characterize the per-token log-probability changes based on the first-order approximation, and explicitly break down the sentence-wise probability change for 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 chosen probability observed in experiments (Figure 3).*

It is worth mentioning that Theorem 3 explicitly presents the size of probability changes. The same conclusion on the change direction can also be derived with a per-token gradient inner product condition similar to Condition 1, see Appendix D.2. The increase of chosen presented in (7) follows the same intuition in Theorem 1 that if two contrastive tokens are picked by chosen and rejected responses under a similar context, then the chosen token probability will increase while the rejected decreases. An intuitive explanation of what causes the decrease of both the chosen and rejected in

(8) could be that the chosen and rejected gradients are highly correlated as they pick the same token under a similar context. Mathematically, the assumption we adopted implies that the gradient inner product between chosen and rejected can be lower bounded.

Combining our insights gained in Section A.1 and A.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.[10] Below we verify our theoretical findings empirically.

A.3 Empirical observations

We empirically verify our theoretical intuition regarding when the gradient condition may be held or violated, by aligning GPT-2 small to a curated sentiment preference dataset.

The preference dataset is curated from `mteb/tweet_sentiment_extraction`: for a data point (x, y_w, y_l), prompt x is a statement, e.g., "1 week to my Birthday!" The chosen response y_w reflects the true sentiment label of x. We filter statements in the original dataset and only retain those with binary sentiments: "**positive**" or "**negative**", and set the rejected response y_l to reflect the flipped wrong sentiment label of x. We curate four datasets with the following styles of responses:

- **single token** (Data Setup 1): y_w: Positive. y_l: Negative.
- **short suffix**: y_w: Positive sentiment. y_l: Negative sentiment.
- **long suffix**: y_w: Positive sentiment based on my judgement. y_l: Negative sentiment based on my judgment.
- **prefix+suffix** (Data Setup 2): y_w: It has a positive sentiment based on my judgement. y_l: It has a negative sentiment based on my judgment.

Our theoretical results suggest: (1) in the **single token** case, the chosen and rejected gradients will have negative inner product and thus DPO will allow the chosen log-probability to increase while the rejected to decrease (Theorem 1). (2) For the **short suffix** and **long suffix** cases, we expect DPO to reduce the chosen log probability more for the latter, as responses in **long suffix** contain more tokens following the differing spot, leading to more chosen tokens with decreasing log probability (Theorem 3). Additionally, (3) for the differing token ("positive" or "negative"), the token-wise gradient inner product would be negative, while for other identical tokens, the token-wise gradient inner product would be positive.

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

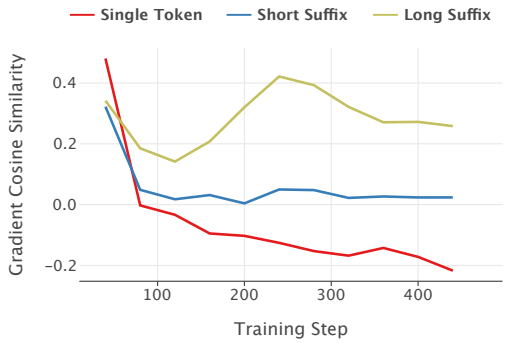
The three implications obtained from our theorems are validated by empirical observation. First, the chosen log probability increases only in the **single token** case, and the **short suffix** chosen log probability decreases less than that of the **long suffix**, aligning with our theoretical results (Figure 3). Second, the gradient cosine similarity in the **single token** case quickly declines and stays negative during training, while that in the **short suffix** and **long suffix** is positive and increases as the suffix length (i.e., the number of identical tokens after the difference) grows (Figure 4a). This aligns with our gradient condition (Condition 1), 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 for the **prefix+suffix** case. From the heat map of token-wise gradient similarities (Figure 4b), we observe that on the diagonal, the inner product between the gradients on the tokens "positive" and

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

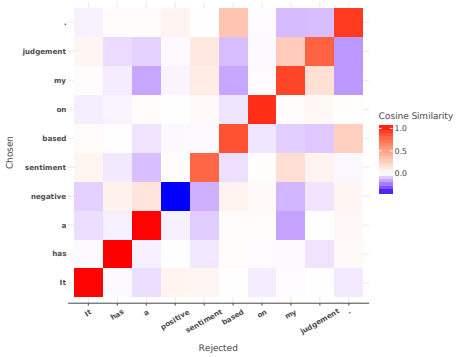
"negative" is below 0, whereas for other identical tokens in the two responses, the gradient cosine similarities are significantly higher and close to 1 for some tokens.

Our theoretical and empirical investigation into the token-level gradient inner product suggests broader implications for general alignment tasks. **Significant tokens** (e.g., "positive"/"negative") contrasting the chosen and rejected responses the most, exhibit negative gradient correlation and prevent gradient entanglement. Meanwhile, those non-contrastive **insignificant tokens** (e.g., identical tokens) cause gradient entanglement due to the high similarity in their gradients.

This insight highlights the importance of token-level gradient dynamics and their contribution to the entanglement effect, motivating a fine-grained alignment method that contrasts only the significant tokens in the chosen/rejected response pair. This approach retains the simplicity of margin-based methods while potentially reducing the gradient entanglement effect in margin-based losses. Further details on the potential algorithm design are discussed in Section 4.2.



(a) Cosine similarity between $\nabla_\theta \log \pi_w$ and $\nabla_\theta \log \pi_l$.



(b) Token-wise gradient cosine similarity.

Figure 4: Gradient cosine similarity behaviors on the sentence-level and token-level for sentiment tasks. Figure 4a gives the cosine similarity between $\nabla_\theta \log \pi_w$ and $\nabla_\theta \log \pi_l$ for DPO on **single token**, **short suffix** and **long suffix** datasets, defined as: $\frac{\langle \nabla_\theta \log \pi_w, \nabla_\theta \log \pi_l \rangle}{\|\nabla_\theta \log \pi_w\| \|\nabla_\theta \log \pi_l\|}$. Figure 4b shows the token-wise gradient similarity for an instance in the **prefix+suffix** task.

B General gradient entanglement effect

We now move on to the general margin-based loss (1). Here, we additionally consider regularizers used in these losses:

$$\ell(\theta) = -\Big(m(h_w(\log \pi_w) - h_l(\log \pi_l)) + \Lambda(\log \pi_w)\Big), \tag{9}$$

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

	$m(a)$	$h_w(a)$	$h_l(a)$	$\Lambda(a)$
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 losses. Constants satisfy $\beta, \gamma, \delta, \lambda_w, \lambda_l > 0$.

Based on this unified formulation of preference optimization objectives (9), we derive general gradient entanglement for all margin-based losses (derivations in Appendix C.1):

Gradient Entanglement (General)

The chosen log-probability change depends on the rejected gradient, and vice versa. The mutual dependency is characterized by:

$$\Delta \log \pi_w \approx \eta \left(d_w \|\nabla \log \pi_w\|^2 - d_l \langle \nabla \log \pi_w, \nabla \log \pi_l \rangle \right),$$
$$\Delta \log \pi_l \approx \eta \left(d_w \langle \nabla \log \pi_w, \nabla \log \pi_l \rangle - d_l \|\nabla \log \pi_l\|^2 \right).$$

In the general form of gradient entanglement, d_w and d_l are scalars defined as

$$d_w := m'(h_w(\log \pi_w) - h_l(\log \pi_l)) h_w'(\log \pi_w) + \Lambda'(\log \pi_w), \tag{10}$$
$$d_l := m'(h_w(\log \pi_w) - h_l(\log \pi_l)) h_l'(\log \pi_l). \tag{11}$$

We derive a generalized version of DPO's gradient condition (Condition 1) for general margin-based losses.

Condition 2 (Gradient condition for general margin-based objectives)**.** *For margin-based preference optimization objectives*(9)*, the conditions for* $\log \pi_w$ *to increase and for* $\log \pi_l$ *to decrease are:*

$$\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle \leq \frac{d_w}{d_l} \|\nabla \log \pi_w\|^2 \iff \Delta \log \pi_w \geq 0, \log \pi_w \textit{ increases}; \tag{12}$$
$$\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle \leq \frac{d_l}{d_w} \|\nabla \log \pi_l\|^2 \iff \Delta \log \pi_l \leq 0, \log \pi_l \textit{ decreases}. \tag{13}$$

Accordingly, we can instantiate Condition 2 for different algorithms by using their specialized m, h_w, h_l, Λ in Table 2. Note that between conditions (12) and (13), for one condition to be more lenient (e.g., if $d_w/d_l > 1$ in the chosen condition), the other condition becomes more strict (then $d_l/d_w < 1$ in the rejected condition). When $\nabla \log \pi_w$ and $\nabla \log \pi_l$ have similar norms and are positively correlated, it is likely that one of (12) and (13) holds while the other fails, explaining why it is easy to observe a simultaneous increase or decrease in the probabilities of chosen and rejected responses.

This general gradient inner product condition also suggests an interesting new algorithm to achieve our ideal case: we can reweigh the chosen and rejected log-probabilities in the margin-based loss such that $d_w/d_l = \|\nabla \log \pi_l\| / \|\nabla \log \pi_w\|$, which ensures that both parts in Condition 2 are satisfied at the same time. We provide more discussion on a potential algorithm design inspired by this observation in Section 4.1.

B.1 How do other margin-based methods work differently from DPO?

Utilizing the gradient condition we derived, we provide in the following a brief discussion on some existing preference optimization algorithms and explain why these algorithms may work differently from DPO under certain settings.

- **DPO**: $\frac{d_w}{d_l} = \frac{d_l}{d_w} = 1$, reproducing the Condition 1.
- **SPPO**: $\frac{d_w}{d_l} = \frac{\beta^{-1} - \log \pi_w}{\beta^{-1} + \log \pi_l} > 1$[11], where β^{-1} is a large constant. Compared with DPO, SPPO loss ensures that it is easier for $\log \pi_w$ to increase based on (12) and harder for $\log \pi_l$ to decrease due to (13).
- **KTO**: $\frac{d_w}{d_l} \propto \frac{\lambda_w}{\lambda_l}$, where λ_w, λ_l are two hyperparameters in KTO, fine-tuned according to different tasks and datasets. Thus no general conclusion on the chosen/rejected probability change can be made from our conditions.

[11] See Section C.2 for the derivation.

- **Explicit regularization on chosen log-probability** (**CPO**, **DPOP**[12], **RRHF** and **Slic-HF**): According to the formulas of d_w and d_l in (10) and (11), the negative log-likelihood (NLL) regularizer on chosen responses enlarges d_w while having no influence on d_l as $\Lambda' \geq 0$ and only appears in (10). As a result, larger $\frac{d_w}{d_l}$ makes condition (12) more lenient and thus the chosen log-probability is more likely to increase.

- **Length-normalization** (**SimPO**, **RRHF** and **IPO**): In SimPO, $\frac{d_w}{d_l} = \frac{|y_l|}{|y_w|}$ and condition (12) and (13) can be rewritten as:

$$\left\langle \frac{\nabla \log \pi_w}{|y_w|}, \frac{\nabla \log \pi_l}{|y_l|} \right\rangle \leq \left\| \frac{\nabla \log \pi_w}{|y_w|} \right\|^2; \quad \left\langle \frac{\nabla \log \pi_w}{|y_w|}, \frac{\nabla \log \pi_l}{|y_l|} \right\rangle \leq \left\| \frac{\nabla \log \pi_l}{|y_l|} \right\|^2. \tag{14}$$

  These conditions imply the following: to ensure increasing chosen log-probability while decreasing rejected log-probability, (14) should hold. This is more lenient than the corresponding condition posed for DPO that $\langle \nabla \log \pi_w, \nabla \log \pi_l \rangle \leq \min(\|\nabla \log \pi_w\|^2, \|\nabla \log \pi_l\|^2)$, when the length of chosen and rejected responses is biased, resulting in either the chosen or rejected gradient norm being significantly higher than the other. Therefore, compared to DPO, SimPO leans towards increasing the chosen probability and decreasing that of the rejected when the preference data is heavily length-biased. The same reasoning also applies to **RRHF** and **IPO**[13] for their length normalization design.

C Derivations for gradient entanglement and conditions in Section 3

C.1 Derivation for gradient entanglement

DPO. After one step of gradient descent with step size $\eta > 0$ for decreasing the loss ℓ_{DPO}, the change in the log-probability of the chosen response denoted by $\Delta \log \pi_w$, as well as the change in the log-probability of the rejected response denoted by $\Delta \log \pi_l$, can be approximated by the first-order Taylor expansion:

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

General Losses. First, the gradient of (9) can be written as

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

where d_w and d_l are scalars such that

$$d_w := m'(h_w(\log \pi_w) - h_l(\log \pi_l)) h_w'(\log \pi_w) + \Lambda'(\log \pi_w),$$
$$d_l := m'(h_w(\log \pi_w) - h_l(\log \pi_l)) h_l'(\log \pi_l).$$

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

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

C.2 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

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

[12] For DPOP, the regularizer is included in its $h_w(a)$ term in Table 2, due to its design to turn on/off the regularizer based on the value of chosen log-probability.

[13] In the TRL library, the implementation of IPO averages the log-probabilities by the number of tokens.

Then

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

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.

D Proofs for the Gradient Inner product in Section A

D.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 predicting the L-th token, $s(\cdot)$ is the softmax function.

$$\nabla_\theta \log\pi(y_w^L \mid x, y_w^{<L}) = \nabla_\theta\left(\log s(h_L^\top\theta)[y_w^L]\right) \tag{15}$$
$$\nabla_\theta \log\pi(y_l^L \mid x, y_l^{<L}) = \nabla_\theta\left(\log s(h_L^\top\theta)[y_l^L]\right) \tag{16}$$

Compute the gradient with chain rule,

$$\nabla_\theta \log\pi_w^L = [-s(1)h_L, \cdots, (1 - s(i_w))h_L, \cdots, -s(i_l)h_L, \cdots, -s(V)h_L] \tag{17}$$
$$\nabla_\theta \log\pi_l^L = [-s(1)h_L, \cdots, -s(i_w)h_L, \cdots, (1 - s(i_l))h_L, \cdots, -s(V)h_L], \tag{18}$$

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.

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

$$\nabla\log\pi_l^L = [-\frac{1}{M}h_L, \cdots, \underbrace{-\frac{1}{M}h_L}_{i_w-th}, \cdots \underbrace{\left(1 - \frac{1}{M}\right)h_L}_{i_l-th}, \cdots - \frac{1}{M}h_L] \tag{20}$$

$$\left\langle \nabla\log\pi_w^L, \nabla\log\pi_l^L\right\rangle = \frac{M-2}{M^2}\|h_L\|^2 - 2\cdot\frac{1}{M}\cdot\frac{M-1}{M}\|h_L\|^2 = -\frac{1}{M}\|h_L\|^2. \tag{21}$$

$\left\langle \nabla \log \pi_w^L, \nabla \log \pi_l^L \right\rangle$ 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 Condition 1:

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

D.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, $\overline{\theta}_w$ and $\overline{\theta}_l$ are learnable logits in LM. 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

$$\Delta \log s(\overline{\theta}_w[i, j_{i,w}^*]) \approx \sum_{i'=1}^{L} \langle \nabla \log s(\overline{\theta}_w[m, j_{m,w}^*]), \nabla \mathcal{L}[i', :] \rangle = \langle \nabla \log s(\overline{\theta}_w[m, j_{m,w}^*]), \nabla \mathcal{L}[m, :] \rangle$$

$$= \langle \nabla \log s(\overline{\theta}_w[m, j_{m,w}^*]), \nabla \log s(\overline{\theta}_w[m, j_{m,w}^*]) - \nabla \log s(\overline{\theta}_w[m, j_{m,l}^*]) \rangle$$

$$= \left(\sum_{j' \neq j_{m,w}^*} s_w[m, j']^2 \right) + (1 - s_w[m, j_{m,w}^*])^2$$

$$- \left(\sum_{j' \neq j_{m,w}^*, j' \neq j_{m,l}^*} s_w[m, j']^2 \right) + s_w[m, j_{m,w}^*](1 - s_w[m, j_{m,w}^*]) + s_w[m, j_{m,l}^*](1 - s_w[m, j_{m,l}^*])$$

$$= 1 + (s_w[m, j_{m,l}^*] - s_w[m, j_{m,w}^*]) \geq 0, \tag{22}$$

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

$$\Delta \log s(\overline{\theta}_w[i, j_{i,w}^*]) \approx \sum_{i'=1}^{L} \langle \nabla \log s(\overline{\theta}_w[i, j_i^*]), \nabla \mathcal{L}[i', :] \rangle = \langle \nabla \log s(\overline{\theta}_w[i, j_i^*]), \nabla \mathcal{L}[i, :] \rangle$$

$$= \langle \nabla \log s(\overline{\theta}_w[i, j_i^*]), \nabla \log s(\overline{\theta}_w[i, j_i^*]) - \nabla \log s(\overline{\theta}_l[i, j_i^*]) \rangle$$

$$= (1 - s_w[i, j_i^*])(s_l[i, j_i^*] - s_w[i, j_i^*]) - \sum_{j' \neq j_i^*} s_w[i, j'](s_l[i, j'] - s_w[i, j']) \tag{23}$$

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

Connection to the derivation in Pal et al. [9]. The assumption in Pal et al. [9] mainly ensures the sign of (23). 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 (23), 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 $m+1$:

$$\begin{aligned}&(22) \geq 0 \text{ always holds,}\\ &\forall i \in [m+1, L],\ s_w[i, j_i^*] \geq s_l[i, j_i^*],\ \forall j \neq j_i^*, s_w[i, j] \leq s_l[i, j] \implies (23) \leq 0\end{aligned}$$

E Experiment details

E.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.

E.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.

E.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: $\lambda = 50$

E.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}.

E.5 Additional empirical results

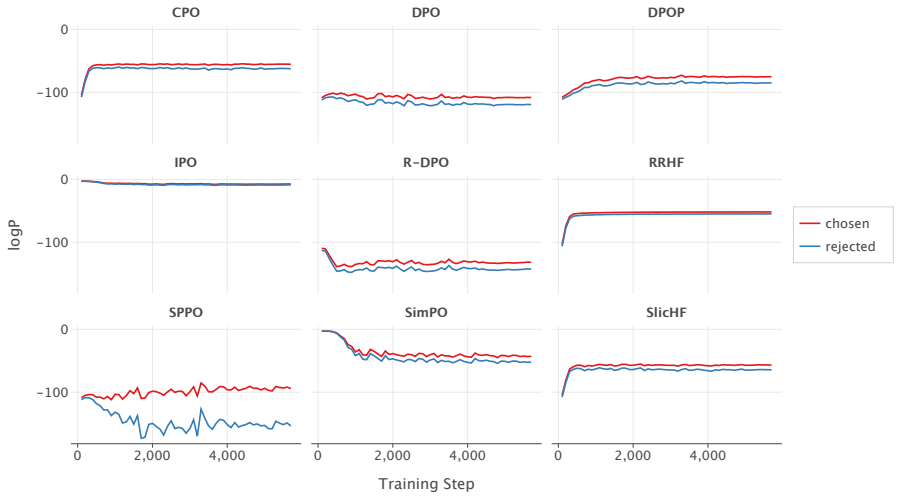


Figure 5: Training dynamics of the chosen and rejected log probabilities on the TL;DR dataset for different preference optimization algorithms trained on Llama-3 8B. All algorithms exhibit synchronized increases and decreases in the chosen and rejected log probabilities. Note: For SimPO and IPO, the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

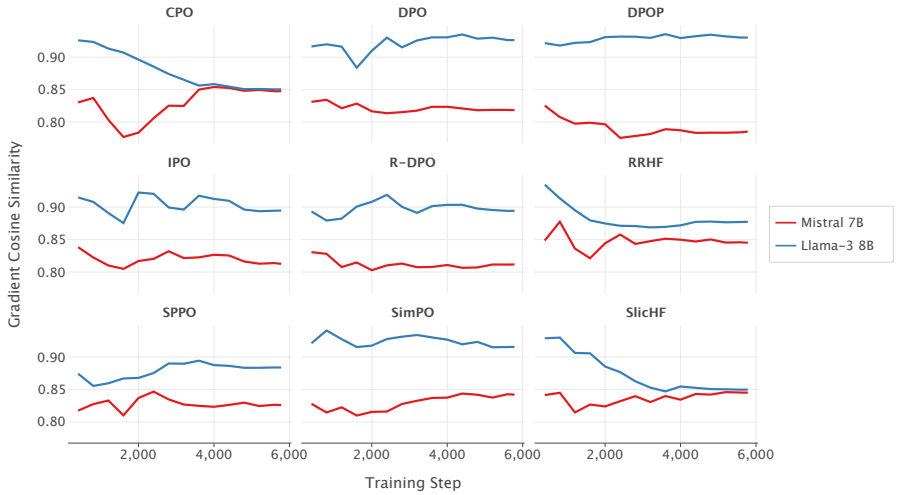
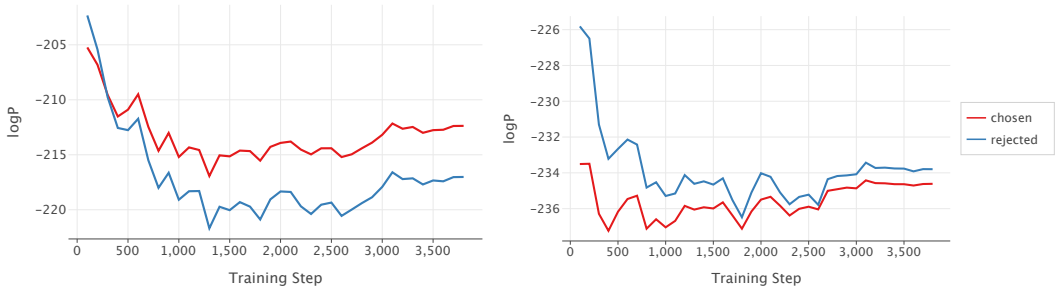


Figure 6: Cosine similarity between $\nabla_\theta \log \pi_w$ and $\nabla_\theta \log \pi_l$ on the TL;DR dataset for different preference optimization algorithms trained on Llama-3 8B and Mistral 7B.



(a) Mistral 7B

(b) Llama-3 8B

Figure 7: Training dynamics of the chosen and rejected log probabilities on the UltraFeedback dataset [3], with log probabilities reported on the evaluation set. As the margin between the two increases, the chosen and rejected log-probabilities exhibit synchronized increases and decreases per step. In Figure 7a and Figure 7b, both chosen and rejected log-probabilities have an overall trend of increasing, especially towards the end of training.