NORMALIZED REWARDS FOR PREFERENCE OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Direct Alignment Algorithms (DAAs) such as DPO have become a common way to post-train and align LLMs with human preferences. However, DAAs have been observed to over-optimize their implicit reward model and decrease the likelihood of preferred responses. We provide evidence for a hypothesis that the over-optimization stems in part from a mismatch in the partition function estimate of the learned model and the optimal model. In particular, transformers return a normalized distribution over tokens and therefore have a partition function of one, suggesting that the true partition function should remain fixed throughout training. However, existing DAAs do not account for this as their objectives do not include terms to optimize the partition function. To counteract this undesired side-effect of DAAs, we examine using objectives that add a regularization term to maintain the total length-normalized probabilities of the chosen and rejected responses. To better understand over optimization, we investigate how response likelihood changes are distributed over the tokens with and without regularization. We find that a significant portion of the likelihood changes are due to a small set of outlier tokens, which explains how DAAs improve generation quality despite decreasing the likelihoods of chosen responses. We apply the proposed regularization to reference-based (DPO) and reference-free (SimPO) methods and find (1) improved trade-offs between generation quality and general benchmark capability and (2) improvements in reward modeling across datasets. For example, on Llama-3.1-8B-Instruct, we see both a $> 20\%$ increase in AlpacaEval2 scores and $> 9\%$ performance gains on general benchmarks. Additionally, we find that the added regularization term effectively mitigates the amount of displacement within preferred responses overall, and for the outlier tokens specifically, by utilizing low-likelihood tokens.

1 INTRODUCTION

With the rise in interactions between large language models (LLMs) and humans, training LLMs to produce responses that are considered desirable by human users has become a vital step. Such training is commonly performed with methods that learn what it means for a response to be desirable from a dataset of paired preferred and non-preferred responses, such as Reinforcement Learning from Human Feedback (RLHF) Ouyang et al. (2022) and Direct Preference Optimization (DPO) Rafailov et al. (2023). The main difference between the two approaches is RLHF relies on a two-step training process while DPO uses only one. RLHF first learns a reward function that assigns more value to the preferred versus non-preferred response, and then uses the reward function to train an LLM policy. DPO directly updates the LLM by maximizing the likelihood of the preferred versus non-preferred responses. Due to the simplicity and reduced computational cost of DPO, there has been a rise in the use and development of Direct Alignment Algorithms (DAAs), which train directly on the preferred and non-preferred paired dataset.

Despite their popularity, recent work has identified a critical limitation of DAAs: they can over-optimize their implicit reward model and ultimately constrain improvements to the quality of the LLM's generations Razin et al. (2024); Huang et al. (2024). A common, concerning manifestation of over-optimization is likelihood displacement Razin et al. (2024), a phenomenon whereby the likelihoods of both the preferred and the non-preferred responses drop simultaneously, potentially resulting in harmful behavior. However, despite these documented issues, DAAs are nonetheless still successfully used in post-training Dubey et al. (2024); Groeneveld et al. (2024).

Several hypotheses have been proposed to explain the causes of over-optimization in DAAs: high embedding or textual similarity between preferred and non-preferred responses Pal et al. (2024); Tajwar et al. (2024); Razin et al. (2024), or the insufficient regularization provided by the shape of the implicit reward function Huang et al. (2024); Gupta et al. (2025). However, no single explanation satisfactorily generalizes across all DAA objectives. For example, a common difference between DAAs is the presence versus absence of a reference model in the reward computation (e.g., DPO uses a reference model for its reward, while SimPO Meng et al. (2024) is reference-free), and hypotheses that explain over optimization for a reference-free method do not hold or have not been applied for methods without a reference model and vice versa. This is evidenced by the fact that existing analyses on over-optimization Yoon et al. (2025); Huang et al. (2024); Razin et al. (2024) focus on single instances of DAA, either reference-based or reference-free, and not both. The limitations of the current hypotheses motivate our research question: *how can we explain and mitigate reward over-optimization for both reference-based and reference-free rewards*?

In this work, we propose that reward over-optimization—particularly likelihood displacement—stems from a lack of normalization of the implicit reward. To counteract this, we introduce a regularization term designed to conserve the total response probability within preferred and non-preferred response pairs. To test the validity of our regularization term, we evaluate its impact on both reference-based (DPO) and reference-free (SimPO) methods. We find that its inclusion leads to (1) improved trade-offs between generation quality and general capability benchmarks, and (2) comparable or better reward modeling across datasets. Furthermore, our analysis reveals new insights into the mechanics of likelihood displacement. We discover that this phenomenon is highly concentrated, with a small subset of outlier tokens accounting for the majority of likelihood shift. These findings further support the use of our regularization term, and shed more light on why DAAs, like DPO and SimPO, improve generation despite causing likelihood displacement.

We provide a summary of the key results below:

1. We provide evidence that miscalibrated rewards in DAAs can be attributed to a poor estimation of the partition functions.
2. We identify outlier tokens as a significant contributor to likelihood displacement and find our regularization mitigates their outsized impact on response likelihood by utilizing low-likelihood tokens.
3. We demonstrate that our modified objective improves trade-offs between generation quality and benchmark performance Lin et al. (2023) for both DPO and SimPO.

2 Background

In this section, we introduce how RLHF and DPO utilize preference data.

RLHF. The preference learning component of RLHF is composed of two stages. The first consists of training a reward model $r_\phi(x, y)$ parameterized by ϕ on pairwise comparisons to assign scores to generated responses y to a given prompt x. Given a pairwise preference $y_w \succ y_l$, the reward model $r_\phi(x, y)$ is trained to minimize the negative log-likelihood of the reward assignments under the Bradley-Terry model

$$-\log p(y_w \succ y_l) = -\log \sigma(r(x, y_w) - r(x, y_l)), \tag{1}$$

where σ is the logistic function. Using this reward model in the second stage, the language model π_θ is then trained to maximize the expected reward of its responses under a KL constraint—added to mitigate drifting too far from the original model. The objective can be written as

$$\mathbb{E}_{x\sim D, y\sim\pi_\theta(\cdot|x)}\big[r(x, y) - \beta\mathbb{KL}[\pi_\theta(\cdot|x)||\pi_{\text{ref}}(\cdot|x)]\big], \tag{2}$$

where π_θ is the current model, π_{ref} is the reference model, and β is a hyperparameter for the KL constraint.

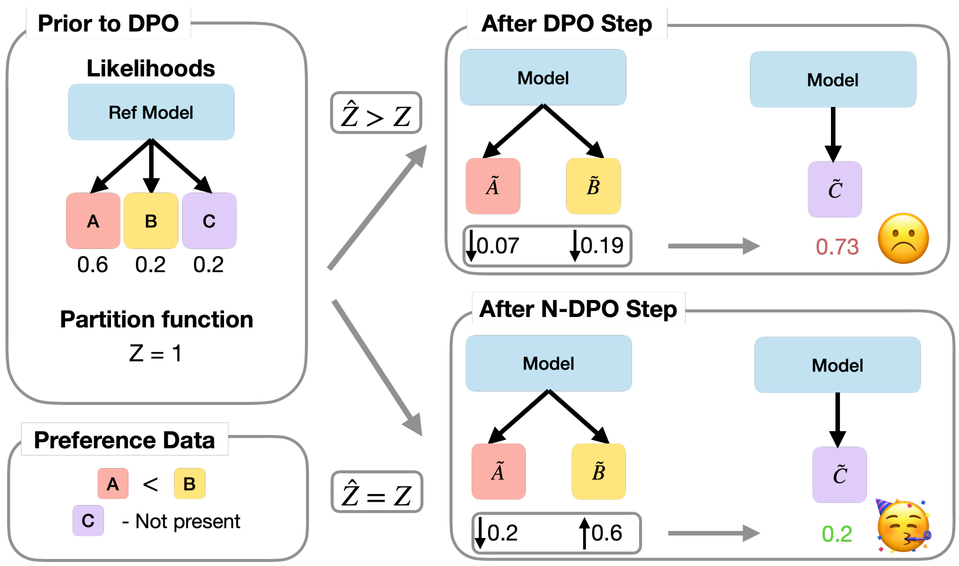

Figure 1: To address the lack of optimization of the partition function in the implicit rewards of DAAs which is necessary for reward normalization, we propose a regularization term to conserve total response probabilities. We show how despite learning the same set of rewards, if the partition function estimate is large, this results in a negative offset in likelihoods. When the partition function is 1, the likelihoods for the preferred response increases and the likelihood of responses outside the preference data is conserved.

DPO. DPO is a direct alignment algorithm derived from the RLHF objective which removes the need for an external reward model and directly updates the language model using preference data. This is done by utilizing the fact that the optimal policy π^* under the RLHF objective can be written as

$$\pi^*(y|x) = \frac{1}{Z(x)}\pi_{\text{ref}}(y|x)\exp\left(\frac{r(x,y)}{\beta}\right), \tag{3}$$

where $Z(x)$ is defined as

$$Z(x) = \sum_y \pi_{\text{ref}}(y|x)\exp\left(\frac{r(x,y)}{\beta}\right), \tag{4}$$

and denotes the partition function, which normalizes the output distribution of $\pi^*(y|x)$, ensuring the sum of response probabilities is 1. From this, we can write the reward $r(x,y)$ in terms of the optimal policy and the reference policy

$$r(x,y) = \beta\log\frac{\pi^*(y|x)}{\pi_{\text{ref}}(y|x)} + \beta\log Z(x). \tag{5}$$

Then, by defining

$$r_\theta(x,y) = \beta\log\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} + \beta\log Z(x), \tag{6}$$

which corresponds to the reward model under which π_θ is optimal. We note that the partition function acts as an offset that ensures the rewards are calibrated and only when the partition function is 1 do the rewards directly correspond to log-likelihood ratios. By maximizing the likelihood of the policy under the Bradley-Terry model, the reward model and policy are simultaneously optimized, and under mild conditions, should result in the same optimal policy as RLHF. Expanding the reward terms, we have that the DPO objective is

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

with the partition function term cancelling out and therefore missing due to using the difference of rewards.

SimPO. A variant of DPO introduced to further simplify alignment training is SimPO. This variant removes the need for a reference model by using only the current model's likelihood of a response as the reward, and considers length-normalized probabilities. We can write the SimPO reward $r_\theta(x, y)$ for prompt x and response y as

$$r_\theta(x, y) = \beta \bar{\pi}_\theta(y|x) + Z(x), \tag{8}$$

where $\bar{\pi}_\theta(y|x) = \pi_\theta(y|x)^{1/|y|}$ is the length-normalized likelihood of a response for some model π_θ with $|y|$ being the length of the response and $Z(x)$ is the partition function defined as

$$Z(x) = \sum_y \pi_{\text{ref}}(y|x) \exp\left(\frac{r_\theta(x, y)^{|y|}}{\beta}\right) \tag{9}$$

In addition, SimPO introduces the use of a margin term to further separate the likelihoods of the preferred and non-preferred responses. Finally, the SimPO objective can be written as follows:

$$\mathcal{L}_{\text{SimPO}} = \mathbb{E}_{(x,y_w,y_l)\sim D}\left[-\log\sigma\left(\beta\log\bar{\pi}_\theta(y_w|x) - \beta\log\bar{\pi}_\theta(y_l|x) - \beta\gamma\right)\right], \tag{10}$$

where γ is a hyperparameter for the margin size.

Dangers of neglecting the partition function Crucially, neither the DPO nor the SimPO objectives contain the partition function $Z(x)$. Indeed, since they consider the *differences* in rewards for each pair of responses, the $Z(x)$ terms cancel out in the respective objective formulations for DPO and SimPO, effectively rendering them invariant to changes in $Z(x)$. While mathematically convenient, this simplification deprives the model of an important factor — it does not incentivize the preservation of a good estimate of $Z(x)$. This blindness to $Z(x)$ provides a compelling explanation for likelihood displacement: as the response probabilities for a given prompt are offset together by changes in $Z(x)$, a poor estimate of $Z(x)$, in particular large estimates far from 1, result in a large negative shift in likelihood for both responses being necessary to offset the $\log Z(x)$ term. We provide an illustration of how a large estimate of $Z(x)$ can result in reduced likelihood for responses in Figure 1. We expect estimates of the partition function that are unoptimized and based only on two responses to be miscalibrated, also explaining the frequency of likelihood displacement. This points towards the need to optimize the partition function towards a better estimate.

3 METHOD

We first consider what the partition function for the rewards should be. If we consider any parameterized language model π_θ that applies softmax to its outputs, then the output distribution is always normalized and the partition function for the model is 1. As a result, we have the key property that *the partition function should be fixed throughout training*.

However, the implicit DPO reward does not account for this constraint as the objective does not contain a term for the partition function resulting in phenomena such as likelihood displacement Razin et al. (2024). Due to difficulties in effectively estimating the partition function for preference datasets (each prompt has only a single pair of responses), we propose a regularization term motivated by the insight of a fixed partition function. We enforce normalization in the rewards by adding a regularization term that maintains the probability mass over the set of responses seen for a prompt. Given a preference data point with prompt x and responses y_w, y_l, we start with a regularization penalty of

$$\lambda\left(\log\frac{\pi_\theta(y_w|x) + \pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_w|x) + \pi_{\text{ref}}(y_l|x)}\right)^2, \tag{11}$$

where λ is a hyperparameter. The penalty aims to keep the ratio of the total response probability close to 1. Notice that the regularization term is minimized when the ratio of the total response probabilities is 1 and the total probability assigned to the two responses under the optimized model is the same as that under the original model. By maintaining the total probability, we mitigate offsets in likelihood that would occur given a poor estimate of the partition function and as a result, implicitly improve the partition function estimate. However, response probabilities decrease exponentially with length and as responses are often hundreds of tokens long, the ratio of response probabilities may be sensitive to differences in length or small changes in per-token probabilities. To have a more stable penalty and to mitigate length bias, we use length-normalized probabilities. Using $\bar{\pi}_\theta, \bar{\pi}_{\text{ref}}$ to denote length-normalized probabilities, we have the following regularization penalty:

$$\mathcal{R}(\pi_\theta, \pi_{\text{ref}}, x, y_w, y_l) = \lambda \left(\log \frac{\bar{\pi}_\theta(y_w|x) + \bar{\pi}_\theta(y_l|x)}{\bar{\pi}_{\text{ref}}(y_w|x) + \bar{\pi}_{\text{ref}}(y_l|x)} \right)^2. \tag{12}$$

For consistency with the regularization, we modify the original objective with length-normalization which, with the regularization, gives the following objective for DPO:

$$\mathcal{L}_{\text{N-DPO}}(\theta) = \mathcal{L}_{\text{DPO}}(\pi_\theta) + \mathcal{R}(\pi_\theta, \pi_{\text{ref}}, x, y_w, y_l). \tag{13}$$

We refer to the modified version of DPO as N-DPO. We also modify SimPO with the same form of regularization, but since SimPO is already length normalized, we simply add the regularization term resulting in:

$$\mathcal{L}_{\text{N-SimPO}}(\theta) = \mathcal{L}_{\text{SimPO}}(\pi_\theta) + \mathcal{R}(\pi_\theta, \pi_{\text{ref}}, x, y_w, y_l), \tag{14}$$

which we refer to as N-SimPO.

4 Token-wise Analysis

In this section, we consider a finer-grained analysis of the reward distribution and likelihood displacement. We explore how the reward for each token changes by studying empirically the distribution of token-wise rewards with and without regularization, and a theoretical gradient analysis. Our analysis reveals how likelihood displacement is distributed across tokens and how our regularization term uses low-likelihood tokens to reshape the reward distribution and response likelihoods.

4.1 Token-wise Reward Analysis

To understand how the reward model changes, we analyze the token-wise rewards for each of the methods on UltraFeedback Cui et al. (2023). We do so by considering the overall token-wise reward distribution for each model and method as well as the distribution of the minimum token-wise reward per sample. To provide a clear comparison across settings, we use the change in log-likelihood per token as a normalized reward. We provide the results for Llama-3.1-8B-Instruct Dubey et al. (2024) in Figure 3. When using DPO the peak of the minimum reward distribution is around -10 while the overall reward distribution lies mostly within -2.5 and 2.5, suggesting that for many samples when using DPO, the minimum reward lies far outside the typical range. This suggests that a significant part of likelihood displacement comes from outlier tokens significantly dropping the response likelihood. We demonstrate that this change does in fact reduce likelihood displacement by plotting the likelihood of responses over training in Figure 2. We also observe that while the overall token distribution does not change much between DPO and N-DPO, we see that there is a large shift in the minimum reward distribution. This suggests that N-DPO primarily mitigates the effect of these outlier tokens while maintaining the reward otherwise. We demonstrate how our regularization term mitigates these outlier tokens when total likelihood decreases as seen with Llama-3.1-8B-Instruct.

4.2 Token-wise Gradient Analysis

We consider the gradient of

$$\mathcal{R}(\pi_\theta, \pi_{\text{ref}}, x, y_w, y_l) = \lambda \left(\log \frac{\bar{\pi}_\theta(y_w|x) + \bar{\pi}_\theta(y_l|x)}{\bar{\pi}_{\text{ref}}(y_w|x) + \bar{\pi}_{\text{ref}}(y_l|x)} \right)^2 \tag{15}$$

with respect to θ. First, we define the necessary notation. Let $P_\theta(x), P_{\text{ref}}(x)$ be the total length-normalized response probabilities under the trained and reference model respectively and let $\pi_\theta(y_{w/l}^{(i)})$ be the likelihood of the ith token in a response given the previous tokens. Then, we can write the gradient of the regularization term with respect to the parameters θ as

$$\nabla_\theta \mathcal{R} = \frac{2\lambda}{P_\theta(x)} \log\left(\frac{P_\theta(x)}{P_{\text{ref}}(x)}\right) \left(\frac{\bar{\pi}_\theta(y_w|x)}{|y_w|} \sum_{i=1}^{|y_w|} \frac{\nabla_\theta(\pi_\theta(y_w^{(i)}))}{\pi_\theta(y_w^{(i)})} + \frac{\bar{\pi}_\theta(y_l|x)}{|y_l|} \sum_{i=1}^{|y_l|} \frac{\nabla_\theta(\pi_\theta(y_l^{(i)}))}{\pi_\theta(y_l^{(i)})} \right). \tag{16}$$

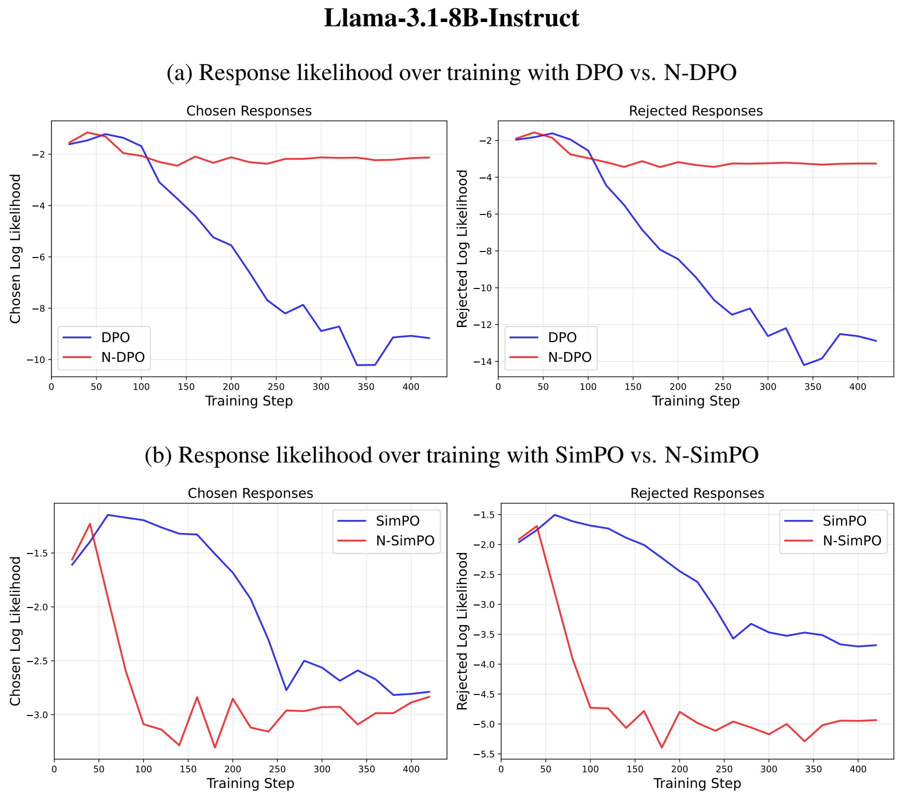

Figure 2: Comparison of response likelihoods between DPO/SimPO and N-DPO/N-SimPO for Llama-3.1-8B-Instruct. On the left is the chosen response likelihood and on the right is the rejected response likelihood.

Notice that all of the token-wise contributions to the gradient have the same sign. The sign of the gradients are determined by $\log\left(\frac{P_\theta(x)}{P_{\text{ref}}(x)}\right)$ where if the total response probability has decreased, the gradients will be negative increasing token probabilities. If the total probability increases, the opposite will occur. Furthermore, looking at each token-wise gradient, we have $\frac{\nabla_\theta(\pi_\theta(y^{(i)}))}{\pi_\theta(y^{(i)})}$ which is inversely proportional to the likelihood of each token. If a token has small likelihood (e.g., $1e-7$ smaller than other tokens) the low-likelihood token's gradient will dominate. This has been observed in the gradient analysis in the ConfPO paper Yoon et al. (2025). Then, if there is an outlier token with small likelihood and the response probability has decreased as seen with models such as Llama-3.1-8B-Instruct, the regularization term will strongly increase the likelihood of the outlier tokens. More generally, if the response likelihood has decreased significantly, the regularization term will prioritize updating the lowest likelihood tokens to increase the overall response likelihood. In this way, when the response likelihood decreases, the regularization term primarily shifts large negative rewards closer to 0.

We can also consider the case when the total response probability has increased. Now, the regularization term has gradients that will result in a decrease in response probability, but similar to before, these gradients will be dominated by the low likelihood tokens. Then, when total response probability has increased compared to the original, the regularization corrects this primarily by decreasing the likelihood of low likelihood tokens. In this way, the overall response probability is maintained with minimal changes to the most likely tokens, which are also most relevant for generation. In this way, the regularization term mitigates overly large likelihoods and does so with minimal distribution shift.

The gradient analysis reveals that introducing the regularization term effectively mitigates shifts in response likelihood and mitigates the presence of outlier tokens. Furthermore, we find that the regularization term does so primarily through low likelihood tokens which have a smaller effect on the overall sampling distribution. In this way, our regularization term demonstrates that low likelihood tokens not only provide an approximation of the gradient Yoon et al. (2025) but also can be utilized to shape the reward and likelihood distribution.

Llama-3.1-8B-Instruct

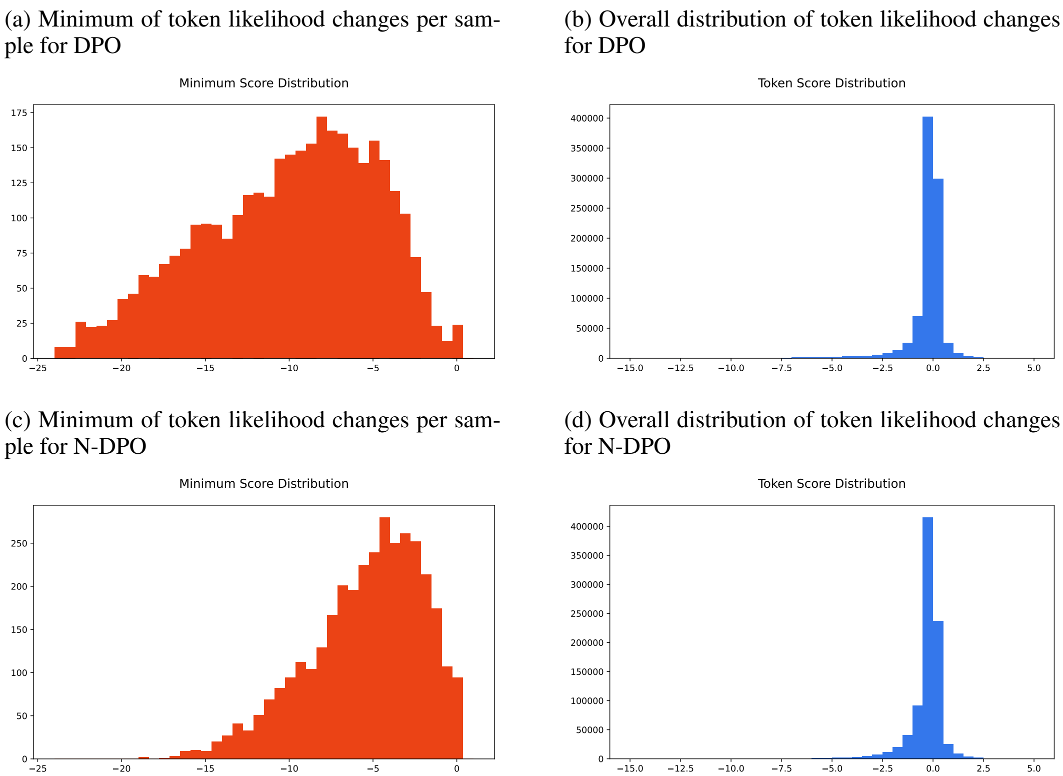

Figure 3: Comparison of token-wise reward distributions between DPO and N-DPO for Llama-3.1-8B-Instruct. On the left is the distribution of the minimum token reward per sample, and on the right is the distribution of all token rewards.

5 Experiments

We evaluate the impact of accounting for reward normalization on downstream performance on instruction-following tasks, common sense and reasoning, and implicit reward modeling.

5.1 Evaluation

We focus our evaluation on understanding the effect of our modifications by comparing DPO with N-DPO and SimPO with N-SimPO. We expect that due to the more strongly enforced normalization of rewards, likelihood displacement would be mitigated, and more generally, the model can learn from the preference data with less of a distribution shift. As a result, we expect to see better trade-offs between generation quality and benchmark performance when using N-DPO or N-SimPO. We also expect that with better normalized rewards, we may see better generalization of the implicit reward.

We train Mistral-7B-Instruct Jiang et al. (2023), Llama-3.1-8B-Instruct Dubey et al. (2024), and OLMo-7B-SFT Groeneveld et al. (2024) on Ultrafeedback Cui et al. (2023) for using DPO, N-DPO, SimPO, and N-SimPO. For all runs, we train for 1 epoch with a cosine learning rate scheduler. We evaluate generation quality using AlpacaEval Dubois et al. (2024) and assess the model on common sense and reasoning benchmarks (ARC, MMLU, HellaSwag, PIQA, SciQ, WinoGrande). We evaluate reward modeling for a range of datasets (Ultrafeedback, HH-RLHF Bai et al. (2022), HelpSteer2 Wang et al. (2024)).

Instruction Following. AlpacaEval Dubois et al. (2024) is a benchmark that evaluates a model based on its win-rate compared to GPT-3.5 OpenAI (2022) for AlpacaEval1 and GPT-4 Achiam et al. (2023) for AlpacaEval2 using an LLM-as-a-judge in an instruction-following setting. For AlpacaEval2 both a raw win-rate (WR) and a length-controlled win-rate (LC) are provided. Table 1 shows the win-rates of the models on AlpacaEval1 and AlpacaEval2, where we can see that N-DPO and N-SimPO generally have improved performance over DPO and SimPO, respectively. In particular, we see over a 20% increase for AlpacaEval2 (LC) for Llama-3.1-8B-Instruct between DPO and N-DPO and a large increase in both AlpacaEval1 and AlpacaEval2 (WR) between SimPO and N-SimPO for OLMo-7B-SFT with over a 75% increase for AlpacaEval2 (WR). While we see a small decrease in instruction-following quality using N-SimPO for Llama-3.1-8B-Instruct, the benchmark performance noticeably improves. We provide comparisons to adding an SFT term to the DPO loss in Appendix A.

Table 1: AlpacaEval scores across methods. A1 corresponds to the AlpacaEval1 win-rate, WR corresponds to the raw AlpacaEval2 win-rate, and LC corresponds to the length-controlled AlpacaEval2 win-rate. Standard deviations are provided in Appendix A.

	Mistral-7B-Instruct			**Llama-3.1-8B-Instruct**			**OLMo-7B-SFT**		
	A1	WR	LC	A1	WR	LC	A1	WR	LC
Reference	93.17	13.97	16.84	90.00	24.93	19.67	58.15	3.32	5.06
DPO	**94.66**	16.80	21.2	**91.67**	27.70	24.46	79.63	6.53	7.07
N-DPO	94.28	**18.16**	**22.10**	91.28	**31.84**	**29.41**	**81.24**	**9.21**	**8.02**
SimPO	90.87	17.61	23.28	**88.93**	**34.89**	**32.79**	71.21	4.57	6.32
N-SimPO	**92.72**	**18.87**	**23.67**	85.45	32.90	31.01	**79.63**	**8.07**	**7.17**

Common Sense and Reasoning. In addition to instruction-following quality, we perform evaluation on various benchmarks from LM Evaluation Harness Gao et al. (2024) to see how well the model maintains its general common sense and reasoning capabilities. We expect reducing likelihood displacement to also reduce distribution shift allowing for better benchmark performance. To quantify how well the model maintains benchmark performance, we consider the difference in scores between the reference model and the fine-tuned model. The results for the evaluation are shown in Table 2. We can see that for Llama-3.1-8B-Instruct, N-DPO and N-SimPO result not only in better maintenance of benchmark performance, but also improve scores on average by over 3%. For Mistral-7B-Instruct, we also see an improvement between DPO versus N-DPO along with a small drop in benchmark performance that is accompanied by an increase in generation quality by a larger margin than the drop in benchmark performance. For OLMo-7B-SFT, we find that N-DPO improves benchmark performance while improving generation quality. We note that for N-SimPO, there is a drop in benchmark performance, but the results suggest this is due to difficulties with applying SimPO to OLMo as generation quality increased the most with the smallest learning rate, margin, and beta. This suggests that larger updates using SimPO do not benefit OLMo-7B-SFT, and that N-SimPO can allow for more dramatic changes in weaker base models that improve generation quality. We provide the hyperparameter sweep range and final hyperparameters in Appendix B.

Implicit Reward. We further evaluate the trained models on their implicit reward modeling capabilities on both UltraFeedback and datasets not seen during training (HH-RLHF, HelpSteer2). We compute the reward accuracy on the eval splits for these datasets using a length-normalized reward. Table 3 shows the results.

5.2 Regularization Ablation

We perform an ablation on the regularization term by setting $\lambda = 0$ to demonstrate that maintaining the probability mass helps improve performance beyond using only length-normalization (L-DPO).

Table 2: Common sense and reasoning benchmarks performance across methods. Standard deviations are provided in Appendix A.

Mistral-7B-Instruct

	MMLU	ARC Chal	ARC Easy	HellaSwag	PiQA	SciQ	WinoG	Avg
Reference	44.32	49.49	70.33	62.86	75.19	90.90	61.48	64.94
DPO	-1.34	-3.41	-5.05	-1.45	-2.67	-3.00	-2.68	-2.80
N-DPO	+0.17	-1.11	-1.90	-0.95	-1.96	-0.80	-0.31	**-0.98**
SimPO	-0.70	-3.84	-3.12	-7.47	-5.06	-0.50	-2.21	**-3.27**
N-SimPO	-0.84	-4.10	-3.62	-6.87	-4.79	-1.20	-2.44	-3.41

Llama-3.1-8B-Instruct

	MMLU	ARC Chal	ARC Easy	HellaSwag	PiQA	SciQ	WinoG	Avg
Reference	43.28	52.82	81.44	57.52	79.76	95.20	67.40	68.20
DPO	+4.71	-4.78	-9.51	+6.43	-5.82	-4.20	-9.55	-3.25
N-DPO	+4.24	+3.84	+0.80	+6.24	+0.82	+0.20	+5.13	**+3.04**
SimPO	+1.69	+4.77	+0.17	-5.49	-0.54	+0.80	+6.55	+1.14
N-SimPO	+8.49	+9.21	+2.27	-0.92	+1.36	+1.10	+4.33	**+3.69**

OLMo-7B-SFT

	MMLU	ARC Chal	ARC Easy	HellaSwag	PiQA	SciQ	WinoG	Avg
Reference	37.94	39.33	67.97	53.54	76.66	91.1	63.93	61.50
DPO	-0.15	-0.85	-2.06	+0.27	-1.42	-4.80	-2.05	-1.58
N-DPO	+0.47	+1.88	+2.11	+3.52	-0.82	-1.20	-0.95	**+0.72**
SimPO	-0.04	-0.34	-0.25	-1.17	-1.09	-0.70	-1.50	**-0.73**
N-SimPO	+0.10	-1.53	-14.01	-3.49	-4.79	-4.20	-5.76	-4.81

Table 3: Reward accuracy on UltraFeedback (UF), HH-RLHF (HH), and HelpSteer2 (HS) across methods.

	Mistral-7B-Instruct			**Llama-3.1-8B-Instruct**			**OLMo-7B-SFT**		
	UF	HH	HS	UF	HH	HS	UF	HH	HS
DPO	80.73	55.12	66.15	74.71	**59.55**	68.23	71.12	53.30	64.06
N-DPO	94.28	**81.94**	**56.33**	**77.20**	58.73	**75.00**	**73.26**	**54.63**	**64.32**
SimPO	**82.12**	**57.17**	**64.32**	**76.22**	57.03	67.19	60.19	55.47	53.65
N-SimPO	82.06	56.82	63.28	75.81	**57.62**	**67.97**	**65.05**	**55.68**	**56.77**

We provide the results on AlpacaEval in Table 4 where we can see that performance improves when λ is a non-zero value.

6 Related Works

DAA Methods. A wide range of works have explored various approaches to improve existing DAA methods like DPO Rafailov et al. (2023). One class of changes to DPO consider modifying the function of the reward margin Azar et al. (2024); Zhao et al. (2023); Tang et al. (2024). A range of works have approached the problem of over-optimization and likelihood displacement through modifying the reward function based on alternatives to KL regularization Huang et al. (2024); Gupta et al. (2025); Wang et al. (2023), focusing on select tokens Yoon et al. (2025), mitigating length-normalization exploits Gupta et al. (b), or adding regularization Liu et al. (2024). Another line of work has proposed modifications to the DPO objective for robustness such as rDPO Chowdhury

Table 4: AlpacaEval scores for length-normalized DPO with and without regularization. WR corresponds to the raw win-rate and LC corresponds to the length-controlled win-rate. Standard deviations are provided in Appendix A.

Mistral-7B-Instruct

	AlpacaEval1	AlpacaEval2 (WR)	AlpacaEval2 (LC)
$\lambda = 0$	93.03	16.54	20.21
N-DPO	94.28	18.16	22.10

Llama-3.1-8B-Instruct

	AlpacaEval1	AlpacaEval2 (WR)	AlpacaEval2 (LC)
$\lambda = 0$	91.25	30.56	28.50
N-DPO	91.28	31.84	29.41

OLMo-7B-SFT

	AlpacaEval1	AlpacaEval2 (WR)	AlpacaEval2 (LC)
$\lambda = 0$	78.43	8.76	7.34
N-DPO	81.24	9.21	8.02

et al. (2024) and ROPO Liang et al. (2024), and other works have explored weighting samples Zhou et al. (2024) or using rejection sampling Xiong et al. (2023); Zhao et al. (2024); Liu et al. (2023). A range of works have considered various forms of data for aligning language models using reward data Chen et al. (2024a) or data from self play Wu et al. (2024); Gupta et al. (a); Tang et al. (2025). Other objectives include KTO Ethayarajh et al. (2024), which uses prospect theory to motivate an objective which does use pairwise comparisons but rather considers a set of desirable responses and undesirable responses, or ORPO Hong et al. (2024), which utilizes the log odds ratio between the preferred and dispreferred response for optimization.

Reward over-optimization. Outside of proposing new DAAs to mitigate over-optimization, works have also analyzed factors that may lead to over-optimization and likelihood displacement. Razin et al. (2024) studies under an simplified model how the embedding geometry may lead to likelihood displacement. Pal et al. (2024) provides an analysis of how likelihood displacement can arise given preference pairs with small edit distance. Reward over-optimization is also a generally observed phenomenon outside of DAAs with RLHF Chen et al. (2024b) and other reinforcement learning settings Skalse et al. (2022); Ibarz et al. (2018).

7 Discussion

Our results suggest that a common factor in reward over-optimization and likelihood displacement across methods and, in particular, across both reference-based and reference-free methods is the lack of reward normalization. The objectives modified to normalize rewards, N-DPO and N-SimPO, demonstrate better trade-offs between generation quality and benchmark performance, sometimes improving both while also maintaining reward modeling abilities. The improvement across various axes suggests that reward normalization has a significant role in DAAs and that enforcing such constraints can be an effective addition to methods. Furthermore, our analysis of token-wise rewards demonstrates that likelihood displacement does not affect the model broadly, but rather primarily on a limited number of tokens, explaining why generation improves despite decreasing likelihood. A token-wise analysis of the gradient of our proposed regularization term demonstrates that our regularization term effectively mitigates such outlier tokens and more generally utilizes low likelihood tokens to reshape the reward and likelihood distribution. These analyses provide insight into the role of different tokens in preference optimization and demonstrate the need for finer-grained analyses of reward model behavior.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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.

Huayu Chen, Guande He, Lifan Yuan, Ganqu Cui, Hang Su, and Jun Zhu. Noise contrastive alignment of language models with explicit rewards. *Advances in Neural Information Processing Systems*, 37:117784–117812, 2024a.

Lichang Chen, Chen Zhu, Davit Soselia, Jiuhai Chen, Tianyi Zhou, Tom Goldstein, Heng Huang, Mohammad Shoeybi, and Bryan Catanzaro. Odin: Disentangled reward mitigates hacking in rlhf. *arXiv preprint arXiv:2402.07319*, 2024b.

Sayak Ray Chowdhury, Anush Kini, and Nagarajan Natarajan. Provably robust dpo: Aligning language models with noisy feedback. *arXiv preprint arXiv:2403.00409*, 2024.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. Ultrafeedback: Boosting language models with high-quality feedback. 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 e-prints*, pp. arXiv–2407, 2024.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL `https://zenodo.org/records/12608602`.

Dirk Groeneveld, Iz Beltagy, Pete Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Harsh Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang, et al. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*, 2024.

Aman Gupta, Shao Tang, Qingquan Song, Sirou Zhu, Jiwoo Hong, Ankan Saha, Viral Gupta, Noah Lee, Eunki Kim, Siyu Zhu, et al. Alphapo: Reward shape matters for llm alignment. *arXiv preprint arXiv:2501.03884*, 2025.

Taneesh Gupta, Rahul Madhavan, Xuchao Zhang, Chetan Bansal, and Saravan Rajmohan. Ampo: Active multi preference optimization for self-play preference selection. In *Forty-second International Conference on Machine Learning*, a.

Taneesh Gupta, Rahul Madhavan, Xuchao Zhang, Chetan Bansal, and Saravan Rajmohan. Refa: Reference free alignment with fine-grained length control. In *Second Conference on Language Modeling*, b.

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

Audrey Huang, Wenhao Zhan, Tengyang Xie, Jason D Lee, Wen Sun, Akshay Krishnamurthy, and Dylan J Foster. Correcting the mythos of kl-regularization: Direct alignment without overoptimization via chi-squared preference optimization. *arXiv preprint arXiv:2407.13399*, 2024.

Borja Ibarz, Jan Leike, Tobias Pohlen, Geoffrey Irving, Shane Legg, and Dario Amodei. Reward learning from human preferences and demonstrations in atari. *Advances in neural information processing systems*, 31, 2018.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

Xize Liang, Chao Chen, Shuang Qiu, Jie Wang, Yue Wu, Zhihang Fu, Zhihao Shi, Feng Wu, and Jieping Ye. Ropo: Robust preference optimization for large language models. *arXiv preprint arXiv:2404.04102*, 2024.

Yong Lin, Hangyu Lin, Wei Xiong, Shizhe Diao, Jianmeng Liu, Jipeng Zhang, Rui Pan, Haoxiang Wang, Wenbin Hu, Hanning Zhang, et al. Mitigating the alignment tax of rlhf. *arXiv preprint arXiv:2309.06256*, 2023.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*, 2023.

Zhihan Liu, Miao Lu, Shenao Zhang, Boyi Liu, Hongyi Guo, Yingxiang Yang, Jose Blanchet, and Zhaoran Wang. Provably mitigating overoptimization in rlhf: Your sft loss is implicitly an adversarial regularizer. *Advances in Neural Information Processing Systems*, 37:138663–138697, 2024.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

OpenAI. Openai gpt-3.5 [text-davinci-003], 2022.

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.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Noam Razin, Sadhika Malladi, Adithya Bhaskar, Danqi Chen, Sanjeev Arora, and Boris Hanin. Unintentional unalignment: Likelihood displacement in direct preference optimization. *arXiv preprint arXiv:2410.08847*, 2024.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471, 2022.

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

Xiaohang Tang, Sangwoong Yoon, Seongho Son, Huizhuo Yuan, Quanquan Gu, and Ilija Bogunovic. Game-theoretic regularized self-play alignment of large language models. *arXiv preprint arXiv:2503.00030*, 2025.

Yunhao Tang, Zhaohan Daniel Guo, Zeyu Zheng, Daniele Calandriello, Rémi Munos, Mark Rowland, Pierre Harvey Richemond, Michal Valko, Bernardo Ávila Pires, and Bilal Piot. Generalized preference optimization: A unified approach to offline alignment. *arXiv preprint arXiv:2402.05749*, 2024.

Chaoqi Wang, Yibo Jiang, Chenghao Yang, Han Liu, and Yuxin Chen. Beyond reverse kl: Generalizing direct preference optimization with diverse divergence constraints. *arXiv preprint arXiv:2309.16240*, 2023.

Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, Gerald Shen, Jiaqi Zeng, Oleksii Kuchaiev, and Yi Dong. Helpsteer2-preference: Complementing ratings with preferences, 2024. URL https://arxiv.org/abs/2410.01257.

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.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint. *arXiv preprint arXiv:2312.11456*, 2023.

Hee Suk Yoon, Eunseop Yoon, Mark A Hasegawa-Johnson, Sungwoong Kim, and Chang D Yoo. Confpo: Exploiting policy model confidence for critical token selection in large language model preference optimization. *arXiv preprint arXiv:2506.08712*, 2025.

Hanyang Zhao, Genta Indra Winata, Anirban Das, Shi-Xiong Zhang, David D Yao, Wenpin Tang, and Sambit Sahu. Rainbowpo: A unified framework for combining improvements in preference optimization. *arXiv preprint arXiv:2410.04203*, 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.

Wenxuan Zhou, Ravi Agrawal, Shujian Zhang, Sathish Reddy Indurthi, Sanqiang Zhao, Kaiqiang Song, Silei Xu, and Chenguang Zhu. Wpo: Enhancing rlhf with weighted preference optimization. *arXiv preprint arXiv:2406.11827*, 2024.

A ADDITIONAL RESULTS

We provide AlpacaEval scores for the DPO+SFT method in Table 5.

Table 5: AlpacaEval scores across methods. A1 corresponds to the AlpacaEval1 win-rate.

	Llama-3.1-8B-Instruct
	A1
Reference	93.17
DPO	**94.66**
N-DPO	94.28
DPO+SFT	93.52

We provide the standard errors for AlpacaEval scores in Table 6, for AlpacaEval scores from the ablation in Table 8, and for common sense and reasoning benchmarks in Table 7.

Mistral-7B-Instruct

	AlpacaEval1	AlpacaEval2 (WR)	AlpacaEval2 (LC)
Reference	0.890	1.082	0.745
DPO	0.793	1.157	0.814
N-DPO	0.820	1.188	0.832
SimPO	0.999	1.163	0.787
N-SimPO	0.910	1.192	0.814

Llama-3.1-8B-Instruct

	AlpacaEval1	AlpacaEval2 (WR)	AlpacaEval2 (LC)
Reference	1.061	1.293	0.630
DPO	0.974	1.335	0.703
N-DPO	0.996	1.403	0.712
SimPO	1.107	1.419	0.546
N-SimPO	1.244	1.403	0.660

OLMo-7B-SFT

	AlpacaEval1	AlpacaEval2 (WR)	AlpacaEval2 (LC)
Reference	1.733	0.558	0.325
DPO	1.415	0.762	0.416
N-DPO	1.371	0.871	0.436
SimPO	1.592	0.633	0.368
N-SimPO	1.418	0.838	0.383

Table 6: AlpacaEval standard error across methods. WR corresponds to the raw win-rate and LC corresponds to the length-controlled win-rate.

Mistral-7B-Instruct

	ARC Challenge	MMLU	HellaSwag	ARC Easy	PiQA	SciQ	WinoGrande
Reference	1.461	0.409	0.482	0.937	1.008	0.910	1.368
DPO	1.457	0.408	0.486	0.977	1.042	1.032	1.383
N-DPO	1.460	0.409	0.485	0.954	1.033	0.945	1.370
SimPO	1.456	0.409	0.496	0.963	1.068	0.932	1.381
N-SimPO	1.455	0.408	0.495	0.967	1.065	0.962	1.382

Llama-3.1-8B-Instruct

	ARC Challenge	MMLU	HellaSwag	ARC Easy	PiQA	SciQ	WinoGrande
Reference	1.459	0.406	0.493	0.798	0.937	0.676	1.317
DPO	1.460	0.409	0.479	0.922	1.024	0.905	1.388
N-DPO	1.448	0.409	0.480	0.784	0.923	0.663	1.254
SimPO	1.444	0.408	0.499	0.795	0.818	0.620	1.233
N-SimPO	1.418	0.408	0.495	0.758	0.913	0.597	1.265

OLMo-7B-SFT

	ARC Challenge	MMLU	HellaSwag	ARC Easy	PiQA	SciQ	WinoGrande
Reference	1.428	0.400	0.498	0.957	0.987	0.901	1.350
DPO	1.422	0.400	0.498	0.973	1.007	1.088	1.365
N-DPO	1.438	0.401	0.494	0.940	0.999	0.953	1.357
SimPO	1.425	0.400	0.498	0.959	1.022	0.932	1.361
N-SimPO	1.417	0.401	0.499	0.982	1.049	1.067	1.386

Table 7: Common sense and reasoning benchmarks performance standard error across methods.

Mistral-7B-Instruct

	AlpacaEval1	AlpacaEval2 (WR)	AlpacaEval2 (LC)
L-DPO	0.899	1.138	0.793

Llama-3.1-8B-Instruct

	AlpacaEval1	AlpacaEval2 (WR)	AlpacaEval2 (LC)
L-DPO	1.000	1.381	0.692

OLMo-7B-SFT

	AlpacaEval1	AlpacaEval2 (WR)	AlpacaEval2 (LC)
L-DPO	1.445	0.865	0.365

Table 8: AlpacaEval standard error for length-normalized DPO without regularization. WR corresponds to the raw win-rate and LC corresponds to the length-controlled win-rate.

B HYPERPARAMETERS

We provide the set of hyperparameters used to perform hyperparameter sweeps for each method and model.

	Mistral-7B-Instruct	Llama-3.1-8B-Instruct	OLMo-7B-SFT
DPO	[0.03, 0.1, 0.3]	[0.01, 0.03, 0.1]	[0.03, 0.1, 0.3]
N-DPO	[0.3, 1.0, 3.0]	[0.3, 1.0, 3.0]	[0.3, 1.0, 3.0]
SimPO	[0.3, 1.0, 3.0]	[0.3, 1.0, 3.0]	[0.3, 1.0, 3.0]
N-SimPO	[3.0]	[3.0]	[0.3]

Table 9: Set of β used for hyperparameter sweep for each model/method

	Mistral-7B-Instruct	Llama-3.1-8B-Instruct	OLMo-7B-SFT
SimPO	[0.4, 0.8, 1.2, 1.6, 2.0]	[0.4, 0.8, 1.2, 1.6, 2.0]	[0.4, 0.8, 1.2, 1.6, 2.0]
N-SimPO	[0.8, 1.2, 1.6, 2.0]	[0.8, 1.2, 1.6, 2.0]	[0.8, 1.2, 1.6, 2.0]

Table 10: Set of γ used for hyperparameter sweep for each model/method

	Mistral-7B-Instruct	Llama-3.1-8B-Instruct	OLMo-7B-SFT
N-DPO	[0.0, 0.025, 0.05, 0.075, 0.1]	[0.0, 0.025, 0.05, 0.075, 0.1]	[0.0, 0.025, 0.05, 0.075, 0.1]
N-SimPO	[0.025, 0.05, 0.075, 0.1]	[0.025, 0.05, 0.075, 0.1]	[0.025, 0.05, 0.075, 0.1]

Table 11: Set of λ used for hyperparameter sweep for each model/method

Mistral-7B-Instruct	Llama-3.1-8B-Instruct	OLMo-7B-SFT
$[3e-8, 1e-7, 3e-7]$	$[1e-7, 3e-7, 1e-6]$	$[1e-7, 3e-7, 1e-6]$

Table 12: Set of learning rates used for hyperparameter sweep for each model

	Mistral-7B-Instruct	Llama-3.1-8B-Instruct	OLMo-7B-SFT
β	(0.03/1.0/3.0/3.0)	(0.01/3.0/3.0/3.0)	(0.03/3.0/0.3/0.3)
γ	(0/0/1.6/2.0)	(0/0/0.8/2.0)	(0/0/0.4/1.6)
λ	(0/0.1/0/0.1)	(0/0.025/0/0.025)	(0/0.05/0/0.025)
Learning Rate	(1e-7/3e-7/1e-7/1e-7)	(3e-7/1e-6/3e-7/1e-6)	(3e-7/1e-6/1e-7/3e-7)

Table 13: Hyperparameters used for model evaluation. (DPO/N-DPO/SimPO/N-SimPO)

C Gradient Derivation

We provide a derivation of the gradient of the regularization term

$$\mathcal{R}(\pi_\theta, \pi_{\text{ref}}, x, y_w, y_l) = \lambda \left(\log \frac{\bar{\pi}_\theta(y_w|x) + \bar{\pi}_\theta(y_l|x)}{\bar{\pi}_{\text{ref}}(y_w|x) + \bar{\pi}_{\text{ref}}(y_l|x)} \right)^2 \tag{17}$$

with respect to parameters θ. Letting $g(\pi_\theta) = \left(\log \frac{\bar{\pi}_\theta(y_w|x) + \bar{\pi}_\theta(y_l|x)}{\bar{\pi}_{\text{ref}}(y_w|x) + \bar{\pi}_{\text{ref}}(y_l|x)} \right)$, we have that

$$\nabla_\theta \mathcal{R} = 2\lambda g(\pi_\theta) \nabla_\theta g(\pi_\theta) \tag{18}$$

Now, defining $h(\pi_\theta) = \frac{\bar{\pi}_\theta(y_w|x) + \bar{\pi}_\theta(y_l|x)}{\bar{\pi}_{\text{ref}}(y_w|x) + \bar{\pi}_{\text{ref}}(y_l|x)}$ and denoting the denominator of $h(x)$ as $P_{\text{ref}}(x)$, we have

$$\nabla_\theta g(\pi_\theta) = \frac{1}{h(\pi_\theta)} \nabla_\theta h(\pi_\theta) = \frac{P_{\text{ref}}(x)}{\bar{\pi}_\theta(y_w|x) + \bar{\pi}_\theta(y_l|x)} \frac{1}{P_{\text{ref}}(x)} \nabla_\theta (\bar{\pi}_\theta(y_w|x) + \bar{\pi}_\theta(y_l|x)) \tag{19}$$

Writing $\bar{\pi}_\theta(y|x)$ as $\pi_\theta(y|x)^{1/|y|}$ and decomposing $\pi_\theta(y|x)^{1/|y|}$ as

$$\exp\left(\frac{1}{|y|} \sum_{i=1}^{|y|} \log \pi_\theta(y^{(i)}) \right) \tag{20}$$

we have

$$\nabla_\theta (\bar{\pi}_\theta(y_w|x) + \bar{\pi}_\theta(y_l|x)) = \left(\frac{\bar{\pi}_\theta(y_w|x)}{|y_w|} \sum_{i=1}^{|y_w|} \frac{\nabla_\theta \pi_\theta(y_w^{(i)})}{\pi_\theta(y_w^{(i)})} + \frac{\bar{\pi}_\theta(y_l|x)}{|y_l|} \sum_{i=1}^{|y_l|} \frac{\nabla_\theta \pi_\theta(y_l^{(i)})}{\pi_\theta(y_l^{(i)})} \right) \tag{21}$$

Through the chain rule and letting $P_\theta(x) = \bar{\pi}_\theta(y_w|x) + \bar{\pi}_\theta(y_l|x)$, we have that

$$\nabla_\theta \mathcal{R} = \frac{2\lambda}{P_\theta(x)} \log\left(\frac{P_\theta(x)}{P_{\text{ref}}(x)}\right) \left(\frac{\bar{\pi}_\theta(y_w|x)}{|y_w|} \sum_{i=1}^{|y_w|} \frac{\nabla_\theta(\pi_\theta(y_w^{(i)}))}{\pi_\theta(y_w^{(i)})} + \frac{\bar{\pi}_\theta(y_l|x)}{|y_l|} \sum_{i=1}^{|y_l|} \frac{\nabla_\theta(\pi_\theta(y_l^{(i)}))}{\pi_\theta(y_l^{(i)})} \right). \tag{22}$$