Enhancing Language Model Alignment: A Confidence-Based Approach to Label Smoothing

Baihe Huang[⋆], **Hiteshi Sharma**[†], **Yi Mao**[†]
[⋆]University of California Berkeley, [†]Microsoft Research
baihe_huang@berkeley.edu, {hiteshi.sharma, maoyi}@microsoft.com

Abstract

In recent years, Large Language Models (LLMs) have demonstrated remarkable capabilities across various domains. Within the training pipeline of LLMs, the Reinforcement Learning with Human Feedback (RLHF) phase is crucial for aligning LLMs with human preferences and values. Label smoothing, a technique that replaces hard labels with soft labels, emerges as promising techniques to enhance RLHF training. Despite the benefits, the choice of label smoothing parameters often relies on heuristic approaches and lack theoretical understanding. This paper addresses the challenge of selecting the label smoothing parameter in a principled manner. We introduce Confidence Aware Label Smoothing (CALS), a method that iteratively updates the label smoothing parameter based on preference labels and model forecasts. Our theoretical analysis characterizes the optimal label smoothing parameter, demonstrates its dependence on the confidence level, and reveals its influence on training dynamics and equilibrium. Empirical evaluations on state-of-the-art alignment tasks show that CALS achieves competitive performance, highlighting its potential for improving alignment.

1 Introduction

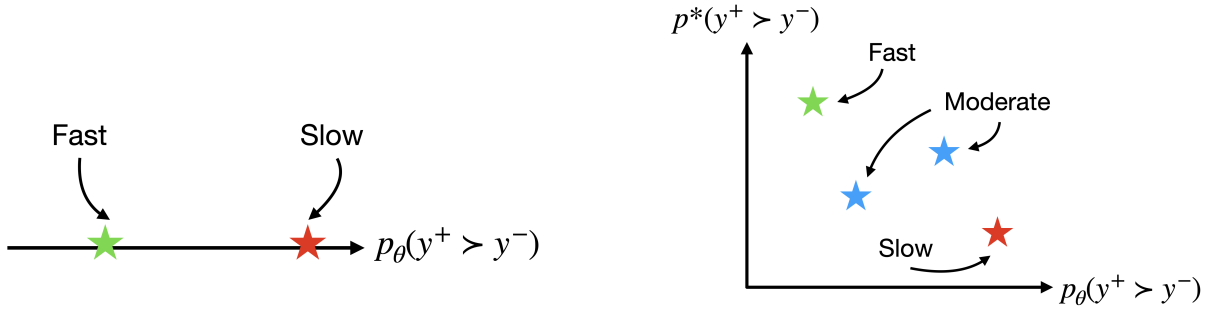
Large Language Models (LLMs), empowered by deep transformer architecture (Vaswani et al., 2017) and huge text datasets, have acquired remarkable capabilities in various domains (Brown et al., 2020; Bubeck et al., 2023; Chowdhery et al., 2023), achieving tremendous success on diverse tasks. The full training pipeline of LLMs can be divided into three stages: Pretraining phase (Radford et al., 2018) where the LLM is trained with unsupervised learning on huge corpus of text data, Supervised Fine-Tuning (SFT) phase (Wei et al., 2021) where the LLM is fine-tuned with supervised learning on datasets for the downstream task, and Reinforcement Learning with Human Feedback (RLHF) phase (Stiennon et al., 2020; Ouyang et al., 2022) where the LLM is trained on pairwise (or listwise) comparison dataset to improve the alignment with human preferences. In particular, the RLHF phase plays a critical role in controlling the model behavior and aligning LLMs with human values (Perez et al., 2022; Ganguli et al., 2022; Bai et al., 2022).

Proximal Policy Optimization (PPO) (Schulman et al., 2017) is widely adopted as the RL optimizer in RLHF phase. Recently, Direct Preference Optimization (DPO) (Rafailov et al., 2023) emerges as a promising alternative to Reinforcement Learning methods, as it directly optimizes the model with a classification loss and bypasses the need of fitting a separate reward model. In DPO and more generally MLE training, label smoothing (Müller et al., 2019; Mitchell, 2023) is a standard technique to mitigate the noise in preference labels and overfitting issues. However, the choice of the label smoothing parameter is often set heuristically and theoretical understanding remains scarce. Therefore in this work, we ask the question:

How to select the label smoothing parameter in a more principled way?

To answer this question, we first analyze the trade-off between bias and variance of the gradient estimate and then characterize the optimal choice of the label smoothing parameter that minimize the expected error of gradient estimate under different distance measures. Our theorem suggests that the label smoothing parameter should depend on the confidence level of the preference label. Motivated by this result, we propose Confidence Aware Label Smoothing (CALS), a method that iteratively updates the label smoothing parameter to reflect the confidence of preference label. Our method ensures that the gradient is weighted by both the confidence level of preference label and the correctness of model forecast based on preference label and model forecasts, illustrated in Figure 1.

Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 21341–21352
November 12-16, 2024



(a) DPO with no/constant label smoothing.

(b) DPO with confidence-aware label smoothing.

Figure 1: Illustration of our method, where $p_\theta(y^+ \succ y^-)$ represents how correctly the model is able to predict human preference and $p^*(y^+ \succ y^-)$ corresponds to how confident the preference label is. In vanilla DPO gradient, the likelihood update is weighted by $p_\theta(y^+ \succ y^-)$ in the sense that samples with lower $p_\theta(y^+ \succ y^-)$ (where the model is incorrect) get faster update. This is illustrated in Figure 1a. In confidence-aware label smoothing (Figure 1b), we add another dimension to the diagram, so that the update rate is also dependent on confidence of preference labels. In effect, samples with lower $p_\theta(y^+ \succ y^-)$ (where the model is incorrect) and higher $p^*(y^+ \succ y^-)$ (where the label is confident) will obtain faster update.

We analyze the equilibrium and dynamics of DPO with CALS, showing that our proposed method has a more stable gradient and conservative equilibrium. Finally, we empirically evaluate our method in state-of-the-art alignment tasks, which shows competitive performance over the baseline.

1.1 Related works

RLHF. Ziegler et al. (2019); Stiennon et al. (2020); Ouyang et al. (2022); Bai et al. (2022) establish RLHF pipelines for LLM alignment. With the framework, various methods refine the PPO-based RLHF algorithms, by improving the reward function training (Zhu et al., 2023a; Song et al., 2024; Zhu et al., 2024), reducing variance in RL optimization (Wu et al., 2023; Zhu et al., 2023b), or including sampling techniques (Dong et al., 2023; Gulcehre et al., 2023; Wang et al., 2024). Recently, alternative methods have been proposed to directly align human preference, including DPO (Rafailov et al., 2023), RRHF (Yuan et al., 2023), IPO (Azar et al., 2024), DRPO (Chang et al., 2024), Slic (Zhao et al., 2023), RSO (Liu et al., 2023), GPO (Tang et al., 2024),KTO (Ethayarajh et al., 2024), LiPO (Liu et al., 2024), ΨPO (Azar et al., 2024), and GRPO (Ramesh et al., 2024). These methods focuses on the form of loss functions, and are parallel to our work.

Label smoothing. Label smoothing has been used successfully in improving deep learning models in various tasks (Szegedy et al., 2016; Chorowski and Jaitly, 2016; Zoph et al., 2018). In knowledge distillation, using soft labels help the training of student networks (Hinton et al., 2015). Various methods propose adaptive or instance-dependent label smoothing techniques (Krothapalli and Abbott, 2020; Maher and Kull, 2021; Liu et al., 2022; Park et al., 2023; Guo et al., 2024) to strengthen the method. It has been shown that appropriate label smoothing can improve generalization (Müller et al., 2019), mitigate label noises (Liang et al., 2024), and prevent reward overfitting (Zhu et al., 2024).

2 Preference Learning

RLHF pipelines. Reinforcement Learning with Human Feedback (Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022) is a process to align with human preference, typically happening after training after pretraining and Supervised Finetuning (SFT). In this phase, a set of preference data (x, y^+, y^-) is given, where x is the prompt and y^+ is the preferred response over y^-.

First, RLHF trains a reward function $r_\phi(x, y)$ that maps a pair of prompt and response to a scalar value. In Bradley-Terry model (Bradley and Terry, 1952), the probability of preferring response y_1 over y_2 given prompt x is expressed by

$$p(y_1 \succ y_2|x) = \frac{\exp(r^*(x, y_1))}{\exp(r^*(x, y_1)) + \exp(r^*(x, y_2))} = \sigma(r^*(x, y_1) - r^*(x, y_2))$$

where r^* is the ground-truth reward and $\sigma(x) = 1/(1 + \exp(-x))$ is the sigmoid function. Under this model, RLHF learns the reward function by

minimizing the MLE loss:

$$\mathcal{L}_{\text{rew}}(r_\phi; \mathcal{D}) = -\mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}}[\log \sigma(r_\phi(x, y^+) - r_\phi(x, y^-))] \quad (1)$$

where $\mathcal{D}$ is the empirical distribution over the training dataset.

Using the learned reward function r_ϕ, RLHF then applies Reinforcment Learning (RL) to train the policy model π_θ, maximizing the objective

$$\mathbb{E}_{\substack{x\sim\mathcal{D}_{\text{RL}} \\ y\sim\pi(\cdot|x)}} [r_\phi(x, y) - \beta \cdot \text{KL}(\pi_\theta(\cdot|x)\|\pi_{\text{sft}}(\cdot|x))] \quad (2)$$

where $\mathcal{D}_{\text{RL}}$ is the empirical distribution over the training dataset for the RL phase, $\text{KL}(\cdot\|\cdot)$ is the KL-divergence, and π_{sft} is the model learned after the SFT phase. The regularization term $-\beta \cdot \text{KL}(\pi_\theta(\cdot|x)\|\pi_{\text{sft}}(\cdot|x))$ is used to mitigate the distribution shift issue and to prevent over-optimization. Typically, PPO (Schulman et al., 2017) is used to optimize the objective function.

Direct preference learning. Besides RLHF, directly learn from human preference may also be directly learned (Rafailov et al., 2023; Zhao et al., 2023; Azar et al., 2024) without the need of first learning a reward function. Among Direct Preference Optimization (Rafailov et al., 2023) (DPO) directly optimizes the following objective

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \mathcal{D}) = \mathbb{E}_{(x,y^+,y^-)\sim\mathcal{D}}\Big[-\log \sigma\left(\beta\frac{\pi_\theta(x, y^+)}{\pi_{\text{sft}}(x, y^+)} - \beta\frac{\pi_\theta(x, y^-)}{\pi_{\text{sft}}(x, y^-)}\right)\Big]$$

By plugging in the optimal solution to the KL-constrained objective in Eq. (2)

$$r(x, y) = \beta\frac{\pi_\theta(x, y)}{\pi_{\text{sft}}(x, y)} + \beta \log Z(x)$$

it is shown that DPO optimizes the same objective as in Eq. (1).

2.1 RLHF as a classification problem

RLHF can be rewritten as a classification problem over the data (w, z), where the input $w = (x, y_1, y_2)$ is a tuple of one prompt and two responses, and the label $z = \mathbb{1}(y_1 \succ y_2) \in \{0, 1\}$ indicates whether human prefers y_1 or y_2. The prediction model can be expressed by the reward model:

$$p_\phi(w) = \sigma(r_\phi(y_1|x) - r_\phi(y_2|x))$$

or directly by the preference model:

$$p_\theta(w) = \sigma\left(\log\frac{\pi_\theta(y_1|x)}{\pi_{\text{sft}}(y_1|x)} - \log\frac{\pi_\theta(y_2|x)}{\pi_{\text{sft}}(y_2|x)}\right).$$

In either case, RLHF can be expressed as learning the prediction model p with Binary Cross-Entropy (BCE) loss

$$\mathcal{L}_{\text{BCE}}(p; \mathcal{D}) = \sum_{i=1}^{n} l_i \quad (3)$$

where $l_i = -z_i \log p(w_i) + (1 - z_i) \log(1 - p(w_i))$

Label smoothing. Label smoothing may be applied to classification losses to improve generalization and tolerance to label noises (Szegedy et al., 2016; Mitchell, 2023; Zhu et al., 2024; Liang et al., 2024). Let α_i denote the label smoothing parameter for datapoint (w_i, z_i), then the BCE loss with label smoothing is given by

$$\mathcal{L}^\alpha_{\text{BCE}}(p; \mathcal{D}) = \sum_{i=1}^{n} l^\alpha_i \quad (4)$$

where $l^\alpha_i = -(1 - \alpha_i) \cdot \big(z_i \log p(w_i) + (1 - z_i) \log(1 - p(w_i))\big) - \alpha_i \cdot \big((1 - z_i) \log p(w_i) + z_i \log(1 - p(w_i))\big)$.

3 Gradient Estimation

In this section, we analyze the influence of label smoothing parameters on gradient estimation. Consider the BCE loss in Eq. (3) . For each l_i, its population version $l^*_i := \mathbb{E}_{z_i}[l_i]$ represents the average loss on w_i under the ground truth distribution $z_i \sim \text{Bernoulli}(p^*(w_i))$, where $p^*(w_i) = \sigma(r^*(x, y_1) - r^*(x, y_2))$ in Bradley-Terry model. Essentially, l^*_i is the KL-distance between the true distribution of z_i and the model prediction, $\text{Bernoulli}(p_\theta(w_i))$. Our objective is to minimize l^*_i, which is equivalent to fitting $p_\theta(w_i)$ to $p^*(w_i)$.

In gradient-based methods, we estimate gradients using $\nabla l^{\alpha_i}_i$ instead of $\nabla(l^*_i)$. The parameter α_i controls the trade-off between the bias and variance: when $\alpha = 0$, the gradient estimate is unbiased, in the sense that $\mathbb{E}_{z_i}[\nabla l^{\alpha_i}_i] = \nabla(l^*_i)$; when $\alpha_i = 0.5$, the gradient estimate has zero variance, i.e., $\text{Var}(\nabla l^{\alpha_i}_i) = 0$. Therefore, it is crucial to determine the α^*_i that minimizes the distance between $\nabla l^{\alpha_i}_i$ and ∇l^*_i. This is answered by the following theorem.

Theorem 3.1. *We abbreviate $q := p^*(w_i)$. Consider the expected distance $\mathbb{E}_{z_i}[d(\nabla l^{\alpha_i}_i, \nabla l^*_i)]$ between the gradient estimate $\nabla l^{\alpha_i}_i$ and the expected gradient ∇l^*_i under distance d, then:*

1. *if* $d(u,v) = \|u-v\|_\beta^\beta$ *where* $\|\cdot\|_\beta$ *is the* ℓ_β *metric* ($\beta > 1$)*, the expected distance* $\mathbb{E}_{z_i}[d(\nabla l_i^{\alpha_i}, \nabla l_i^*)]$ *is minimized when* α^* *is the unique solution of the equation*

$$q(q+x-1)^{\beta-1} - (1-q)(q-x)^{\beta-1} = 0$$

2. *when* $d(u,v) = \mathbb{1}(u \neq v)$ *is the* ℓ_0 *metric, the expected distance* $\mathbb{E}_{z_i}[d(\nabla l_i^{\alpha_i}, \nabla l_i^*)]$ *is minimized when*

$$\alpha_i^* = \min\{q, 1-q\}.$$

The proof is found in Appendix A. The above result characterizes the optimal choice of the label-smoothing parameter under a range of distance measures. Specifically, for α_2, the optimal α^* has the closed form expression

$$\alpha_i^* = 2p^*(w_i) - 2p^*(w_i)^2.$$

Figure 2 illustrates the curves of the optimal

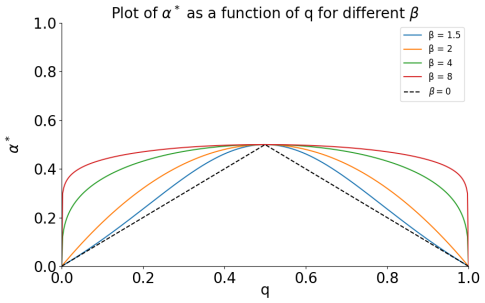


Figure 2: Optimal label smoothing parameter.

label-smoothing parameter with respect to q, under different β. It's observed that α^* always lie in $[\min\{q, 1-q\}, 0.5]$, indicating a conservative loss function. Furthermore, α^* increases with $\min\{q, 1-q\}$: when $q = 1/2$, the optimal parameter $\alpha^* = 1/2$ and $\nabla l_i = \nabla l^*$; when $q = 0, 1$, the optimal parameter $\alpha^* = 0$. This aligns with the intuition that the label smoothing parameter should be small when the label has high confidence, while it can be set higher when the label is uncertain.

4 Confidence Aware Label Smoothing (CALS)

Motivated by the previous section, we develop confidence-aware label smoothing where the label smoothing parameter is set to reflect the confidence level of the label. Ideally, we hope the parameter for data (w, z) to be certain function of the confidence *per data*: $p^*(w) = \mathbb{P}^*(z = 1|w)$. However, $p^*(w)$ is usually not accessible and difficult to estimate as each individual prompt only appears once or a few times in the RLHF dataset.

To deal with this issue, we define a function $\widetilde{\alpha} : [0,1] \to [0, 0.5)$ such that for any $x \in [0,1]$,

$$\widetilde{\alpha}(x) = (\mathbb{P}^*(z \neq \mathbb{1}(x > 0/5)|p_\theta(w) = x))_{<0.5}$$

where the probability $\mathbb{P}^*$ is with respect to $w \sim \mathcal{D}$ and $z \sim \text{Bernoulli}(p^*(w))$, and the notation $(\cdot)_{<0.5}$ is defined with $(x)_{<0.5} = x \cdot \mathbb{1}(x < 0.5), \forall x \in [0,1]$. Then for each data point w, we set the label-smoothing parameter to be $\widetilde{\alpha}(p_\theta(w))$. To understand this quantity, first notice that $p_\theta(w) = x$ means the model forecast that $z = 1$ happens with probability x, and $x > 0/5$ means the model predicts that $z = 1$ is more likely to happen. Therefore, $\mathbb{P}^*(z \neq \mathbb{1}(x > 0/5)|p_\theta(w) = x)$ represents the probability that the model p_θ's prediction $\mathbb{1}(x > 0/5)$ misaligns with the realization z, conditional on that the model forecasts x as the probability. Then we clip the conditional probability with $(\cdot)_{<0.5}$ to ensure that it always falls in $[0, 0.5)$. Finally, $\widetilde{\alpha}(p_\theta(w))$ expresses the frequency that the prediction fails to align with realization among the data where the forecast is $p_\theta(w)$, thereby reflecting the true confidence level of the label *per forecast*, instead of *per data*. This quantity is related to notion of calibration (Foster and Vohra, 1998; Hébert-Johnson et al., 2018), and it converges to the true probability $p_\theta^*(w)$ under certain regularity conditions (Blasiok et al., 2024). The lower the confidence level, the higher the value of $\widetilde{\alpha}$, suggesting that the gradient update should be more conservative.

This gives rise to the BCE loss with confidence-aware label smoothing:

$$\mathcal{L}_{\text{CALS}}^{\alpha}(p; \mathcal{D}) = \sum_{i=1}^{n} l_i^{\widetilde{\alpha}} \tag{5}$$

where $l_i^{\widetilde{\alpha}} = -(1 - \widetilde{\alpha}(p(w_i))) \cdot \big(z_i \log p(w_i) + (1 - z_i)\log(1 - p(w_i))\big) - \widetilde{\alpha}(p(w_i)) \cdot \big((1 - z_i)\log p(w_i) + z_i \log(1 - p(w_i))\big)$.

4.1 DPO with CALS

Now we introduce our DPO algorithm with confidence-aware label smoothing, detailed in Algorithm 1. The algorithms takes the preference dataset $\mathcal{D}$ and a prior α_0 as inputs. We use a symmetric (over line $x = 1/2$) piecewise-constant function $\alpha : [0,1] \to [0, 0.5)$ to model $\widetilde{\alpha}$. In partic-

Algorithm 1 DPO with confidence-aware label smoothing

Input: $\mathcal{D} = \{(x_i, y_i^+, y_i^-), i \in [n]\}, \alpha_0$
for $t = 1, 2, \ldots, T$ **do**
  Sample mini-batch $\mathcal{D}_t \subset \mathcal{D}$
  Apply gradient descent

$$\theta_{t+1} \leftarrow \theta_t - \eta \cdot \nabla_\theta \mathcal{L}^{\alpha}_{\mathrm{CALS}}(\pi_{\theta_t}, \mathcal{D}_t)$$

with the loss function over batch $\mathcal{D}_t$

$$\mathcal{L}^{\alpha}_{\mathrm{CALS}}(\pi_\theta; \mathcal{D}_t) = -\sum_{j \in \mathcal{D}} \left((1 - \alpha(p_\theta(w_j))) \cdot \log p_\theta(w_j) + \alpha(p_\theta(w_j)) \cdot \log\left(1 - p_\theta(w_j)\right)\right)$$

where

$$\alpha(p_\theta(w_j)) = \widehat{\alpha}_k, \text{ if } p_\theta(w_j) \in \left[\frac{1}{2} + \frac{k}{2K}, \frac{1}{2} + \frac{(k+1)}{2K}\right] \cup \left[\frac{1}{2} - \frac{k}{2K}, \frac{1}{2} - \frac{(k+1)}{2K}\right]$$

$$p_\theta(w_j) = \sigma\left(\log \frac{\pi_\theta(y_j^+|x_j)}{\pi_{\mathrm{sft}}(y_j^+|x_j)} - \log \frac{\pi_\theta(y_j^-|x_j)}{\pi_{\mathrm{sft}}(y_j^-|x_j)}\right)$$

Update label smoothing parameters $(\widehat{\alpha}_0, \ldots, \widehat{\alpha}_{K-1})$:

$$\widehat{\alpha}_k \leftarrow \frac{\alpha_0 \cdot N_k + \sum_{i=1}^{t} \sum_{(w,z) \in \mathcal{D}_i} \mathbb{1}\left(p_{\theta_i}(w) \in \left[\frac{1}{2} - \frac{k}{2K}, \frac{1}{2} - \frac{(k+1)}{2K}\right]\right)}{N_k + \sum_{i=1}^{t} \sum_{(w,z) \in \mathcal{D}_i} \mathbb{1}\left(p_{\theta_i}(w) \in \left[\frac{1}{2} + \frac{k}{2K}, \frac{1}{2} + \frac{(k+1)}{2K}\right] \cup \left[\frac{1}{2} - \frac{k}{2K}, \frac{1}{2} - \frac{(k+1)}{2K}\right]\right)}. \tag{6}$$

end for
Return π_{θ_T}

ular, we discretize $[0, 1]$ into $2K$ bins

$$\left[0, \frac{1}{2} - \frac{K-1}{2K}\right], \ldots, \left[\frac{1}{2} - \frac{1}{2K}, \frac{1}{2}\right],$$
$$\left[\frac{1}{2}, \frac{1}{2} + \frac{1}{2K}\right], \ldots, \left[\frac{1}{2} + \frac{K-1}{2K}, 1\right]$$

Notice that for any preference data $(w, z) = (x, y_1, y_2, z)$, the BCE loss $l(p_\theta; (w, z))$ is invariant under the flipping $(y_1, y_2) \to (y_2, y_1)$ and $z \to 1 - z$. Thus we merge the bins $\left[\frac{1}{2} + \frac{k}{2K}, \frac{1}{2} + \frac{(k+1)}{2K}\right]$ and $\left[\frac{1}{2} - \frac{k}{2K}, \frac{1}{2} - \frac{(k+1)}{2K}\right]$, and assign the same value $\hat{\alpha}_k$. Eq. (6) displays how we update $\hat{\alpha}_k$. At initialization, all $\hat{\alpha}_k$'s are set as the prior α_0. The value N_k represents the weight of the prior: a larger N_k indicates that $\hat{\alpha}_k$ is updated more slowly and remains closer to the prior. In the denominator, the second term expresses the number of the past data where the model's forecast lies in interval $\left[\frac{1}{2} + \frac{k}{2K}, \frac{1}{2} + \frac{(k+1)}{2K}\right] \cup \left[\frac{1}{2} - \frac{k}{2K}, \frac{1}{2} - \frac{(k+1)}{2K}\right]$. The numerator indicates the number of data out of those considered in the denominator where the model's prediction does not align with the human feedback, corresponding to the event $z \neq \mathbb{1}(p_\theta(w) > 0.5)$. Summing up, $\hat{\alpha}_k$ forms an estimate of the confidence $\widetilde{\alpha}$.

A nice property of $\widehat{\alpha}$ is that it doesn't involve additional computation related to the model. In fact, the terms $p_{\theta_i}(w_j)$ have been already computed in the forward pass of the model. As a result, the function α_t can be updated on the fly, using only pre-computed values.

4.2 Theoretical analysis

In this section, we study the theoretical property of the BCE loss in Eq. (4).

Equilibrium. The following result characterizes the global minimizer of the population loss when the label smoothing parameter is set as a function of model forecast.

Theorem 4.1. *Consider the population loss*

$$\mathcal{L}^{\alpha}_{\mathrm{CALS}}(\theta; \mathcal{D}) := \mathbb{E}\big[l(p_\theta; (w, z))\big]$$

where $l(p_\theta; (w, z)) = -(1 - \alpha(p(w))) \cdot \big(z \log p(w) + (1 - z) \log(1 - p(w))\big) - \alpha(p(w)) \cdot \big((1 - z) \log p(w) + z \log(1 - p(w))\big)$. *For any function* $\alpha : [0, 1] \to [0, 0.5)$ *, the population loss has a unique minimizer* $\widetilde{p}$*, such that for any* w*,* $\widetilde{p}(w)$ *is the unique solution of the equation*

$$\widetilde{p}(w) = p^*(w) + \alpha(\widetilde{p}(w)) - 2p^*(w)\alpha(\widetilde{p}(w)).$$

Furthermore, when α is set as

$$\widetilde{\alpha}(x) = (\mathbb{P}^*(z \neq \mathbb{1}(x > 0/5)|p_\theta(w) = x))_{<0.5}$$

then the minimizer $\widetilde{p}(w)$ is given by

$$\widetilde{p}(w) = \begin{cases} 1 - 2p^*(w) + 2p^*(w)^2, & p^*(w) > 0.5 \\ 2p^*(w) - 2p^*(w)^2, & p^*(w) \leq 0.5 \end{cases}.$$

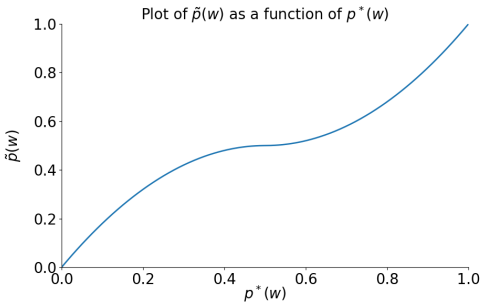


Figure 3: Equilibrium of the population loss.

The proof is deferred to Appendix B. Figure 3 illustrates $\widetilde{p}(w)$ as a function of $p^*(w)$ when $\widetilde{\alpha}(x)$ is set as $(\mathbb{P}^*(z \neq \mathbb{1}(x > 0/5)|p_\theta(w) = x))_{<0.5}$. In the equilibrium, $\alpha(p(w))$ becomes the actual probability that the preference label aligns with the model prediction. As compared to the ground truth probability, it can be observed that $\widetilde{p}(w)$ is more conservative in the sense that the prediction is closer to 0.5. This implies that the model will favor generating responses with higher likelihood to be preferred by human.

Dynamics. We discuss the gradient of the loss function on datapoint $w = (x, y^+, y^-)$, written as

$$\nabla_\theta l(p_\theta; (x, y^+, y^-)) = \lambda \cdot \left(-\nabla_\theta \log \frac{\pi_\theta(y^+|x)}{\pi_\theta(y^-|x)}\right)$$

where

$$\lambda = 1 - p_\theta(w) - \widetilde{\alpha}(p_\theta(w))$$

$$p_\theta(w) = \sigma\left(\log \frac{\pi_\theta(y^+|x)}{\pi_{\text{sft}}(y^+|x)} - \log \frac{\pi_\theta(y^-|x)}{\pi_{\text{sft}}(y^-|x)}\right).$$

In the expression of $\nabla_\theta l(p_\theta; (x, y^+, y^-))$, the component $-\nabla_\theta \log \frac{\pi_\theta(y^+|x)}{\pi_\theta(y^-|x)}$ serves as increasing the likelihood of y^+ and decreasing the likelihood of y^-. The weight λ controls the increasing/decreasing rate of the likelihood. In λ, the term $1 - p_\theta(w)$ represents how incorrectly the implicit reward model orders the completions, and the term $\widetilde{\alpha}(p_\theta(w))$ corresponds to how unconfident the human label is. Therefore, samples with incorrect model forecast (i.e., large $1 - p_\theta(w)$) and high label confidence (i.e., low $\widetilde{\alpha}(p_\theta(w))$) will achieve faster likelihood update, as depicted in Figure 1. This is desirable for stabilizing the training and preventing from overfit.

5 Experiments

5.1 Logistic regression

Task. We consider logistic regression over normalized Gaussian data in $\mathbb{R}^d$. The model family is given by $\{p_\theta(x) = \sigma(\theta^\top x) : \theta \in \mathbb{R}^d\}$. The train and test data is sampled i.i.d. from $x \sim \mathcal{N}(0, I_d/d)$, and $y \sim \text{Bernoulli}(p_{\theta^*}(x))$ where θ^* is the ground truth parameter. We focus on the near-high-dimensional setting where the dimensionality d is not significantly smaller than the number of training data. This scenario is common in RLHF, where LLMs typically possess large model sizes.

Methods. We consider two confidence-aware label smoothing methods proposed in Theorem 3.1:

1. **MLE-CALS-2**: we set the label smoothing parameter as $\alpha_i^* = 2p^*(w_i) - 2p^*(w_i)^2$, i.e., the minimizer under ℓ_2 metric.

2. **MLE-CALS-0**: we set the label smoothing parameter as $\alpha_i^* = \min\{p^*(w_i), 1 - p^*(w_i)\}$, i.e., the minimizer under ℓ_0 metric.

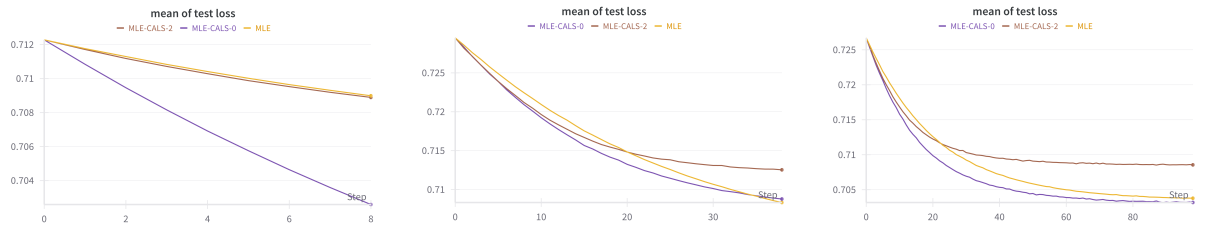
We compare the methods MLE-CALS-2, MLE-CALS-0 with standard MLE training with no label smoothing.

Evaluation. We utilize BCE loss over a large test dataset to evaluate the learned models. In particular, we set the same initialization and test dataset, and plot the mean and standard deviation of test loss of the three algorithms. The mean and standard deviation are computed over multiple samples of the training dataset and the optimizer.

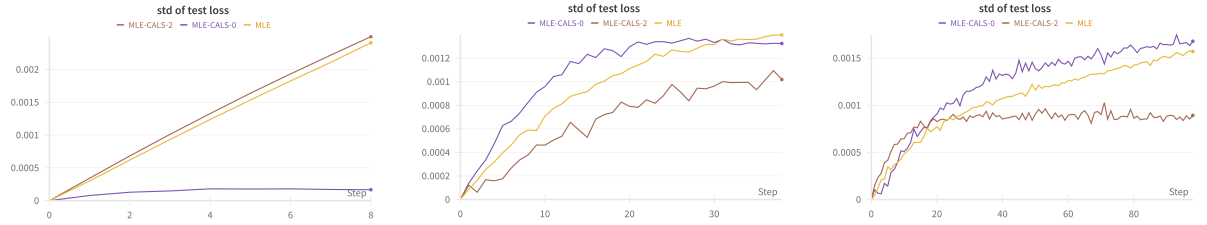
Results. Figure 4 displays the loss curves under $d = 20, 200, 500$. It can be observed that MLE-CALS-0 has lower average test losses than the other two methods, which justify our selection of this method in Algorithm 1. In comparison, the test loss under MLE-CALS-2 achieves lower variance. It is because the label smoothing parameter in MLE-CALS-2 is closer to 0.5, as shown in Figure 2, suggesting more conservative training dynamics.

5.2 Preference learning

We evaluate our proposed algorithms' ability to align with human preference.



(a) $d = 20, n_{\text{train}} = 50, \text{epoch} = 100$ (b) $d = 200, n_{\text{train}} = 10^3, \text{epoch} = 20$ (c) $d = 500, n_{\text{train}} = 10^3, \text{epoch} = 50$



(d) $d = 20, n_{\text{train}} = 50, \text{epoch} = 100$ (e) $d = 200, n_{\text{train}} = 10^3, \text{epoch} = 20$ (f) $d = 500, n_{\text{train}} = 10^3, \text{epoch} = 50$

Figure 4: Mean and standard deviation of test loss in logistic regression.

Task. Our experiments explore explore open-ended text generation task. The data come from Ultrafeedback dataset (Cui et al., 2023), a large-scale, fine-grained dataset that contains diverse preference values, and Orca (Mukherjee et al., 2023), a rich instruction-tuning dataset. We finetune two models of size 7B and 15B: Zephyr-7B (Tunstall et al., 2023b) and Starchat2-15B (Tunstall et al., 2023a). The base policies for alignment training for both models have gone through supervised fine-tuning.

Methods. We use DPO (Rafailov et al., 2023) as the baseline method. For fair comparison, we set the initialization α_0 in Algorithm 1 to be the same as the label smoothing parameter used in DPO. We follow the state-of-the-art implementations of trl (von Werra et al., 2020) and alignment-handbook (Tunstall et al., 2023b). A complete configuration of training parameters can be found in Appendix C.

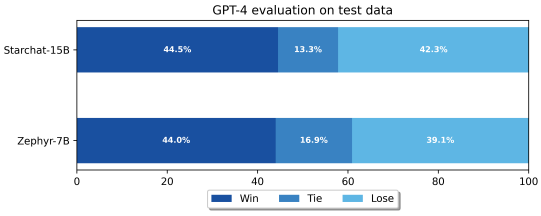


Figure 5: GPT-4 comparison between our method and the baseline, on Zephyr-7B and Starchat2-15B.

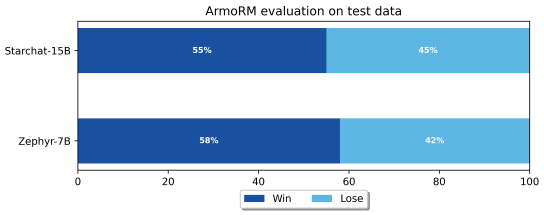


Figure 6: ArmoRM comparison between our method and the baseline, on Zephyr-7B and Starchat2-15B.

Evaluation We evaluate algorithms using head-to-head win rate between our method and the DPO baseline. We consider two evaluators: GPT-4 (OpenAI, 2023) and ArmoRM (Dong et al., 2024), since the latter is the best model on the RewardBench (Lambert et al., 2024) leaderboard[1]. The dataset is split into training set and test set. For each trained model, we generate responses on the prompts of the test dataset, and use the evaluators to compute the winning probability of our algorithm versus the DPO baseline. For GPT evaluator, we use the prompt shown in Appendix D to elicit GPT's preference on response A or B or tie, and we apply random flipping of the two responses to mitigate the bias of GPT responses. For ArmoRM, we directly compute the reward attribute 'ultrafeedback-overall_score' of the prompt-response pairs and compute the probability that our method's reward is higher.

[1]Retrieved on June 12 2024.

5.2.1 Results

The GPT and ArmoRM evaluation results are shown in Figure 5 and Figure 6 respectively. In these figures, bars with dark blue color correspond to the percentage of test data where the evaluator prefers responses from our method, and bars with light blue color correspond to the percentage of test data where the evaluator prefers responses from baseline method. GPT evaluator may also respond with 'tie', indicating the two responses are close in quality. The comparison exhibits that our method has a higher winning rate compared to baseline, in both 7B and 15B models. In the implementation, we find that it is helpful to select larger N_k for $k \leq K/2$. This is due to that in the early phase of the training there are typically a large number of data points where the model's forecast is close to 0.5, and so setting larger N_k for those forecasts can make sure that all $\hat{\alpha}_k$'s update in a similar rate.

Effect of initialization. We further vary the label smoothing parameter of DPO and accordingly the initialization α_0 in Algorithm 1. For each $\alpha \in \{0.8, 0.9, 1.0\}$, we apply our algorithm and the baseline on supervised-funetuned Zephyr-7B model, while making sure that the initialization α_0 in our method is the same as the label smoothing parameter in DPO.

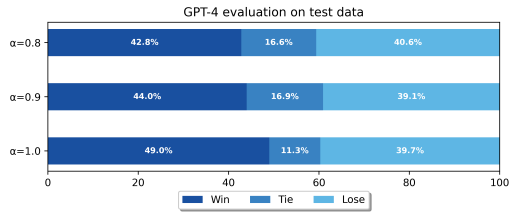


Figure 7: GPT-4 comparison between our method and the baseline, under different label smoothing parameter α.

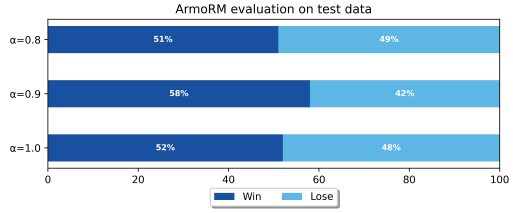


Figure 8: ArmoRM comparison between our method and the baseline, under different label smoothing parameter α.

The GPT and ArmoRM evaluation results are shown in Figure 7 and Figure 8 respectively. One can observe that our algorithm consistently outperforms DPO under different label smoothing parameter initialization, affirming that it could better align with human preference than the baseline. We notice that the initialization parameter α_0 has little affect on the overall performance of our method.

6 Conclusions

In this paper, we present Confidence Aware Label Smoothing (CALS) as a principled approach to select the label smoothing parameter in Direct Preference Optimization (DPO) for the RLHF phase of LLM training. By analyzing the trade-off between bias and variance in gradient estimation, we found that the optimal label smoothing parameter that minimizes the expected error under different distance measures should align with the confidence levels of the label. The proposed method ensures that the gradient is weighted based on both the confidence level of preference label and the correctness of model forecast, leading to more accurate alignment with human preferences. Empirical results on alignment tasks using state-of-the-art models demonstrated that CALS improves performance over baseline methods.

Limitations. While our proposed method shows promising results in enhancing the alignment of Large Language Models (LLMs) with human preferences, it comes with some limitations. First, CALS relies on the accurate estimation of confidence levels for preference labels, which can be challenging in scenarios where the data is sparse or of low quality. Second, our analysis on gradient estimation in Theorem 3.1 is difficult to generalize to larger batch sizes. Furthermore, while our empirical evaluations demonstrate competitive performance, the generalizability of CALS across different tasks and model architectures requires further investigation.

Acknowledgments

This research was conducted during Baihe Huang's internship at Microsoft Research.

References

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human

preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR.

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

Jaroslaw Blasiok, Parikshit Gopalan, Lunjia Hu, and Preetum Nakkiran. 2024. When does optimizing a proper loss yield calibration? *Advances in Neural Information Processing Systems*, 36.

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

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Jonathan D Chang, Wenhao Zhan, Owen Oertell, Kianté Brantley, Dipendra Misra, Jason D Lee, and Wen Sun. 2024. Dataset reset policy optimization for rlhf. *arXiv preprint arXiv:2404.08495*.

Jan Chorowski and Navdeep Jaitly. 2016. Towards better decoding and language model integration in sequence to sequence models. *arXiv preprint arXiv:1612.02695*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Wei Zhu, Yuan Ni, Guotong Xie, Zhiyuan Liu, and Maosong Sun. 2023. Ultrafeedback: Boosting language models with high-quality feedback. *Preprint*, arXiv:2310.01377.

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

Hanze Dong, Wei Xiong, Bo Pang, Haoxiang Wang, Han Zhao, Yingbo Zhou, Nan Jiang, Doyen Sahoo, Caiming Xiong, and Tong Zhang. 2024. Rlhf workflow: From reward modeling to online rlhf. *arXiv preprint arXiv:2405.07863*.

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

Dean P Foster and Rakesh V Vohra. 1998. Asymptotic calibration. *Biometrika*, 85(2):379–390.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. 2022. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. 2023. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*.

Xutao Guo, Yanwu Yang, Chenfei Ye, Guoqing Cai, and Ting Ma. 2024. Calseg: Improving calibration of medical image segmentation via variational label smoothing. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1601–1605. IEEE.

Ursula Hébert-Johnson, Michael Kim, Omer Reingold, and Guy Rothblum. 2018. Multicalibration: Calibration for the (computationally-identifiable) masses. In *International Conference on Machine Learning*, pages 1939–1948. PMLR.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Ujwal Krothapalli and Lynn Abbott. 2020. One size doesn't fit all: Adaptive label smoothing.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2024. Rewardbench: Evaluating reward models for language modeling. *Preprint*, arXiv:2403.13787.

Xize Liang, Chao Chen, Jie Wang, Yue Wu, Zhihang Fu, Zhihao Shi, Feng Wu, and Jieping Ye. 2024. Robust preference optimization with provable noise tolerance for llms. *arXiv preprint arXiv:2404.04102*.

Bingyuan Liu, Ismail Ben Ayed, Adrian Galdran, and Jose Dolz. 2022. The devil is in the margin: Margin-based label smoothing for network calibration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 80–88.

Tianqi Liu, Zhen Qin, Junru Wu, Jiaming Shen, Misha Khalman, Rishabh Joshi, Yao Zhao, Mohammad Saleh, Simon Baumgartner, Jialu Liu, et al. 2024. Lipo: Listwise preference optimization through learning-to-rank. *arXiv preprint arXiv:2402.01878*.

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

Mohamed Maher and Meelis Kull. 2021. Instance-based label smoothing for better calibrated classification networks. In *2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 746–753. IEEE.

Eric Mitchell. 2023. A note on dpo with noisy preferences and relationship to ipo.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *arXiv preprint arXiv:2306.02707*.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. 2019. When does label smoothing help? *Advances in neural information processing systems*, 32.

R OpenAI. 2023. Gpt-4 technical report. *ArXiv*, 2303.

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

Hyekang Park, Jongyoun Noh, Youngmin Oh, Donghyeon Baek, and Bumsub Ham. 2023. Acls: Adaptive and conditional label smoothing for network calibration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3936–3945.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Shyam Sundhar Ramesh, Yifan Hu, Iason Chaimalas, Viraj Mehta, Pier Giuseppe Sessa, Haitham Bou Ammar, and Ilija Bogunovic. 2024. Group robust preference optimization in reward-free rlhf. *arXiv preprint arXiv:2405.20304*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2024. Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18990–18998.

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

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

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

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Shengyi Huang, Kashif Rasul, Alexander M. Rush, and Thomas Wolf. 2023a. The alignment handbook. https://github.com/huggingface/alignment-handbook.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, et al. 2023b. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. 2020. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl.

Haoxiang Wang, Yong Lin, Wei Xiong, Rui Yang, Shizhe Diao, Shuang Qiu, Han Zhao, and Tong

Zhang. 2024. Arithmetic control of llms for diverse user preferences: Directional preference alignment with multi-objective rewards. *arXiv preprint arXiv:2402.18571*.

Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*.

Tianhao Wu, Banghua Zhu, Ruoyu Zhang, Zhaojin Wen, Kannan Ramchandran, and Jiantao Jiao. 2023. Pairwise proximal policy optimization: Harnessing relative feedback for llm alignment. *arXiv preprint arXiv:2310.00212*.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. 2023. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*.

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

Banghua Zhu, Jiantao Jiao, and Michael I Jordan. 2023a. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. *arXiv preprint arXiv:2301.11270*.

Banghua Zhu, Michael I Jordan, and Jiantao Jiao. 2024. Iterative data smoothing: Mitigating reward overfitting and overoptimization in rlhf. *arXiv preprint arXiv:2401.16335*.

Banghua Zhu, Hiteshi Sharma, Felipe Vieira Frujeri, Shi Dong, Chenguang Zhu, Michael I Jordan, and Jiantao Jiao. 2023b. Fine-tuning language models with advantage-induced policy alignment. *arXiv preprint arXiv:2306.02231*.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. 2018. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710.

A Proof of Theorem 3.1

Proof. We first rewrite the loss

$$l_i^{\alpha_i} = u \cdot \log p(w_i) + (-1-u) \cdot \log(1-p(w_i))$$

where the coefficient u is defined as

$$u = 2\alpha_i z_i - \alpha_i - z_i.$$

The gradient of the empirical loss is given by

$$\nabla l_i^{\alpha_i} = \left(\frac{u}{p(w_i)} + \frac{1+u}{1-p(w_i)}\right) \cdot \nabla p(w_i)$$

The gradient of the population loss is given by

$$\nabla l_i^{*} = \left(\frac{-q}{p(w_i)} + \frac{1-q}{1-p(w_i)}\right) \cdot \nabla p(w_i)$$

If $d(u,v) = \|u-v\|_\beta^\beta$. The expected distance $\mathbb{E}_{z_i}[d(\nabla l_i^{\alpha_i}, \nabla l_i^*)]$ expands as

$$\mathbb{E}\left[\left|\frac{u+q}{p(w_i)} + \frac{u+q}{1-p(w_i)}\right|^\beta \cdot \|\nabla p(w_i)\|_\beta^\beta\right]$$

$$= f(\alpha_i) \cdot \frac{\|\nabla p(w_i)\|_\beta^\beta}{p^\beta (1-p)^\beta}$$

where

$$f(x) = q \cdot |x+q-1|^\beta + (1-q) \cdot |q-x|^\beta.$$

Now it remains to minimize $f(x)$. We may assume without loss of generality that $q > 1/2$, while the case $q < 1/2$ is symmetrical. First notice that projecting any $x \neq [1-q, q]$ to the interval $[1-q, q]$ strictly decreases the value of f, thus the minimizer lies in $[1-q, q]$.

The derivative of f is given by

$$f'(x)$$
$$= q\beta(x+q-1)^{\beta-1} - (1-q)\beta(q-x)^{\beta-1}.$$

Since $f'(1-q) < 0 < f'(q)$ and f' is increasing over $[1-q, q]$, the minimizer is the unique zero of $f'(x)$, i.e. the unique solution to the equation

$$q\beta(x+q-1)^{\beta-1} - (1-q)\beta(q-x)^{\beta-1} = 0.$$

If $d(u,v) = \|u-v\|_0$. The expected distance $\mathbb{E}_{z_i}[d(\nabla l_i^{\alpha_i}, \nabla l_i^*)]$ is equal to

$$q \cdot \mathbb{1}(x = 1-q) + (1-q) \cdot \mathbb{1}(x = q).$$

It follows that the minimizer is $\min\{q, 1-q\}$. □

B Proof of Theorem 4.1

Proof. We rewrite the loss as

$$\mathbb{E}\Big[u(w) \cdot \log p_\theta(w) +$$
$$+ (-1-u(w)) \cdot \log(1-p_\theta(w))\Big]$$

where $u(w) = 2\alpha(p_\theta(w))p^*(w) - \alpha(p_\theta(w)) - p^*(w)$. Let $\widetilde{p}$ be any the equilibrium solution, first

order optimality gives

$$\widetilde{p}(w) = p^*(w) + \alpha(\widetilde{p}(w)) - 2p^*(w)\alpha(\widetilde{p}(w)).$$

Notice that this gives a injective map from $p^*(w)$ to $\widetilde{p}(w)$. It follows that $\widetilde{p}(w)$ is the unique solution of the above equation.

When

$$\widetilde{\alpha}(x) = (\mathbb{P}^*(z \neq \mathbb{1}(x > 0/5)|p_\theta(w) = x))_{<0.5}$$

we have

$$\widetilde{\alpha}(\widetilde{p}(w)) = \begin{cases} 1 - p^*(w), & \widetilde{p}(w) > 0/5 \\ p^*(w), & \widetilde{p}(w) \leq 0/5 \end{cases}.$$

Plugging back to the first order optimality condition, we have

$$\widetilde{p}(w) = \begin{cases} 1 - 2p^*(w) + 2p^*(w)^2, & p^*(w) > 0.5 \\ 2p^*(w) - 2p^*(w)^2, & p^*(w) \leq 0.5 \end{cases}.$$

This completes the proof. □

C Hyperparameters for Preference Training

Hyperparameter	Value(s)
Learning rate	5.0e-7
Batch size per device	8
Number of epochs	1
Temperature	0.05
Warmup ratio	0.1

Table 1: Hyperparameters for Zephyr-7B training. We use AdamW (Kingma and Ba, 2014) optimizer.

Hyperparameter	Value(s)
Learning rate	5.0e-7
Batch size per device	2
Number of epochs	2
Temperature	0.05
Warmup ratio	0.1

Table 2: Hyperparameters for Starchat2-15B training. We use AdamW (Kingma and Ba, 2014) optimizer.

D Prompts for GPT Evaluation

```
For the following query to a chatbot,
which response is more helpful?

Query:
<prompt>

Response A:
<response_A>

Response B:
<response_B>

FIRST provide a one-sentence
comparison of the two responses
and explain which you feel is
more helpful. SECOND, on a new
line, state only 'A' or 'B' or
'tie' to indicate which response
is more helpful. Your response
should use the format:

Comparison: <one-sentence
comparison and explanation>
More helpful: <'A' or 'B' or
'tie'>
```