Soft Preference Optimization: Aligning Language Models to Expert Distributions

Anonymous authors
Paper under double-blind review

Abstract

Preference optimization methods such as DPO often yield aligned models that are overly deterministic, reducing output diversity and increasing the risk of mode collapse. This can limit downstream applications that benefit from multiple plausible outputs, such as reasoning and search. We propose Soft Preference Optimization (SPO), a reward-model-free algorithm that controls entropy of the aligned model through a "softness" parameter. SPO minimizes a preference-based loss together with a global KL regularization term, which helps prevent unwanted distribution shifts outside the preference dataset. While the method does not rely on any reward model assumption, we provide theoretical guarantees that under a Bradley–Terry assumption, it converges to a softmax distribution over the expert rewards. We present the methodology, theoretical analysis, and comparative advantages in alignment precision and output diversity.

1 Introduction

The *alignment problem* aims to adapt a generative model—most notably, a Large Language Model (LLM)—so that its outputs reflect human preferences and ethical norms without sacrificing the breadth of ideas the model can express. Diversity matters specifically in tasks such as multi-step reasoning, open-domain question answering, and creative ideation that benefit from a spectrum of plausible responses rather than a single "best" response. Yet, standard supervised fine-tuning often amplifies popular patterns in the data and can unintentionally suppress rarer—but valuable—modes.

A widely embraced approach involves refining these models based on expert (i.e., human) preferences, typically expert-provided comparisons of pairs of model-generated outputs (Christiano et al., 2017). Given a *preference dataset* $\mathcal{D}$ and a pre-trained model π_{ref}, preference alignment seeks to train a new model, π_θ, whose outputs are better aligned with the preference in $\mathcal{D}$ (Radford et al., 2018; Ramachandran et al., 2016). A notable advancement in this field has been Reinforcement Learning from Human Feedback (RLHF), which first trains a reward model and then optimizes π_θ to maximize this learned reward while remaining close to π_{ref} (Ouyang et al., 2022). Despite its success, RLHF introduces a multi-stage pipeline and can carry over reward-model biases to the final policy.

More recent studies have proposed reward-model-free alternatives, such as Direct Preference Optimization (DPO) and its descendants (Rafailov et al., 2023; Amini et al., 2024; Chowdhury et al., 2024; Xu et al., 2024; Yin et al., 2024; Xu et al., 2023; Tunstall et al., 2023). These methods replace the reward model with a single supervised objective, simplifying implementation and reducing compute. However, emerging evidence shows that many of these objectives implicitly *sharpen* the output distribution, driving the model toward deterministic solutions and causing *mode collapse*—a loss of diversity in generated text (Murthy et al., 2024; Schoelkopf et al., 2024; Shypula et al., 2025; Kirk et al., 2024; Azar et al., 2023).

This paper introduces *Soft Preference Optimization* (SPO), a reward-model-free alignment method that minimizes a separable objective

$$\text{AlignmentLoss}(\pi_\theta, \pi_{\text{ref}}, \mathcal{D}) = \text{PreferenceLoss}(\pi_\theta, \mathcal{D}) + \text{Regularizer}(\pi_\theta, \pi_{\text{ref}}), \tag{1}$$

where the regularizer may be chosen as the KL divergence between the aligned model and the reference model. SPO explicitly controls entropy through a user-set softness parameter (α) in the PreferenceLoss. We prove (Section 3) that, under the Bradley–Terry assumption (Bradley & Terry, 1952) and in the limit of ample data, SPO converges to a softmax over scaled expert rewards, thereby offering theoretical support for entropy control. Empirically, SPO maintains higher entropy than baseline methods while matching or exceeding their alignment quality on multiple datasets and model scales.

Unlike RLHF and DPO, the development of SPO does not rely on assumptions regarding the existence of underlying rewards, such as the Bradley–Terry (BT) model (Bradley & Terry, 1952). Nevertheless, we demonstrate that if the BT model is applicable and given an asymptotically large preference dataset, SPO is theoretically guaranteed to converge to a softmax of the rewards, which inspires the designation "*Soft* Preference Optimization". Because the softness parameter α is disentangled from the KL term, SPO can retain meaningful stochasticity even when regularization is turned off, mitigating the determinism reported for DPO when its KL weight is set near zero (Azar et al., 2023).

In summary, SPO (i) aligns to expert preferences through a convex-in-logits loss that admits a closed-form fixed point, (ii) employs a global KL regularizer to avoid pathological shifts outside the preference data, and (iii) offers a direct handle on output entropy, thereby addressing growing concerns over diversity loss in aligned LLMs.

2 Background

Preference data. Let $\mathcal{X}$ be a set of queries and $\mathcal{Y}$ a set of candidate responses. Given $x \in \mathcal{X}$, a behaviour policy proposes two responses $(y_1, y_2) \in \mathcal{Y}^2$, which expert (human or strong-model) annotators then order as $y_1 \succ y_2$ if y_1 is preferred over y_2. The true expert preferences are represented by a probability, $p^*(y_1 \succ y_2 \mid x)$, reflecting the inherent randomness due to the variable nature of the experts, who may be a group of humans with slightly differing preferences. Collecting many such triples yields the pairwise preference dataset $\mathcal{D} = \{(x; y_w, y_l)\}$, where y_w (winner) is preferred to y_l (loser).

RLHF. Reinforcement Learning from Human Feedback follows a two-stage recipe. The first stage fits a reward model $r_\phi(y \mid x)$ by maximum-likelihood under the *Bradley–Terry* (BT) assumption

$$p^{\mathrm{BT}}(y_1 \succ y_2 | x) \stackrel{\mathrm{def}}{=} \sigma\big(r(y_1|x) - r(y_2|x)\big) \tag{2}$$

where σ is the sigmoid functions. The Plackett–Luce model (Plackett, 1975; Luce, 2005) extends this to n-way rankings:

$$p^{\mathrm{PL}}(y_1 \succ \cdots \succ y_n \,|\, x) \stackrel{\mathrm{def}}{=} \prod_{k=1}^{n-1} \frac{\exp\big(r(y_k|x)\big)}{\sum_{i=k}^{n} \exp\big(r(y_i|x)\big)}, \qquad \forall (x; y_1, \ldots, y_n) \in \mathcal{X} \times \mathcal{Y}^n. \tag{3}$$

The second stage maximizes

$$\mathcal{L}_{\mathrm{RLHF}} = -\mathbb{E}_{x \sim \mathcal{D},\, y \sim \pi_\theta}\big[r_\phi(y \mid x)\big] + \beta\, \mathcal{D}_{\mathrm{KL}}\big(\pi_\theta \| \pi_{\mathrm{ref}}\big)$$

via reinforcement learning algorithms such as PPO (Schulman et al., 2017). This pipeline is compute-intensive, and reward-model mis-generalization can steer the policy (Schick et al., 2023).

Direct Preference Optimization (DPO). DPO (Rafailov et al., 2023) replaces the reward model with the supervised loss

$$\mathcal{L}_{\mathrm{DPO}} = -\mathbb{E}_{\mathcal{D}}\Big[\log \sigma\Big(\beta \log \frac{\pi_\theta(y_w)}{\pi_{\mathrm{ref}}(y_w)} - \beta \log \frac{\pi_\theta(y_l)}{\pi_{\mathrm{ref}}(y_l)}\Big)\Big]. \tag{4}$$

(Rafailov et al., 2023). Under the BT model and infinite data, (4) shares the same minimizer as RLHF. However, $\mathcal{L}_{\mathrm{DPO}}$ implicitly encourages large log-likelihood gaps between y_w and y_l, which—when β or model capacity is high—drives the policy toward *determinism*, reduces output entropy, and narrows conceptual space (Azar et al., 2023; Murthy et al., 2024; Schoelkopf et al., 2024; Shypula et al., 2025; Kirk et al., 2024). SPO re-introduces an explicit entropy knob α and a *global* KL term to counteract this trend.

3 SPO - Basic

Following (1), we consider a loss function of the form:

$$\mathcal{L}_{\text{SPO}}(\pi_\theta, \pi_{\text{ref}}, \mathcal{D}) = \mathcal{L}_{\text{pref}}(\pi_\theta, \mathcal{D}) + \text{Reg}(\pi_\theta, \pi_{\text{ref}}), \tag{5}$$

where $\mathcal{L}_{\text{pref}}$ and Reg stand for *preference loss* and *regularizer*, respectively.

The regularization term $\text{Reg}(\pi_\theta, \pi_{\text{ref}})$ ensures that π_θ avoids producing outputs highly improbable under π_{ref}. A common and effective choice is KL divergence, $\mathcal{D}_{\text{KL}}(\pi_\theta \parallel \pi_{\text{ref}})$, though other options are viable (Zhao et al., 2022). Notably, $\text{Reg}(\pi_\theta, \pi_{\text{ref}})$ does not depend on the preference dataset $\mathcal{D}$. This is because within $\mathcal{D}$, the model aims to fit to the target preferences, making additional regularization within $\mathcal{D}$ unnecessary. Instead, the regularization primarily aims to constrain π_θ outside $\mathcal{D}$ to avoid unwanted distribution shifts. This contrasts with DPO and several other approaches (see Section 8), which apply regularization only within $\mathcal{D}$.

We define a model's preference as the probability that it favors one response over another. Formally, for a query x and two responses y_1 and y_2, the model π_θ prefers y_1 over y_2 with probability:

$$\mathcal{P}_{\pi_\theta}(y_1 \succ y_2 \,|\, x) = \frac{\pi_\theta(y_1|x)}{\pi_\theta(y_1|x) + \pi_\theta(y_2|x)}. \tag{6}$$

This represents the likelihood that π_θ generates y_1 given that it produces either y_1 or y_2. We can then employ log-likelihood loss to measure the alignment of preference-probabilities' with the preference-dataset labels,

$$-\mathbb{E}_{(x;y_w,y_l)\sim\mathcal{D}}\big[\log \mathcal{P}_{\pi_\theta}(y_w \succ y_l \mid x)\big]. \tag{7}$$

We consider a preference loss $\mathcal{L}^{\alpha}_{\text{pref}}(\pi_\theta, \mathcal{D})$ that extends the above cross entropy loss by employing arbitrary exponents for π_θ. Specifically, we let for any $\alpha > 0$,

$$\mathcal{L}^{\alpha}_{\text{pref}}(\pi_\theta) \stackrel{\text{def}}{=} -\frac{1}{\alpha}\mathbb{E}_{(x;y_w,y_l)\sim\mathcal{D}}\left[\log \frac{\pi_\theta(y_w|x)^\alpha}{\pi_\theta(y_w|x)^\alpha + \pi_\theta(y_l|x)^\alpha}\right], \tag{8}$$

where the expectation is over $(x; y_w, y_l) \sim \mathcal{D}$. For $\alpha = 0$, we *define* the loss

$$\mathcal{L}^{0}_{\text{pref}}(\pi_\theta) \stackrel{\text{def}}{=} -\frac{1}{2}\mathbb{E}_{(x;y_w,y_l)\sim\mathcal{D}}\left[\log \frac{\pi_\theta(y_w \mid x)}{\pi_\theta(y_l \mid x)}\right]. \tag{9}$$

This definition is chosen so that $\nabla_\theta \mathcal{L}^{0}_{\text{pref}}(\pi_\theta)$ coincides with $\lim_{\alpha\downarrow 0} \nabla_\theta \mathcal{L}^{\alpha}_{\text{pref}}(\pi_\theta)$, even though $\mathcal{L}^{0}_{\text{pref}}$ is *not* the pointwise limit of $\mathcal{L}^{\alpha}_{\text{pref}}$ as $\alpha \to 0$. For completeness, the gradient-limit derivation is provided in Appendix A. As $\alpha \to 0$, the gradient of $\mathcal{L}^{\alpha}_{\text{pref}}$ approaches that of $\mathcal{L}^{0}_{\text{pref}}$. The gradient $\nabla_\theta \mathcal{L}^{\alpha}_{\text{pref}}(\pi_\theta)$, for $\alpha > 0$, is

$$-\nabla_\theta \mathcal{L}^{\alpha}_{\text{pref}}(\pi_\theta) = \mathbb{E}_{(x;y_w,y_l)\sim\mathcal{D}}\left[\frac{\pi_\theta(y_l|x)^\alpha}{\pi_\theta(y_w|x)^\alpha + \pi_\theta(y_l|x)^\alpha} \nabla_\theta \log \frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)}\right].$$

The term $\pi_\theta(y_l|x)^\alpha / \big(\pi_\theta(y_w|x)^\alpha + \pi_\theta(y_l|x)^\alpha\big)$ quantifies the model's error in preferring y_w over y_l and scales the update accordingly—larger deviations of model's preference from the dataset labels lead to larger adjustments. The loss $\mathcal{L}^{\alpha}_{\text{pref}}(\pi_\theta, \mathcal{D})$ recovers the cross-entropy loss in (7) when $\alpha = 1$. The parameter α controls the model's entropy: larger values encourage higher entropy, while smaller values push the model toward deterministic behavior (akin to DPO), as formalized in the next theorem.

While the SPO framework does not require an underlying reward function (or BT) assumption, it is still insightful to analyze the preference loss $\mathcal{L}^{\alpha}_{\text{pref}}$ under conditions where the BT model holds. The next theorem explores the landscape of $\mathcal{L}^{\alpha}_{\text{pref}}$ under this assumption. To avoid local minima and saddle points arising from nonlinear model classes, we adopt a *tabular model* that encodes $\pi_\theta(y|x)$ for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ as a large vector. We further relax the standard full-support assumption[1] to a more practical *connected support* condition. We say a distribution $\mathcal{D}$ has connected support if the graph formed by responses as nodes and edges (y_w, y_l) for pairs with non-zero probability in $\mathcal{D}$ is connected.

[1] Full support in this context means that the probability distribution assigns a non-zero sampling probability to all $(x; y_w, y_l) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}$.

Theorem 1. *Suppose that the BT model holds with rewards $r(\cdot|x)$, and fix any probability distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y} \times \mathcal{Y}$ that has connected support and is consistent with the BT assumption.*[2] *Then, for any $\alpha \geq 0$, in the tabular model, $\mathcal{L}^{\alpha}_{\text{pref}}$ has a unique minimizer* $\text{Softmax}(r(\cdot|x)/\alpha)$ *(reducing to* $\operatorname{argmax} r(\cdot|x)$ *for $\alpha = 0$). Furthermore, , in the tabular setting, this minimizer is globally absorbing, and the landscape of $\mathcal{L}^{\alpha}_{\text{pref}}$ contains no other first-order stationary point (i.e., no other local minima, local maxima, or saddle points).*

The proof is given in Appendix B. Theorem 1 shows that the minimizer of $\mathcal{L}^{\alpha}_{\text{pref}}$ is the softmax of BT rewards scaled by $1/\alpha$, with α controlling the model's entropy. In the asymptotically large data limit, when $\alpha = 1$, the loss is minimized by the *BT expert model*, defined as $\text{Softmax}(r(\cdot|x))$, that generates the preference labels.

4 The General SPO Algorithm

We further extend the SPO preference loss by introducing sample weighting, where weights may depend on π_θ. This affects only the optimization dynamics, not the fixed point, as shown below.

A function $\mu : \mathcal{Y} \times \mathcal{Y} \times \mathcal{X} \to \mathbb{R}^+$ is called *symmetric positive* if $\mu(y_1, y_2 \mid x) = \mu(y_2, y_1 \mid x) > 0$ for all $x \in \mathcal{X}$ and $y_1, y_2 \in \mathcal{Y}$. Given such a function μ and $\alpha > 0$, we define *weighted* preference loss as

$$\mathcal{L}^{\alpha,\mu}_{\text{pref}}(\pi_\theta) \stackrel{\text{def}}{=} -\frac{1}{\alpha}\mathbb{E}\left[\mu(y_w, y_l|x)\ \log\ \frac{\pi_\theta(y_w|x)^\alpha}{\pi_\theta(y_w|x)^\alpha + \pi_\theta(y_l|x)^\alpha}\right]. \tag{10}$$

For $\alpha = 0$ this simplifies to

$$\mathcal{L}^{0,\mu}_{\text{pref}}(\pi_\theta) \stackrel{\text{def}}{=} -\frac{1}{2}\mathbb{E}\left[\mu(y_w, y_l|x)\ \log\ \frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)}\right]. \tag{11}$$

The weight function μ adjusts the influence of individual samples in the loss. Its utility lies in recognizing that not all preference pairs are equally significant. For example, downweighting samples where both responses are low quality (e.g., low probability) can be beneficial. This can be achieved, for instance, by setting

$$\mu(y_1, y_2|x) = 2\,\sigma\Big(\big(\pi_\theta(y_1|x) + \pi_\theta(y_2|x)\big)^\gamma - \hat{\mathbb{E}}_{(y_1',y_2'|x')\sim\mathcal{D}}\left[\big(\pi_\theta(y_1'|x') + \pi_\theta(y_2'|x')\big)^\gamma\right]\Big), \tag{12}$$

where σ is the sigmoid function and $\gamma \geq 0$ is a hyperparameter (e.g., 0.01) that dampens the effect of very small probabilities. The term $\hat{\mathbb{E}}$ ensures pairwise significance is measured relative to other pairs, typically via batch averaging. When $\gamma = 0$, μ reduces to uniform weighting. While μ may depend on π_θ, it is important to note that gradient propagation through μ is not permitted. Specifically,

$$\nabla_\theta \mathcal{L}^{\alpha,\mu}_{\text{pref}}(\pi_\theta, \mathcal{D}) = \mathbb{E}\left[\mu(y_w, y_l|x)\frac{\pi_\theta(y_l|x)^\alpha}{\pi_\theta(y_w|x)^\alpha + \pi_\theta(y_l|x)^\alpha}\nabla_\theta \log\frac{\pi_\theta(y_w|x)}{\pi_\theta(y_l|x)}\right].$$

Interestingly, the weight function μ only affects the optimization process, not the final fixed point, under certain assumptions, as we show in the next theorem. The proof is given in Appendix B.

Theorem 2. *Suppose that the conditions of Theorem 1 hold. Then for any $\alpha \geq 0$ and any symmetric positive function μ, the softmax of the BT rewards divided by α,* $\text{Softmax}(r(\cdot|x)/\alpha)$ *(reducing to* $\operatorname{argmax} r(\cdot|x)$ *for $\alpha = 0$), is the unique globally absorbing fixed point of the differential equation $\dot{\boldsymbol{\pi}} = \prod\big(-\nabla_\theta \mathcal{L}^{\alpha,\mu}_{\text{pref}}(\pi_\theta, \mathcal{D})\big)$, where $\prod(\cdot)$ stands for projection onto the probability simplex, and the gradient is given in* (4).

To compute the $\mathcal{D}_{\text{KL}}$ regularizer in (5), we estimate $\mathcal{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})$ using online samples from π_θ. Since sampling is costly for sequential models, we instead generate a batch of samples intermittently (e.g., every T steps) and reuse them to approximate $\mathcal{D}_{\text{KL}}$ until the next batch is drawn. To reduce variance, we apply the following token-wise formula to a batch of samples (x, y) with $y \sim \pi_\theta(\cdot|x)$:

$$\widehat{\mathcal{D}_{\text{KL}}}(\pi_\theta \parallel \pi_{\text{ref}}) \stackrel{\text{def}}{=} \frac{1}{|\mathcal{B}|}\sum_{(x,y)\in\mathcal{B}}\sum_{\tau=1}^{|y|}\mathcal{D}_{\text{KL}}\Big(\pi_\theta(\,Y_\tau \mid x, y_{:\tau-1}\,) \parallel \pi_{\text{ref}}(\,Y_\tau \mid x, y_{:\tau-1}\,)\Big). \tag{13}$$

[2]Consistency with the BT holds if $\mathcal{D}(x; y_1, y_2)/\mathcal{D}(x; y_2, y_1) = p^{\text{BT}}(y_1 \succ y_2|x)/p^{\text{BT}}(y_2 \succ y_1|x) = \exp\big(r(y_1 \mid x) - r(y_2 \mid x)\big)$, for all $(x; y_1, y_2) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}$, where p^{BT} is defined in (2) and $r(\cdot|\cdot)$ is the reward function in the BT model.

Algorithm 1 SPO

for $t = 0, 1, 2, \ldots$ **do**
  if t is a multiple of T **:** # once every T iterations
    Generate a batch $\mathcal{B}$ of online samples $y \sim \pi_\theta(\cdot|x)$, for a set of recently observed $x \sim \mathcal{D}$.
  Compute $\mathcal{L}^{\alpha,\mu}_{\text{pref}}(\pi_\theta, \mathcal{D})$ from (10), using the μ function given in (12).
  Compute token-wise regularizer $\widehat{\mathcal{D}_{\text{KL}}}(\pi_\theta \parallel \pi_{\text{ref}})$ from (13), using the online samples batch $\mathcal{B}$.
  Form the SPO loss function $\mathcal{L}_{\text{SPO}}(\pi_\theta, \pi_{\text{ref}}, \mathcal{D}) = \mathcal{L}^{\alpha,\mu}_{\text{pref}}(\pi_\theta, \mathcal{D}) + \widehat{\mathcal{D}_{\text{KL}}}(\pi_\theta \parallel \pi_{\text{ref}})$.
  Update the network using an optimizer of interest for the loss function $\mathcal{L}_{\text{SPO}}(\pi_\theta, \pi_{\text{ref}}, \mathcal{D})$.

The term $\pi(Y_\tau = s \mid x, y_{1:\tau-1})$ is readily available from the network's outputs, so the sum in (13) adds negligible overhead. While $\widehat{\mathcal{D}_{\text{KL}}}$ is a biased estimate of sequence-level $\mathcal{D}_{\text{KL}}$, we found that its reduced variance outweighs the bias in practice. Like sequence-level $\mathcal{D}_{\text{KL}}$, the token-wise version remains a valid proximity measure between π_θ and π_{ref}, making it a sound regularizer. Algorithm 1 summarizes the SPO Algorithm.

5 SPO for Other Data-Types: Best-of-n Preference and Ranked Preference

In this section, we generalize the SPO algorithm for other types of preference data: best-of-n preference data and ranked-data. We extend the definition of a symmetric function to n-responses by calling a function $\mu : \mathcal{Y}^n \times \mathcal{X} \to \mathbb{R}^+$ *symmetric positive* if $\mu(y_{\tau(1)}, \ldots, y_{\tau(n)} \mid x) = \mu(y_1, \ldots, y_n \mid x) > 0$, for all $x \in \mathcal{X}$ all $y_1, \ldots, y_n \in \mathcal{Y}$, and all permutations τ of $(1, \ldots, n)$.

Best-of-n preference data: Given an $n \geq 2$, a sample $(x; y_1, \ldots, y_n; i^*)$ of a best-of-n preference dataset consists of a query x along with n responses $y_1, \ldots, y_n$, one of which (i.e., y_{i^*}) is labeled by the expert as the best response. Given a symmetric positive function μ and an $\alpha > 0$, we propose the following preference loss for a best-of-n preference dataset $\mathcal{D}$:

$$\mathcal{L}^{\alpha,\mu}_{\text{pref-}n}(\pi_\theta) \stackrel{\text{def}}{=} -\frac{1}{\alpha}\mathbb{E}\left[\mu(y_1,..., y_n|x) \log \frac{\pi_\theta(y_{i^*}|x)^\alpha}{\sum_{i=1}^n \pi_\theta(y_i|x)^\alpha}\right]. \tag{14}$$

where the expectation is over $(x; y_1, \ldots, y_n; i^*) \sim \mathcal{D}$.

As before, we stop the gradient from propagating through μ, even though μ may depend on π_θ. Similar to the pairwise case, the following theorem shows that the loss in (14) is minimized at the softmax of rewards, assuming an underlying reward function exists. In particular, given a reward function $r(\cdot|x) : \mathcal{X} \to \mathcal{Y}$ and a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}^n \times \{1, \ldots, n\}$, we say that $\mathcal{D}$ is *consistent with n-ary BT model* if for any $(x; y_1, \ldots, y_n) \in \mathcal{X} \times \mathcal{Y}^n$ and any $i, j \in \{1, \ldots, n\}$, $\mathcal{D}(x; y_1, \ldots, y_n; i)/\mathcal{D}(x; y_1, \ldots, y_n; j) = \exp\big(r(y_i \mid x) - r(y_j \mid x)\big)$. This generalizes the pairwise BT consistency condition from Section 3. Proof of the following theorem is given in Appendix C.

Theorem 3. *Consider a reward function $r(\cdot \mid x)$ and a probability distribution $\mathcal{D}$ with full support over $\mathcal{X} \times \mathcal{Y}^n \times \{1, \ldots, n\}$ that is consistent with the n-ary BT model. Then, for any $\alpha > 0$ and any symmetric positive function μ, in the tabular model,* $\text{Softmax}(r(\cdot|x)/\alpha)$ *is the unique globally absorbing fixed point of the differential equation $\dot{\boldsymbol{\pi}} = \prod\big(-\nabla_\theta \mathcal{L}^{\alpha,\mu}_{\text{pref-}n}(\pi_\theta, \mathcal{D})\big)$, where $\prod(\cdot)$ stands for projection onto the probability simplex.*

Ranked Preference Data: A ranked preference dataset consists of samples of the form $(x; y_1, \ldots, y_n; \tau)$, where x is a query, $y_1, \ldots, y_n$ are n responses, and τ is a permutation representing the relative preference $y_{\tau(1)} \succ \cdots \succ y_{\tau(n)}$ of the expert over these responses. Given an $\alpha > 0$ and a sequence of symmetric positive function $\mu_k : \mathcal{X} \times \mathcal{Y}^k \to \mathbb{R}$ for $k = 2, \ldots, n$, we propose the following preference loss for a ranked preference dataset $\mathcal{D}$:

$$\mathcal{L}^{\alpha,[\mu]}_{\text{rank}}(\pi_\theta) \stackrel{\text{def}}{=} -\frac{1}{\alpha}\mathbb{E}_{(x;y_1,\ldots,y_n;\tau)\sim\mathcal{D}} \sum_{k=1}^{n-1} \mu_k(y_{\tau(k)},..., y_{\tau(n)}|x) \log \frac{\pi_\theta(y_{\tau(k)}|x)^\alpha}{\sum_{j=k}^n \pi_\theta(y_{\tau(j)}|x)^\alpha}. \tag{15}$$

We can control the importance weight of responses in different ranks through appropriate adjustment of weight functions $\mu^1, \ldots, \mu^{n-1}$. For example, by setting $\mu_k = 0$ for $k = 2, \ldots, n-1$, $\mathcal{L}_{\text{rank}}^{\alpha,[\mu]}$ boils down to $\mathcal{L}_{\text{pref-}n}^{\alpha,\mu^1}$. Here again, the gradient is not allowed to propagate through $\mu^1, \ldots, \mu^{n-1}$, even though these functions may depend on π_θ. The following theorem shows that, assuming existence of underlying rewards under the PL model (3), the softmax of these rewards is the unique minimizer of $\mathcal{L}_{\text{rank}}^{\alpha,\mu}$. The proof relies on Theorem 3, and is given in Appendix D.

Theorem 4. *Suppose that the PL model holds with rewards $r(\cdot|x)$, and a probability distribution $\mathcal{D}$ with full support over $\mathcal{X} \times \mathcal{Y}^n \times \{\text{Identity permutation}\}$ that is consistent with the PL model.*[3] *Then, for any $\alpha > 0$ and any sequence $[\mu] = \mu^1, \ldots, \mu^{n-1}$ of symmetric positive functions, in the tabular model,* $\text{Softmax}(r(\cdot|x)/\alpha)$ *is the unique globally absorbing fixed point of the differential equation $\dot{\boldsymbol{\pi}} = \prod\big(-\nabla_\theta \mathcal{L}_{\text{rank}}^{\alpha,[\mu]}(\pi_\theta, \mathcal{D})\big)$, where $\prod(\cdot)$ stands for projection onto the probability simplex.*

6 Comparative Analysis: SPO Versus DPO

Here we contrast SPO and DPO conceptually. Detailed empirical comparison follows in Section 7.

A key difference between SPO and DPO lies in how regularization ($\mathcal{D}_{\text{KL}}$) is applied. The DPO loss function (4) applies regularization only to preference dataset samples, which is suboptimal because 1) it fails to prevent distribution shifts in unexplored regions, and 2) regularizing within the dataset can hinder alignment with the preferences. SPO instead employs a *global* KL term that is estimated from freshly sampled trajectories; this term damps undesirable shifts everywhere in output space, not only on the finite set of labeled pairs; which helps prevent undesired out-of-dataset distribution shifts. [4]

SPO has an advantage over DPO and RLHF in avoiding determinism. In cases where the preference dataset is comparable to pre-training data size, regularization ($\mathcal{D}_{\text{KL}}$) becomes unnecessary (and we can set $\beta \simeq 0$). Under such settings, DPO has been shown to converge toward near-deterministic policies (Azar et al., 2023). SPO mitigates this through two mechanisms: the explicit entropy knob α in (8) and the global regularizer. Damping large excursions away from the reference model, via global $\mathcal{D}_{\text{KL}}$, helps keep low-probability regions (and modes) "alive". This design preserves response diversity, making SPO particularly suitable for applications requiring exploration, though it incurs the computational cost of online sampling.

Moreover, SPO differs from DPO in its operational regime. DPO is inherently offline, while SPO generates fresh trajectories to estimate the global KL term.

It is noteworthy that unlike RLHF and DPO, the SPO framework does not assume the existence of an underlying reward models like the BT model. Instead, SPO's preference loss directly aligns π_θ with the preferences in the dataset, making it potentially more adaptable to broader alignment contexts. Additionally, SPO is not limited to using $\mathcal{D}_{\text{KL}}$ for regularization, unlike DPO and IPO, which depend on $\mathcal{D}_{\text{KL}}$ for derivations of their loss functions.

We also note that the DPO loss does not admit a decomposition of the form (5), where (i) the preference term is independent of π_{ref} and (ii) all dependence on π_{ref} enters only through an additive regularizer. In DPO, the reference policy appears *inside* the sigmoid logit in (4), entangling "fit-to-preferences" and "stay-close-to-π_{ref}" in a non-separable way. A simple way to see why DPO loss is non-separable is to fix a sample $(x; y_w, y_l)$ with $\pi_\theta(y_w \mid x) = \pi_\theta(y_l \mid x)$ while $\pi_{\text{ref}}(y_w \mid x) \neq \pi_{\text{ref}}(y_l \mid x)$. For SPO, both $\mathcal{L}_{\text{pref}}^{\alpha}$ and $\mathcal{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})$ are symmetric w.r.t. swapping $\pi_{\text{ref}}(y_w \mid x)$ and $\pi_{\text{ref}}(y_l \mid x)$, hence the total loss in (5) is unchanged. In contrast, the DPO logit becomes $\beta \log \frac{\pi_{\text{ref}}(y_l|x)}{\pi_{\text{ref}}(y_w|x)}$, which generally changes under the swap; therefore DPO's dependence on π_{ref} cannot be represented as an additive regularizer independent of the preference term.

[3] Consistency with the PL model (3) holds if for all $(x; y_1, \ldots, y_n) \in \mathcal{X} \times \mathcal{Y}^n$ and all permutations τ and τ', $\mathcal{D}(x; y_1, \ldots, y_n; \tau)/\mathcal{D}(x; y_1, \ldots, y_n; \tau') = p^{\text{PL}}(y_{\tau(1)} \succ \cdots \succ y_{\tau(n)}|x)/p^{\text{PL}}(y_{\tau'(1)} \succ \cdots \succ y_{\tau'(n)}|x)$.

[4] In the context of a question-answering task, the term "out-of-dataset region" refers to query-response pairs where the response is not part of the preference dataset.

7 Experiments

This section presents empirical evaluations of SPO. Codes and datasets are available online at (Anonymous, 2024).

7.1 Alignment to AlpacaEval with Llama2-7B

Experiment setting: To evaluate the performance of SPO, we trained a Llama2-7B model (Touvron et al., 2023) on a pairwise preference dataset for question-answering available in AlpacaFarm (Dubois et al., 2023), and computed the win-rates against the Llama2-7B supervised fine-tuned (SFT) model on AlpacaEval 2 (Li et al., 2023), using GPT4-Turbo API. We compared the performance of SPO with several alignment algorithms, namely PPO (Schulman et al., 2017), DPO (Rafailov et al., 2023), IPO (Azar et al., 2023), KTO (Ethayarajh et al., 2024), CPO (Xu et al., 2024), R-DPO (Park et al., 2024), and SimPO (Meng et al., 2024). the experiment includes both the basic and weighted versions of SPO, with the weight function μ given in (12). For each algorithm, we report the maximum end-of-epoch win-rate over the course of a few training epochs. Additional details are provided in Appendix E.1.

Results: Table 1 presents the win-rates and length-controlled (LC) win-rates (Dubois et al., 2023). SPO–especially with the proposed weighting scheme–achieves the highest win-rate while preserving the highest LC win-rate, indicating improved robustness to verbosity bias. LC win-rates in Table 1 are evaluated on models with hyperparameters optimized for win-rates. Notably, SPO shows better generalization to LC win-rates compared to other baselines, some of which show less than 50% LC winrate against the SFT model. Standard deviations are below 1.5%.

Table 1: Alignment of Llama2-7B on AlpacaFarm dataset. (mean win-rate; evaluation std. $< 1.5\%$).

Alignment method	**Win-rate(%)**	**LC Win-rate(%)**
SFT	50.00	50.00
PPO	56.10	50.67
R-DPO	52.50	41.49
CPO	54.10	39.38
SimPO	58.48	49.73
KTO	58.50	51.94
IPO	58.59	49.60
DPO	59.16	51.26
SPO-basic (no μ)	60.83	53.17
SPO	**61.63**	**56.25**

7.2 Alignment on UltraFeedback with Llama3-8B

Experiment setting: To test generality of the results across models, we aligned the Llama-3-8B base model on the UltraFeedback preference dataset. We compared SPO (with and without the global KL term) to DPO and SimPO, using the a similar training recipe. The win-rates are computed against GPT4-generated responses, by a GPT-4-Turbo judge.

Results: Table 2 shows that SPO attains the highest win-rate. Removing the global KL term ("SPO in-dataset $\mathcal{D}_{\text{KL}}$") yields a measurable drop, underscoring the benefit of global regularization. Standard deviations are below 1.4%.

7.3 Diversity Analysis

Table 3 and Table 4 report diversity for models fine-tuned from Llama-3 8B and Llama-2 8B, respectively. The diversities are computed for checkpoints at which we reported win-rates in Subsections 7.1 and 7.2.

Table 2: Alignment of Llama3-8B on UltraFeedback. (mean win-rate; evaluation std. $< 1.4\%$).

Alignment method	**Win-rate (%)**
DPO	15.38
SimPO	17.70
SPO in-dataset $\mathcal{D}_{\text{KL}}$	17.81
SPO (global $\mathcal{D}_{\text{KL}}$)	**19.19**

To avoid rewarding noise, we restrict our diversity evaluation to high-quality outputs only. Entropy—a common measure of diversity—can be artificially inflated by low-quality or degenerate generations. This makes it a poor standalone proxy for meaningful variety. To address this, we follow a quality-conditioned approach: we compute entropy only over responses that are *preferred over GPT4 responses* on UltraFeedback. We refer to this measure as **useful entropy**, reflecting the idea that diversity should be measured only over outputs that exhibit demonstrable utility.

Formally, useful entropy is defined as the *average per-token entropy*:

$$H(\pi) = \mathbb{E}_{(x,y)\in\mathcal{W}}\Big[-\tfrac{1}{|y|}\sum_t \log \pi(y_t \mid x, y_{<t})\Big],$$

where $\mathcal{W}$ is the set of *winner* outputs—those preferred over GPT4 outputs. Related "quality-conditioned entropy" tests exist in LLM alignment studies on mode collapse (Shypula et al., 2025).

Table 3: Useful entropy (*nats / token*) of preferred outputs in Llama3-8B.

Algorithm	Useful Entropy
DPO	2.76
SimPO	2.78
SPO	**3.53**

Table 4: Useful entropy (*nats / token*) of preferred outputs in Llama2-7B.[This table is NEW]

Algorithm	Useful Entropy
DPO	1.31
SimPO	1.28
SPO	**1.39**

As shown in Table 3, SPO exhibits substantially higher useful entropy, consistent with the theoretical role of α and the stabilizing effect of global KL regularization.

7.4 Alignment to Ranking and Best-of-n Preference Data

Experiment setting: *Dataset:* Alignment research has mainly focused on pairwise preferences, with few public datasets for other formats. To address this, we generated an n-ary ranked preference dataset ($n = 4$) using GPT-4o as the labeler. Based on TinyStories (Eldan & Li, 2023)—a synthetic collection of short stories for children aged 3–4—we created a new dataset aimed at aligning stories for older readers. It includes 5,000 samples, each with four stories generated by the 110M TinyStories model, ranked by GPT-4o for coherence and engagement. The best-of-n version was created by discarding the 2nd–4th rank labels. See AppendixF for details; the datasets are available at (Anonymous, 2024).

Training: Using the implementation from (Karpathy, 2024) and the SFT model from (Karpathy, 2023), we aligned a 110M-parameter model using both ranking and best-of-n versions of SPO. We compared it against three baselines: ranking-based DPO (Appendix A3 of (Rafailov et al., 2023)), S-DPO (Chen et al., 2024), and Best-response-SFT, which involves supervised fine-tuning on all top-ranked responses. Further details are in Appendix E.3.

Results: Tables 5 and 6 show peak win rates against the Best-response-SFT model for aligning the 110M TinyStories SFT model using various algorithms on ranking and best-of-n datasets. SPO outperforms all baselines on both tasks. The standard deviation of win rates is 1.4%.

Table 5: Alignment of TinyStories SFT model to ranking data. (mean win-rate; eval. std. $< 1.4\%$).

Alignment method	Win-rate (%)
Best-response-SFT	50.0
DPO (ranking version)	67.0
S-DPO (ranking)	55.1
SPO (ranking)	**68.5**

Table 6: Alignment of TinyStories SFT model to best-of-n data. (mean win-rate; eval. std. $< 1.4\%$).

Alignment method	Win-rate (%)
Best-response-SFT	50.0
S-DPO (best-of-n)	66.0
SPO-basic (best-of-n)	67.8
SPO (best-of-n)	**70.5**

7.5 Ablation Study

Global regularization vs in-dataset regularization: To assess the impact of global regularization, we trained a Llama2-7B SFT model using the SPO-basic algorithm, replacing the global $\mathcal{D}_{\text{KL}}$ regularizer in (13) with in-dataset alternatives. We tested three such regularizers, including the commonly used $-\log \pi(y_w|x)$ from methods like CPO, SLiC-HF (Zhao et al., 2023), and RRHF (Yuan et al., 2024). Other settings are similar to Section 7.1. Table 7 lists the tested regularizers and the corresponding peak win rates of SPO-basic. As shown, the global $\mathcal{D}_{\text{KL}}$ regularizer outperforms all in-dataset alternatives.

Weight function μ: In the AlpacaFarm experiment (Section 7.1), weighted SPO achieved a win rate of 61.63% vs. 60.83% for SPO-basic. In the best-of-n experiment (Section 7.4), weighted SPO improved the win rate of SPO -basic from 67.8% to 70.5%. In the ranking experiment, non-uniform μ did not improve performance. In all cases, we used the μ function defined in (12) with $\gamma = 0.01$, and no hyperparameter tuning was performed.

Table 7: Ablation of SPO regularizer (model: Llama2-7B, dataset: AlpacaFarm)

Regularizer type	Regularizer formula	Win-rate vs. SFT
in-dataset (log probability)	$-\mathbb{E}_{(x;y_w,y_l)\in\mathcal{D}}[\log \pi_\theta(y_w|x)]$	54.1
in-dataset (tokenwise)	$\mathbb{E}_{(x;y)\in D}\left[\sum_{\tau=0}^{|y|-1} \mathbb{E}_{Y\sim\pi_\theta(\cdot|x,y_{1:\tau})} \log \frac{\pi_\theta(Y|x,y_{1:\tau})}{\pi_{\text{ref}}(Y|x,y_{1:\tau})}\right]$	59.7
in-dataset (importance sampling)	$\mathbb{E}_{(x;y)\in D}\left[\frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)}\right]$	59.9
global	Eq. (13)	**60.8**

8 Related Works

RLHF popularized preference-based tuning for LLMs (Ouyang et al., 2022; Touvron et al., 2023). Subsequent refinements reduce reward-model over-fitting (rejection sampling (Dong et al., 2023)), cut optimisation cost (grow-and-improve loops (Gulcehre et al., 2023)) or relabel demonstrations on-line (Zhang et al., 2023). Nonetheless, RLHF inherits reward-model biases and incurs high optimization costs.

More recently, reward-model-free methods have gained popularity. As discussed in Section 2, DPO reframes alignment as supervised learning on pairwise preferences, bypassing the need for an explicit reward model (Rafailov et al., 2023). This has sparked a wide range of follow-up work. EXO introduces a reverse KL objective for better numerical stability (Ji et al., 2024), while IPO generalizes the DPO objective to mitigate deterministic collapse (Azar et al., 2023). To reduce sensitivity to noisy labels, an unbiased estimator of the DPO loss has been proposed (Chowdhury et al., 2024), and extensions like R-DPO (Park et al., 2024) and SimPO (Meng et al., 2024) introduce regularization terms to control response length or replace DPO's log-ratio with mean log probabilities. Token-level generalizations enable the use of search-based algorithms such as MCTS (Rafailov et al., 2024), while KTO extends preference optimization to unpaired datasets (Ethayarajh et al., 2024). Alternatives like BOND (Sessa et al., 2024) and BoNBoN (Gui et al., 2024) further innovate on the modeling of preference scores or data selection. Some works have modeled alignment from a game-theoretic lens. For example, Munos et al. (2023) frames preference optimization as a maximin game

between competing policies, with refinements using no-regret learning in (Rosset et al., 2024) and (Swamy et al., 2024).

Concurrent to our work, CPO (Xu et al., 2024) motivated the preference loss in (8) as a heuristic simplification of DPO by removing π_{ref}. This leads to a contrastive loss similar to SPO-basic (8). In contrast, we derived (8) and its extension in (10) as what should be truly minimized, regardless of complexity considerations. Other major differences include CPO's use of in-dataset regularization versus SPO's global regularization, the incorporation of a weighting mechanism in SPO, the generalization of SPO to other data types, and theoretical guarantees proposed in this work. As we demonstrated in experiments (see Table 7), the choice of regularizer significantly impacts performance.

Separable alignment methods like SLiC (Zhao et al., 2022) and SLiC-HF (Zhao et al., 2023) also use contrastive formulations, with the latter extending to human preference data. However, SLiC-HF regularizes only over available training data, while SPO incorporates sampled trajectories to mitigate out-of-distribution drift and enables broader generalization.

A growing body of evidence shows that strong preference optimization often *shrinks* the support of LLMs. 'Alignment Reduces Conceptual Diversity' (Murthy et al., 2024) identifies this trend in conceptual space, while others have documented surface-level effects. Schoelkopf et al. (2024) reported mode collapse in models fine-tuned with PPO and DPO, and Shypula et al. (2025) found that preference tuning reduces lexical and syntactic diversity. Kirk et al. (2024) similarly note a trade-off between alignment and output diversity under RLHF. Our experiments confirm these concerns and demonstrate that SPO's softness parameter and global KL term recover substantial entropy (Section 7.3).

The effectiveness of alignment techniques heavily depends on preference dataset quality, as noisy labels can hinder language models from capturing human intent. Liu et al. (2023) combined DPO with rejection sampling to mitigate distribution mismatches between training and expert preferences. Tunstall et al. (2023) curated a conservative preference dataset using AI feedback and GPT-4 scoring before applying DPO for alignment. (Yin et al., 2024) leveraged semantic correlations of prompts in the dataset to form more conservative response pairs. For a given prompt $(x; y_w, y_l)$, a prompt x' with a similar semantic to a tuple $(x'; y'_w, y'_l)$ is used to form more conservative pairs.

9 Limitations

SPO introduces a simple, theory-grounded alignment method; however several caveats remain. First, estimating the global KL term needs online sampling from π_θ, which is computationally costly. In our largest runs, this adding up to ≈50% runtime. Section 4 suggests intermittent batch generation to mitigate this cost, however further shrinkage warrants future studies. Due to computational constraints, we do not repeat full training across multiple random seeds, hence we do not report across-seed training variance; the error bars we discuss are evaluation uncertainty from finite evaluation dataset. Another caveat involves our evaluation focuses on win-rates and alignment leaderboards. The advantage of SPO in downstream tasks that explicitly benefit from diversity, such as multi-step reasoning or continual fine-tuning, remain to be explored in future work.

References

Afra Amini, Tim Vieira, and Ryan Cotterell. Direct preference optimization with an offset. *arXiv e-prints*, pp. arXiv–2402, 2024.

Anonymous. Spo. https://anonymous.4open.science/r/SPO-4159, 2024.

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences. *arXiv preprint arXiv:2310.12036*, 2023.

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

Yuxin Chen, Junfei Tan, An Zhang, Zhengyi Yang, Leheng Sheng, Enzhi Zhang, Xiang Wang, and Tat-Seng Chua. On softmax direct preference optimization for recommendation. *arXiv preprint arXiv:2406.09215*, 2024.

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

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

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

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback, 2023.

Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*, 2023.

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

Lin Gui, Cristina Gârbacea, and Victor Veitch. Bonbon alignment for large language models and the sweetness of best-of-n sampling. *arXiv preprint arXiv:2406.00832*, 2024.

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

Haozhe Ji, Cheng Lu, Yilin Niu, Pei Ke, Hongning Wang, Jun Zhu, Jie Tang, and Minlie Huang. Towards efficient exact optimization of language model alignment. In *International Conference on Machine Learning*, pp. 21648–21671. PMLR, 2024.

Karpathy. https://huggingface.co/karpathy/tinyllamas/commit/5c22d0a5f31b635f3e4bf8f2d4dd87363ae3a275, 2023.

Andrej Karpathy. llama2.c: Inference llama 2 in one file of pure c. https://github.com/karpathy/llama2.c, 2024. GitHub repository.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of rlhf on llm generalisation and diversity. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=PXD3FAVHJT.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`, 2023.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. Statistical rejection sampling improves preference optimization. In *The Twelfth International Conference on Learning Representations*, 2023.

R Duncan Luce. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2005.

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

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

Sonia K. Murthy, Tomer Ullman, and Jennifer Hu. One fish, two fish, but not the whole sea: Alignment reduces language models' conceptual diversity, 2024. URL `https://arxiv.org/abs/2411.04427`.

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.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization. *arXiv preprint arXiv:2403.19159*, 2024.

Robin L Plackett. The analysis of permutations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 24(2):193–202, 1975.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018. URL `https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf`.

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

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

Prajit Ramachandran, Peter J Liu, and Quoc V Le. Unsupervised pretraining for sequence to sequence learning. *arXiv preprint arXiv:1611.02683*, 2016.

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

Timo Schick, Jane Yu, Roberto Dessi, Roberta Raileanu, et al. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*, 2023.

Hailey Schoelkopf et al. Attributing mode collapse in the fine-tuning of large language models. In *ICLR 2024 Workshop on Understanding of Foundation Models (ME-FoMo)*, 2024. URL `https://openreview.net/forum?id=3pDMYjpOxk`.

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

Pier Giuseppe Sessa, Robert Dadashi, Léonard Hussenot, Johan Ferret, Nino Vieillard, Alexandre Ramé, Bobak Shahriari, Sarah Perrin, Abe Friesen, Geoffrey Cideron, et al. Bond: Aligning llms with best-of-n distillation. *arXiv preprint arXiv:2407.14622*, 2024.

Alexander Shypula, Shuo Li, Botong Zhang, Vishakh Padmakumar, Kayo Yin, and Osbert Bastani. Evaluating the diversity and quality of llm generated content. *arXiv preprint arXiv:2504.12522*, 2025. URL `https://arxiv.org/abs/2504.12522`.

Gokul Swamy, Christoph Dann, Rahul Kidambi, Zhiwei Steven Wu, and Alekh Agarwal. A minimaximalist approach to reinforcement learning from human feedback. *arXiv preprint arXiv:2401.04056*, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

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

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *arXiv preprint arXiv:2401.08417*, 2024.

Jing Xu, Andrew Lee, Sainbayar Sukhbaatar, and Jason Weston. Some things are more cringe than others: Preference optimization with the pairwise cringe loss. *arXiv preprint arXiv:2312.16682*, 2023.

Yueqin Yin, Zhendong Wang, Yi Gu, Hai Huang, Weizhu Chen, and Mingyuan Zhou. Relative preference optimization: Enhancing llm alignment through contrasting responses across identical and diverse prompts. *arXiv preprint arXiv:2402.10958*, 2024.

Hongyi Yuan, Zheng Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback. *Advances in Neural Information Processing Systems*, 36, 2024.

Tianjun Zhang, Fangchen Liu, Justin Wong, Pieter Abbeel, and Joseph E Gonzalez. The wisdom of hindsight makes language models better instruction followers. *arXiv preprint arXiv:2302.05206*, 2023.

Yao Zhao, Mikhail Khalman, Rishabh Joshi, Shashi Narayan, Mohammad Saleh, and Peter J Liu. Calibrating sequence likelihood improves conditional language generation. In *The Eleventh International Conference on Learning Representations*, 2022.

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.

Appendices

A Gradient limit as $\alpha \downarrow 0$

Here we justify the $\alpha = 0$ definition in (9) by showing that its gradient matches the $\alpha \downarrow 0$ limit of the gradient of (8). Fix a sample $(x; y_w, y_l)$ and write $a = \log \pi_\theta(y_w \mid x)$ and $b = \log \pi_\theta(y_l \mid x)$. For $\alpha > 0$, the inner term of (8) is

$$-\frac{1}{\alpha} \log \frac{e^{\alpha a}}{e^{\alpha a} + e^{\alpha b}} = \frac{1}{\alpha} \log \big(1 + e^{\alpha(b-a)}\big).$$

Differentiating w.r.t. θ yields

$$\nabla_\theta \Big[\tfrac{1}{\alpha} \log(1 + e^{\alpha(b-a)}) \Big] = \sigma\big(\alpha(b-a)\big)\, \nabla_\theta (b-a),$$

where σ is the sigmoid. Taking $\alpha \downarrow 0$ gives $\sigma(\alpha(b-a)) \to \frac{1}{2}$, hence

$$\lim_{\alpha \downarrow 0} \nabla_\theta \mathcal{L}^{\alpha}_{\text{pref}}(\pi_\theta) = -\tfrac{1}{2}\, \mathbb{E}_{(x;y_w,y_l)\sim\mathcal{D}} \Big[\nabla_\theta (a-b) \Big],$$

which is exactly $\nabla_\theta \mathcal{L}^{0}_{\text{pref}}(\pi_\theta)$ for (9).

B Proof of Theorems 1 and 2

In this appendix, we present the proof of Theorems 1 and 2. The high-level proof idea is to show that moving along the projected[5] negative gradient of the preference loss (i.e., the ODE direction) results in an absolute reduction of the Euclidean distance of π_θ from $\text{Softmax}\big(r(\cdot|x)/\alpha\big)$.

Without loss of generality, we prove the theorem for a single fixed $x \in \mathcal{X}$, and remove x from the notations, for the sake of notation simplicity.

Given the rewards $r(\cdot)$ in the Bradley-Terry model, let

$$\pi^*(\cdot) \stackrel{\text{def}}{=} \text{Softmax}\big(r(\cdot)\big). \tag{16}$$

For any $\alpha \in [0, 1)$, let

$$\pi^*_\alpha(\cdot) \stackrel{\text{def}}{=} \text{Softmax}\big(r(\cdot)/\alpha\big), \tag{17}$$

and let $\boldsymbol{\pi}^\alpha$ be its vector representation. Therefore, for any $\alpha \in [0, 1]$, and for any y,

$$\pi^*(y) = z_\alpha \,\times\, \big(\pi^*_\alpha(y)\big)^\alpha, \qquad \text{where } z_\alpha \stackrel{\text{def}}{=} \frac{\Big(\sum_{y'} e^{r(y')/\alpha}\Big)^\alpha}{\sum_{y'} e^{r(y')}}. \tag{18}$$

Moreover, it follows from the consistency of distribution $\mathcal{D}$ with the Bradley-Terry model that for any pair (y_1, y_2),

$$\frac{\mathcal{D}(y_1, y_2)}{\mathcal{D}(y_1, y_2) + \mathcal{D}(y_2, y_1)} = \mathcal{P}_{\mathcal{D}}\big(y_1 \succ y_2\big) = \frac{\exp r(y_1)}{\exp r(y_1) + \exp r(y_2)} = \frac{\pi^*(y_1)}{\pi^*(y_1) + \pi^*(y_2)}. \tag{19}$$

For any $y_1, y_2 \in \mathcal{Y}$ let

$$\tilde{\mu}(y_1, y_2) \stackrel{\text{def}}{=} \mu(y_1, y_2)\big(\mathcal{D}(y_1, y_2) + \mathcal{D}(y_2, y_1)\big). \tag{20}$$

Note that the symmetry of μ implies symmetry of $\tilde{\mu}$ with respect to its first and second arguments. Then,

$$\mu(y_1, y_2)\, \mathcal{D}(y_1, y_2) = \tilde{\mu}(y_1, y_2)\, \frac{\mathcal{D}(y_1, y_2)}{\mathcal{D}(y_1, y_2) + \mathcal{D}(y_2, y_1)} = \tilde{\mu}(y_1, y_2)\, \frac{\pi^*(y_1)}{\pi^*(y_1) + \pi^*(y_2)}, \tag{21}$$

where the last equality follows from (19).

[5] Projection on the probability simplex.

Consider a $\boldsymbol{\pi}_\theta$ in the relative interior of the probability simplex and let $\mathbf{v}$ be the negative gradient of the preference loss

$$\mathbf{v} \stackrel{\text{def}}{=} -\nabla_{\pi_\theta} \mathcal{L}_{\text{pref}}^{\alpha,\mu}(\pi_\theta, \mathcal{D}), \tag{22}$$

where $\mathcal{L}_{\text{pref}}^{\alpha,\mu}$ is defined in (4). For any $y \in \mathcal{Y}$, let $v(y)$ be the entry of $\mathbf{v}$ that corresponds to y. Then,

$$\begin{aligned}
v(y) &= \sum_{y' \in \mathcal{Y}} \mathcal{D}(y, y')\mu(y, y') \, \frac{\pi_\theta(y')^\alpha}{\pi_\theta(y)^\alpha + \pi_\theta(y')^\alpha} \, \left(\frac{d}{d\,\pi_\theta(y)} \log \pi_\theta(y) - \frac{d}{d\,\pi_\theta(y)} \log \pi_\theta(y') \right) \\
&\quad + \sum_{y' \in \mathcal{Y}} \mathcal{D}(y', y)\mu(y', y) \, \frac{\pi_\theta(y)^\alpha}{\pi_\theta(y)^\alpha + \pi_\theta(y')^\alpha} \, \left(\frac{d}{d\,\pi_\theta(y)} \log \pi_\theta(y') - \frac{d}{d\,\pi_\theta(y)} \log \pi_\theta(y) \right) \\
&= \sum_{y' \in \mathcal{Y}} \tilde{\mu}(y, y') \, \frac{\pi^*(y)}{\pi^*(y) + \pi^*(y')} \, \frac{\pi_\theta(y')^\alpha}{\pi_\theta(y)^\alpha + \pi_\theta(y')^\alpha} \, \frac{1}{\pi_\theta(y)} \\
&\quad - \sum_{y' \in \mathcal{Y}} \tilde{\mu}(y, y') \, \frac{\pi^*(y')}{\pi^*(y) + \pi^*(y')} \, \frac{\pi_\theta(y)^\alpha}{\pi_\theta(y)^\alpha + \pi_\theta(y')^\alpha} \, \frac{1}{\pi_\theta(y)} \\
&= \sum_{y' \in \mathcal{Y}} \frac{\tilde{\mu}(y, y') \left(\pi^*(y)\pi_\theta(y')^\alpha - \pi^*(y')\pi_\theta(y)^\alpha \right)}{\pi_\theta(y) \left(\pi^*(y) + \pi^*(y') \right) \left(\pi_\theta(y)^\alpha + \pi_\theta(y')^\alpha \right)},
\end{aligned} \tag{23}$$

where the first equality follows from (4) and by considering all the terms that include y either as winner (the first sum) or loser (the second sum); the second equality is due to (21) and the fact that $\tilde{\mu}$ is symmetric. To simplify the notation, for any y and y', let

$$h(y, y') \stackrel{\text{def}}{=} \frac{\tilde{\mu}(y, y')}{\pi_\theta(y)\, \pi_\theta(y') \left(\pi^*(y) + \pi^*(y') \right) \left(\pi_\theta(y)^\alpha + \pi_\theta(y')^\alpha \right)}. \tag{24}$$

Then, (23) simplifies to

$$v(y) = \sum_{y' \in \mathcal{Y}} h(y, y') \, \pi_\theta(y') \left(\pi_\theta(y')^\alpha \pi^*(y) - \pi_\theta(y)^\alpha \pi^*(y') \right). \tag{25}$$

Consequently,

$$\begin{aligned}
\mathbf{v}^T(\boldsymbol{\pi}_\theta - \boldsymbol{\pi}_\alpha^*) &= \sum_{y \in \mathcal{Y}} v(y) \left(\pi_\theta(y) - \pi_\alpha^*(y) \right) \\
&= \sum_{y, y' \in \mathcal{Y}} h(y, y') \left(\pi_\theta(y')^\alpha \pi^*(y) - \pi_\theta(y)^\alpha \pi^*(y') \right) \, \pi_\theta(y') \left(\pi_\theta(y) - \pi_\alpha^*(y) \right) \\
&= \frac{1}{2} \sum_{y, y' \in \mathcal{Y}} h(y, y') \left(\pi_\theta(y')^\alpha \pi^*(y) - \pi_\theta(y)^\alpha \pi^*(y') \right) \, \pi_\theta(y') \left(\pi_\theta(y) - \pi_\alpha^*(y) \right) \\
&\quad + \frac{1}{2} \sum_{y', y \in \mathcal{Y}} h(y', y) \left(\pi_\theta(y)^\alpha \pi^*(y') - \pi_\theta(y')^\alpha \pi^*(y) \right) \, \pi_\theta(y) \left(\pi_\theta(y') - \pi_\alpha^*(y') \right) \\
&= \frac{1}{2} \sum_{y, y' \in \mathcal{Y}} h(y, y') \left(\pi_\theta(y')^\alpha \pi^*(y) - \pi_\theta(y)^\alpha \pi^*(y') \right) \, \left(\pi_\theta(y')\pi_\theta(y) - \pi_\theta(y')\pi_\alpha^*(y) \right) \\
&\quad + \frac{1}{2} \sum_{y, y' \in \mathcal{Y}} h(y, y') \left(\pi_\theta(y')^\alpha \pi^*(y) - \pi_\theta(y)^\alpha \pi^*(y') \right) \, \left(\pi_\theta(y)\pi_\alpha^*(y') - \pi_\theta(y)\pi_\theta(y') \right) \\
&= \frac{1}{2} \sum_{y, y' \in \mathcal{Y}} h(y, y') \left(\pi_\theta(y')^\alpha \pi^*(y) - \pi_\theta(y)^\alpha \pi^*(y') \right) \, \left(\pi_\theta(y)\pi_\alpha^*(y') - \pi_\theta(y')\pi_\alpha^*(y) \right) \\
&= -\frac{z_\alpha}{2} \sum_{y, y' \in \mathcal{Y}} h(y, y') \left(\left(\pi_\theta(y')\pi_\alpha^*(y) \right)^\alpha - \left(\pi_\theta(y)\pi_\alpha^*(y') \right)^\alpha \right) \, \left(\pi_\theta(y')\pi_\alpha^*(y) - \pi_\theta(y)\pi_\alpha^*(y') \right) \\
&= -\frac{z_\alpha}{2} \sum_{y, y' \in \mathcal{Y}} h(y, y') \left(\pi_\theta(y)\pi_\theta(y') \right)^{1+\alpha} \left(\left(\frac{\pi_\alpha^*(y)}{\pi_\theta(y)} \right)^\alpha - \left(\frac{\pi_\alpha^*(y')}{\pi_\theta(y')} \right)^\alpha \right) \, \left(\frac{\pi_\alpha^*(y)}{\pi_\theta(y)} - \frac{\pi_\alpha^*(y')}{\pi_\theta(y)} \right),
\end{aligned} \tag{26}$$

where the second equality follows from (25), the fourth equality is due to the symmetry of $h(y, y')$ with respect to y and y', i.e., $h(y, y') = h(y', y)$, and the sixth equality is from (18). It is easy to see that all terms in the sum in the last line are non-negative, and the sum contains at least one non-zero term if $\boldsymbol{\pi}_\theta \neq \boldsymbol{\pi}^*_\alpha$. Therefore, $\mathbf{v}^T(\boldsymbol{\pi}_\theta - \boldsymbol{\pi}^*_\alpha) < 0$ if $\boldsymbol{\pi}_\theta \neq \boldsymbol{\pi}^*_\alpha$. Consequently, $\|\boldsymbol{\pi}_\theta - \boldsymbol{\pi}^*_\alpha\|$ is strictly decreasing when moving along $\mathbf{v}$. Since both $\boldsymbol{\pi}_\theta$ and $\boldsymbol{\pi}^*_\alpha$ lie on the probability simplex, we have $\prod(\mathbf{v})^T(\boldsymbol{\pi}_\theta - \boldsymbol{\pi}^*_\alpha) \leq \mathbf{v}^T(\boldsymbol{\pi}_\theta - \boldsymbol{\pi}^*_\alpha) < 0$. It follows that for any $\boldsymbol{\pi}_\theta$ in the relative interior of the probability simplex, projection of $\mathbf{v}$ on the probability simplex is a strictly decent direction for $\|\boldsymbol{\pi}_\theta - \boldsymbol{\pi}^*_\alpha\|$.

As a result, π^*_α is the globally absorbing unique fixed point of the ODE. Furthermore, when μ is not a function of π_θ, then π^*_α is the unique first order stationary point of the preference loss $\mathcal{L}^{\alpha,\mu}_{\text{pref}}$. In other words, $\mathcal{L}^{\alpha,\mu}_{\text{pref}}$ contains no other local mininum, local maximum, or saddle-point in the probability simplex.

C Proof of Theorem 3

This appendix presents the proof of Theorem 3. The high-level idea, akin to Appendix B, is to show that moving along the ODE direction results in an absolute reduction of the Euclidean distance of π_θ from $\text{Softmax}\big(r(\cdot|x)/\alpha\big)$. The details are however substantially different from Appendix B.

We begin with the following lemma.

Lemma 1. *For any $\eta > 0$ and any pair of n-dimensional vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ with positive entries, we have*

$$\sum_{i=1}^{n} \left(\frac{a_i}{b_i}\right)^{\eta} \left(\frac{b_i}{\sum_{j=1}^{n} b_j} - \frac{a_i}{\sum_{j=1}^{n} a_j}\right) \leq 0, \tag{27}$$

and the equality holds only if $\boldsymbol{a} = c\boldsymbol{b}$ for some scalar c.

Proof of Lemma 1. Fix an arbitrary vector $\boldsymbol{a}$ with positive entries, and consider the following function

$$f(\mathbf{x}) \stackrel{\text{def}}{=} \sum_{i=1}^{n} \left(\frac{a_i}{x_i}\right)^{\eta} \left(\frac{x_i}{\sum_{j=1}^{n} x_j} - \frac{a_i}{\sum_{j=1}^{n} a_j}\right), \qquad \text{for} \quad \mathbf{x} \in \mathbb{R}^n_+, \tag{28}$$

defined on the positive quadrant. We will show that $f(\mathbf{x}) \leq 0$, for all $x \in \mathbb{R}^n_+$. Note that if $f(\mathbf{x}) > 0$ for some x, then $f(c\mathbf{x}) = f(\mathbf{x})/c^\eta > 0$, for all $c > 0$. Therefore, without loss of generality, we confine the domain to a compact set, say to the probability simplex $\mathcal{S} \stackrel{\text{def}}{=} \{\mathbf{x} \in \mathbb{R}^*_+ : \sum_{i=1}^{n} x_i = 1\}$, and show that $f(\mathbf{x}) \leq 0$ for all $\mathbf{x} \in \mathcal{S}$. In the same vein, without loss of generality we also assume that

$$\sum_{i=1}^{n} a_i = 1. \tag{29}$$

Note that $f(\mathbf{x}) = -\infty$ on the boundary of the probability simplex, that is if $x_i = 0$ for some i. Therefore, the maximizer $\mathbf{x}^*$ of f over $\mathcal{S}$, lies in the relative interior of $\mathcal{S}$. Consequently, the gradient of the Lagrangian of f at $\mathbf{x}^*$ is zero. The Lagrangian L of f is as follows:

$$L(\mathbf{x}, \lambda) \stackrel{\text{def}}{=} f(\mathbf{x}) + \lambda \left(\sum_{i=1}^{n} x_i - 1\right), \qquad \text{for } \mathbf{x} \in \mathcal{S},\ \lambda \in \mathbb{R}. \tag{30}$$

Then,

$$\begin{aligned}
\frac{d}{d\,x_k} L(\mathbf{x}, \lambda) &= \frac{d}{d\,x_k} f(\mathbf{x}) + \lambda \\
&= \frac{d}{d\,x_k} \sum_{i=1}^{n} \left(\frac{a_i}{x_i}\right)^{\eta} \left(\frac{x_i}{\sum_{j=1}^{n} x_j} - \frac{a_i}{\sum_{j=1}^{n} a_j}\right) + \lambda \\
&= \frac{d}{d\,x_k} \sum_{i=1}^{n} \left(\frac{a_i^{\eta} x_i^{1-\eta}}{\sum_{j=1}^{n} x_j} - a_i^{1+\eta} x_i^{-\eta}\right) + \lambda \\
&= \frac{(1-\eta) a_k^{\eta} x_k^{-\eta}}{\sum_{j=1}^{n} x_j} - \frac{\sum_{i=1}^{n} a_i^{\eta} x_i^{1-\eta}}{\left(\sum_{j=1}^{n} x_j\right)^2} + \eta a_k^{1+\eta} x_k^{-\eta-1} + \lambda \\
&= (1-\eta) \left(\frac{a_k}{x_k}\right)^{\eta} + \eta \left(\frac{a_k}{x_k}\right)^{1+\eta} + \left[\lambda - \sum_{i=1}^{n} a_i^{\eta} x_i^{1-\eta}\right]
\end{aligned} \tag{31}$$

where the third equality is due to (29), and the last equality is because $\sum_j x_j = 1$. Consider a scalar function $h : \mathbb{R}_+ \to \mathbb{R}_+$ as follows

$$h(y) \stackrel{\text{def}}{=} (1-\eta)\, y^{\eta} \,+\, \eta\, y^{1+\eta} \qquad \text{for} \quad y \geq 0. \tag{32}$$

Then, (31) simplifies to

$$\frac{d}{d\,x_k} L(\mathbf{x}, \lambda) = h\left(\frac{a_k}{x_k}\right) + C(\lambda, \mathbf{x}, \boldsymbol{a}), \tag{33}$$

where $C(\lambda, \mathbf{x}, \boldsymbol{a}) = \lambda - \sum_{i=1}^{n} a_i^{\eta} x_i^{1-\eta}$ is independent of k. Therefore, letting $\nabla_{\mathbf{x}} L(\mathbf{x}, \lambda) = 0$ at $\mathbf{x} = \mathbf{x}^*$, it follows that for any $1 \leq i < j \leq n$,

$$h\left(\frac{a_i}{x_i^*}\right) = h\left(\frac{a_j}{x_j^*}\right). \tag{34}$$

We now consider two cases for η.

Case 1 ($\eta \leq 1$)**.** In this case, h defined in (32) is an strictly increasing function. Therefore, (34) implies that $a_i/x_i^* = a_j/x_j^*$, for all $i, j \leq n$. Equivalently, $\mathbf{x}^* = c\boldsymbol{a}$ for some scalar $c > 0$. In this case, from (28), $f(\mathbf{x}^*) = 0$. The lemma then follows from the fact that $\mathbf{x}^*$ is the maximizer of f.

Case 2 ($\eta > 1$)**.** In this case, h is no longer increasing. In this case, h is unimodal. Specifically, h is strictly decreasing over $\big[0, (\eta-1)/(\eta+1)\big]$ and is strictly increasing over $\big[(\eta-1)/(\eta+1), \infty\big]$. This unimodality implies that the pre-image of any $y \in \mathbb{R}_+$ (i.e., $h^{-1}(y)$) is a set of at most two points. Consequently, (34) implies that we can partition the indices $1, \ldots, n$ into two groups S_1 and S_2 such that within each group, we have $a_i/x_i^* = a_j/x_j^*$. In other words, $a_i/x_i^* = a_j/x_j^*$ for all $(i, j) \in S_1 \times S_1$ and all $(i, j) \in S_2 \times S_2$. Equivalently, the maximum point, $\mathbf{x}^*$, belongs to the set

$$X^* \stackrel{\text{def}}{=} \big\{\mathbf{x} \in \mathbb{R}_+^n :\; x_i = c_1 a_i \text{ for } i \leq k, \text{ and } x_i = c_2 a_i \text{ for } i > k, \text{ for some } c_1, c_2 > 0 \text{ and } k < n\big\}, \tag{35}$$

where we have assumed without loss of generality that $S_1 = \{1, \ldots, k\}$ and $S_2 = \{k+1, \ldots, n\}$ for some $k \leq n$. We will show that $f(\mathbf{x}) \leq 0$ for all $\mathbf{x} \in X^*$.

Fix some $\mathbf{x} \in X^*$, and corresponding constants c_1, c_2, and k, as per (35). Let $A = \sum_{i=1}^{k} a_i$ and $B = \sum_{i=k+1}^{n} a_i$. Then,

$$\begin{aligned}
f(\mathbf{x}) &= \sum_{i=1}^{n} \left(\frac{a_i}{x_i}\right)^{\eta} \left(\frac{x_i}{\sum_{j=1}^{n} x_j} - \frac{a_i}{\sum_{j=1}^{n} a_j}\right) \\
&= \sum_{i=1}^{n} \left(\frac{a_i}{x_i}\right)^{\eta} \left(\frac{x_i}{c_1 A + c_2 B} - \frac{a_i}{A+B}\right) \\
&= \sum_{i=1}^{k} c_1^{-\eta} \left(\frac{c_1 a_i}{c_1 A + c_2 B} - \frac{a_i}{A+B}\right) + \sum_{i=k+1}^{n} c_2^{-\eta} \left(\frac{c_2 a_i}{c_1 A + c_2 B} - \frac{a_i}{A+B}\right) \\
&= \left(\frac{c_1^{1-\eta} A}{c_1 A + c_2 B} - \frac{c_1^{-\eta} A}{A+B}\right) + \left(\frac{c_2^{1-\eta} B}{c_1 A + c_2 B} - \frac{c_2^{-\eta} B}{A+B}\right) \\
&= \frac{c_1^{1-\eta} A + c_2^{1-\eta} B}{c_1 A + c_2 B} - \frac{c_1^{-\eta} A + c_2^{-\eta} B}{A+B} \\
&= \frac{(c_1^{1-\eta} A + c_2^{1-\eta} B)(A+B) - (c_1^{-\eta} A + c_2^{-\eta} B)(c_1 A + c_2 B)}{(c_1 A + c_2 B)(A+B)} \\
&= \frac{(c_1 - c_2)(c_1^{-\eta} - c_2^{-\eta}) AB}{(c_1 A + c_2 B)(A+B)} \\
&\leq 0,
\end{aligned}$$

and the inequality in the last line holds with equality iff either A or B are zero (note that $c_1, c_2, \eta > 0$), which is the case only if $\mathbf{x} = c_1 \boldsymbol{a}$ or $\mathbf{x} = c_2 \boldsymbol{a}$. The lemma then follows from the fact that $\mathbf{x}^*$ is the maximizer of f. This completes the proof of Lemma 1. □

We proceed with the proof of the theorem. Given the rewards $r(\cdot|\cdot)$ in the n-ary BT model (see Section 5), let

$$\pi^*(\cdot|\cdot) \stackrel{\text{def}}{=} \text{Softmax}\big(r(\cdot|\cdot)\big). \tag{36}$$

For any $(x; y_1, \ldots, y_n) \in \mathcal{X} \times \mathcal{Y}^n$, let

$$\bar{\mathcal{D}}(x; y_1, \ldots, y_n) \stackrel{\text{def}}{=} \frac{\sum_{i=1}^{n} \mathcal{D}(x; y_1, \ldots, y_n; i)}{\sum_{i=1}^{n} \pi^*(y_i|x)}. \tag{37}$$

It then follows from the consistency of $\mathcal{D}$ with the n-ary BT model that for any $(x; y_1, \ldots, y_n) \in \mathcal{X} \times \mathcal{Y}^n$ and $i = 1, \ldots, n$

$$\mathcal{D}(x; y_1, \ldots, y_n; i) = \bar{\mathcal{D}}(x; y_1, \ldots, y_n)\, \pi^*(y_i|x). \tag{38}$$

We further define

$$\tilde{\mathcal{D}}(x; [y]) \stackrel{\text{def}}{=} \bar{\mathcal{D}}(x; y_1, \ldots, y_n)\, \mu(x; y_1, \ldots, y_n). \tag{39}$$

For brevity of notation, we denote $y_1, \ldots, y_n$ by $[y]$ and denote $1, \ldots, n$ by $[n]$. The loss function $\mathcal{L}^{\alpha,\mu}_{\text{pref-}n}(\pi, \mathcal{D})$ defined in (14) can then be simplified to

$$\begin{aligned}
\mathcal{L}^{\alpha,\mu}_{\text{pref-}n}(\pi, \mathcal{D}) &= -\frac{1}{\alpha} \mathbb{E}_{(x; y_1, \ldots, y_n; i^*) \sim \mathcal{D}} \left[\mu(y_1, \ldots, y_n \mid x) \log \frac{\pi(y_{i^*} \mid x)^{\alpha}}{\sum_{i=1}^{n} \pi(y_i \mid x)^{\alpha}}\right] \\
&= -\frac{1}{\alpha} \sum_{(x; [y]; i^*) \in \mathcal{X} \times \mathcal{Y}^n \times [n]} \mathcal{D}(x; [y]; i^*)\, \mu([y]|x) \log \frac{\pi(y_{i^*} \mid x)^{\alpha}}{\sum_{i=1}^{n} \pi(y_i \mid x)^{\alpha}} \\
&= -\frac{1}{\alpha} \sum_{(x; [y]; i^*) \in \mathcal{X} \times \mathcal{Y}^n \times [n]} \tilde{D}(x; [y])\, \pi^*(y_{i^*}|x) \log \frac{\pi(y_{i^*} \mid x)^{\alpha}}{\sum_{i=1}^{n} \pi(y_i \mid x)^{\alpha}} \\
&= -\frac{1}{\alpha} \sum_{(x; [y]) \in \mathcal{X} \times \mathcal{Y}^n} \tilde{D}(x; [y]) \sum_{i=1}^{n} \pi^*(y_i|x) \log \frac{\pi(y_i \mid x)^{\alpha}}{\sum_{j=1}^{n} \pi(y_j \mid x)^{\alpha}},
\end{aligned}$$

where the third equality is due to (38) and (39).

In the rest of the proof, without loss of generality, we consider a single fixed $x \in \mathcal{X}$, and remove x from the notations for the sake of notation brevity. Let

$$\pi^*_\alpha(\cdot) \stackrel{\text{def}}{=} \text{Softmax}\big(r(\cdot)/\alpha\big). \tag{40}$$

It follows that for any $y \in \mathcal{Y}$,

$$\pi^*_\alpha(y) = \frac{\pi^*(y)^{1/\alpha}}{\sum_{\tilde{y}\in\mathcal{Y}} \pi^*(\tilde{y})^{1/\alpha}}. \tag{41}$$

Let $\boldsymbol{\pi}$ and $\boldsymbol{\pi}^*_\alpha$ be the vector representation of $\pi(y)$ and $\pi^*_\alpha(y)$ for all $y \in \mathcal{Y}$. Then, for $\mathbf{v} \stackrel{\text{def}}{=} -\nabla_\pi \mathcal{L}^{\alpha,\mu}_{\text{pref-}n}(\pi, \mathcal{D})$ we have

$$\begin{aligned}
\Big(\boldsymbol{\pi} - \boldsymbol{\pi}^{*1/\alpha}\Big)^T \ \mathbf{v} &= -\left(\boldsymbol{\pi} - \boldsymbol{\pi}^*_\alpha\right)^T \ \nabla_\pi \mathcal{L}^{\alpha,\mu}_{\text{pref-}n}(\pi, \mathcal{D}) \\
&= \frac{1}{\alpha} \sum_{[y]\in\mathcal{Y}^n} \tilde{D}([y]) \ \left(\boldsymbol{\pi} - \boldsymbol{\pi}^*_\alpha\right)^T \ \nabla_\pi \sum_{i=1}^n \pi^*(y_i) \ \log \ \frac{\pi(y_i \mid x)^\alpha}{\sum_{j=1}^n \pi(y_j \mid x)^\alpha} \\
&= \frac{1}{\alpha} \sum_{[y]\in\mathcal{Y}^n} \tilde{D}([y]) \sum_{\tilde{y}\in\mathcal{Y}} \left(\pi(\tilde{y}) - \pi^*_\alpha(\tilde{y})\right) \ \frac{d}{d\,\tilde{y}} \sum_{i=1}^n \pi^*(y_i) \ \log \ \frac{\pi(y_i \mid x)^\alpha}{\sum_{j=1}^n \pi(y_j \mid x)^\alpha} \\
&= \frac{1}{\alpha} \sum_{[y]\in\mathcal{Y}^n} \tilde{D}([y]) \sum_{k=1}^n \left(\pi(y_k) - \pi^*_\alpha(y_k)\right) \ \frac{d}{d\,y_k} \sum_{i=1}^n \pi^*(y_i) \ \log \ \frac{\pi(y_i \mid x)^\alpha}{\sum_{j=1}^n \pi(y_j \mid x)^\alpha}.
\end{aligned} \tag{42}$$

For any $[y] = (y_1, \ldots, \mathbf{y}_n) \in \mathcal{Y}^n$, let

$$A\big([y]\big) \stackrel{\text{def}}{=} \frac{1}{\alpha} \sum_{k=1}^n \left(\pi(y_k) - \pi^*_\alpha(y_k)\right) \ \frac{d}{d\,y_k} \sum_{i=1}^n \pi^*(y_i) \ \log \ \frac{\pi(y_i \mid x)^\alpha}{\sum_{j=1}^n \pi(y_j \mid x)^\alpha}. \tag{43}$$

It then follows from (42) that:

$$\mathbf{v}^T \left(\boldsymbol{\pi} - \boldsymbol{\pi}^{*1/\alpha}\right) = \sum_{[y]\in\mathcal{Y}^n} \tilde{D}([y]) \, A\big([y]\big). \tag{44}$$

We proceed to compute $A\big([y]\big)$. For $k = 1, \ldots, n$,

$$\begin{aligned}
\frac{d}{d\,y_k} \sum_{i=1}^n \frac{\pi(y_i \mid x)^\alpha}{\sum_{j=1}^n \pi(y_j \mid x)^\alpha} &= \frac{d}{d\,y_k} \sum_{i=1}^n \pi^*(y_i) \left(\log \ \pi(y_i)^\alpha \ - \ \log \sum_{j=1}^n \pi(y_j)^\alpha\right) \\
&= \alpha \frac{\pi^*(y_k)}{\pi(y_k)} \ - \ \left(\sum_{i=1}^n \pi^*(y_i)\right) \frac{d}{d\,y_k} \log \sum_{j=1}^n \pi(y_j)^\alpha \\
&= \alpha \frac{\pi^*(y_k)}{\pi(y_k)} \ - \ \alpha \left(\sum_{i=1}^n \pi^*(y_i)\right) \frac{\pi(y_k)^{\alpha-1}}{\sum_{j=1}^n \pi(y_j)^\alpha}
\end{aligned} \tag{45}$$

Plugging this into the definition of $A\big([y]\big)$ in (43), we obtain

$$\begin{aligned}
A\big([y]\big) &= \sum_{k=1}^{n} \left(\pi(y_k) - \pi^*_\alpha(y_k)\right) \left(\frac{\pi^*(y_k)}{\pi(y_k)} - \left(\sum_{i=1}^{n} \pi^*(y_i)\right) \frac{\pi(y_k)^{\alpha-1}}{\sum_{j=1}^{n} \pi(y_j)^\alpha}\right) \\
&= \sum_{k=1}^{n} \pi(y_k) \left(\frac{\pi^*(y_k)}{\pi(y_k)} - \left(\sum_{i=1}^{n} \pi^*(y_i)\right) \frac{\pi(y_k)^{\alpha-1}}{\sum_{j=1}^{n} \pi(y_j)^\alpha}\right) \\
&\quad + \sum_{k=1}^{n} \pi^*_\alpha(y_k) \left(\left(\sum_{i=1}^{n} \pi^*(y_i)\right) \frac{\pi(y_k)^{\alpha-1}}{\sum_{j=1}^{n} \pi(y_j)^\alpha} - \frac{\pi^*(y_k)}{\pi(y_k)}\right) \\
&= \sum_{k=1}^{n} \pi^*(y_k) - \left(\sum_{i=1}^{n} \pi^*(y_i)\right) \frac{\sum_{k=1}^{n} \pi(y_k)^\alpha}{\sum_{j=1}^{n} \pi(y_j)^\alpha} \\
&\quad + \sum_{k=1}^{n} \pi^*_\alpha(y_k) \left(\left(\sum_{i=1}^{n} \pi^*(y_i)\right) \frac{\pi(y_k)^{\alpha-1}}{\sum_{j=1}^{n} \pi(y_j)^\alpha} - \frac{\pi^*(y_k)}{\pi(y_k)}\right) \\
&= \sum_{k=1}^{n} \pi^*_\alpha(y_k) \left(\left(\sum_{i=1}^{n} \pi^*(y_i)\right) \frac{\pi(y_k)^{\alpha-1}}{\sum_{j=1}^{n} \pi(y_j)^\alpha} - \frac{\pi^*(y_k)}{\pi(y_k)}\right) \\
&= \frac{\sum_{i=1}^{n} \pi^*(y_i)}{\sum_{i=1}^{n} \pi^*(y_i)^{1/\alpha}} \sum_{k=1}^{n} \left(\frac{\pi^*(y_k)}{\pi(y_k)^\alpha}\right)^{1/\alpha} \left(\frac{\pi(y_k)^\alpha}{\sum_{j=1}^{n} \pi(y_j)^\alpha} - \frac{\pi^*(y_k)}{\sum_{i=1}^{n} \pi^*(y_i)}\right) \\
&\leq 0 \qquad (\texttt{"="}\ \texttt{only if}\ \boldsymbol{\pi}^* = c\boldsymbol{\pi}\ \texttt{for some scalar}\ c > 0),
\end{aligned} \tag{46}$$

where the last equality is due to (41), and the inequality in the last line follows from Lemma 1 by letting $a_k = \pi^*(y_k)$, $b_k = \pi(y_k)^\alpha$, and $\eta = 1/\alpha$. Plugging this into (44), it follows that

$$-\Big(\nabla_\pi \mathcal{L}^{\alpha,\mu}_{\text{pref-}n}(\pi, \mathcal{D})\Big)^T (\boldsymbol{\pi} - \boldsymbol{\pi}^*_\alpha) = \mathbf{v}^T(\boldsymbol{\pi} - \boldsymbol{\pi}^*_\alpha) < 0 \tag{47}$$

if $\boldsymbol{\pi} \neq \boldsymbol{\pi}^*_\alpha$. Consequently, $\|\boldsymbol{\pi} - \boldsymbol{\pi}^*_\alpha\|$ is strictly decreasing when moving along $\mathbf{v}$. Since both $\boldsymbol{\pi}$ and $\boldsymbol{\pi}^*_\alpha$ lie on the probability simplex, we have $\prod(\mathbf{v})^T(\boldsymbol{\pi} - \boldsymbol{\pi}^*_\alpha) \leq \mathbf{v}^T(\boldsymbol{\pi} - \boldsymbol{\pi}^*_\alpha) < 0$. It follows that for any $\boldsymbol{\pi}$ in the relative interior of the probability simplex, projection of $\mathbf{v}$ on the probability simplex is a strictly decent direction for $\|\boldsymbol{\pi} - \boldsymbol{\pi}^*_\alpha\|$. As a result, $\boldsymbol{\pi}^*_\alpha$ is the globally absorbing unique fixed point of the ODE. This completes the proof of Theorem 3.

D Proof of Theorem 4

Here we present the proof of Theorem 4. The high level idea is to show that $\mathcal{L}^{\alpha,[\mu]}_{\text{rank}}(\pi, \mathcal{D})$ can be equivalently written as the sum of $\mathcal{L}^{\alpha,\mu_k}_{\text{pref-}n}(\pi, \mathcal{D}_k)$ for appropriately defined $\mathcal{D}_k$, for $k = 1, \ldots, n-1$; where each $\mathcal{D}_k$ is consistent with the $(n-k+1)$-ary BT model (defined in Section 5). We then use Theorem 3, and in particular (47) in the proof of Theorem 3, to conclude that the softmax distribution is a globally absorbing fixed point of $-\nabla\mathcal{L}^{\alpha,\mu_k}_{\text{pref-}n}(\pi, \mathcal{D}_k)$ for $k = 1, \ldots, n-1$, and is therefore a globally absorbing fixed point of their sum, $-\nabla\mathcal{L}^{\alpha,[\mu]}_{\text{rank}}(\pi, \mathcal{D})$.

As in the previous appendices, without loss of generality we prove the theorem for a single fixed $x \in \mathcal{X}$, and remove x from the equations for notation brevity. To further simplify the notation, without loss of generality, we also remove the permutation τ from the equations, and represent the ranking by mere order of the indices, that is we assume that $y_1 \succ y_2 \succ \cdots \succ y_n$. With these new conventions, the ranking loss (15) simplifies to

$$\mathcal{L}^{\alpha,[\mu]}_{\text{rank}}(\pi, \mathcal{D}) \stackrel{\text{def}}{=} -\frac{1}{\alpha} \mathbb{E}_{(y_1,\ldots,y_n)\sim\mathcal{D}} \left[\sum_{k=1}^{n-1} \mu_k(y_k, \ldots, y_n) \log \frac{\pi(y_k)^\alpha}{\sum_{j=k}^{n} \pi(y_j)^\alpha}\right]. \tag{48}$$

For $k = 1, \ldots, n-1$, we define an $(n-k+1)$-ary preference distribution $\mathcal{D}_k$ as follows. For any $(y_1, \ldots, y_{n-k+1}) \in \mathcal{Y}^n$ and $i = 1, \ldots, n-k+1$,

$$\mathcal{D}_k(y_1, \ldots, y_{n-k+1}; i) = \frac{1}{(n-k)!} \sum_{\substack{(z_1,\ldots,z_{k-1}) \in \mathcal{Y}^{k-1} \\ \text{Permutation } \tau:(1,\ldots,n-k) \to (1,\ldots,\cancel{i},\ldots,n-k+1)}} \mathcal{D}(z_1, \ldots, z_{k-1}, y_i, y_{\tau(1)}, \ldots, y_{\tau(n-k)}). \tag{49}$$

From (48), we have

$$\begin{aligned}
\mathcal{L}_{\text{rank}}^{\alpha,[\mu]}(\pi, \mathcal{D}) &= -\frac{1}{\alpha} \mathbb{E}_{(y_1,\ldots,y_n)\sim\mathcal{D}} \left[\sum_{k=1}^{n-1} \mu_k(y_k, \ldots, y_n) \log \frac{\pi(y_k)^\alpha}{\sum_{j=k}^n \pi(y_j)^\alpha} \right] \\
&= -\frac{1}{\alpha} \sum_{k=1}^{n-1} \mathbb{E}_{(y_1,\ldots,y_n)\sim\mathcal{D}} \left[\mu_k(y_k, \ldots, y_n) \log \frac{\pi(y_k)^\alpha}{\sum_{j=k}^n \pi(y_j)^\alpha} \right] \\
&= -\frac{1}{\alpha} \sum_{k=1}^{n-1} \sum_{(y_1,\ldots,y_n)\in\mathcal{Y}^n} \mathcal{D}(y_1, \ldots, y_n) \left[\mu_k(y_k, \ldots, y_n) \log \frac{\pi(y_k)^\alpha}{\sum_{j=k}^n \pi(y_j)^\alpha} \right] \\
&= -\frac{1}{\alpha} \sum_{k=1}^{n-1} \sum_{y_k,\ldots,y_n} \sum_{(y_1,\ldots,y_{k-1})\in\mathcal{Y}^{k-1}} \mathcal{D}(y_1, \ldots, y_n) \left[\mu_k(y_k, \ldots, y_n) \log \frac{\pi(y_k)^\alpha}{\sum_{j=k}^n \pi(y_j)^\alpha} \right] \\
&= -\frac{1}{\alpha} \sum_{k=1}^{n-1} \sum_{y_k,\ldots,y_n} \frac{\mathcal{D}_k(y_1, \ldots, y_{n-k+1}; 1)}{\big((n-k)!\big)^2} \left[\mu_k(y_k, \ldots, y_n) \log \frac{\pi(y_k)^\alpha}{\sum_{j=k}^n \pi(y_j)^\alpha} \right] \\
&= -\frac{1}{\alpha} \sum_{k=1}^{n-1} \frac{1}{(n-k+1)!\,(n-k)!} \mathbb{E}_{(y_k,\ldots,y_n;i)\sim\mathcal{D}_k} \left[\mu_k(y_k, \ldots, y_n) \log \frac{\pi(y_k)^\alpha}{\sum_{j=k}^n \pi(y_j)^\alpha} \right] \\
&= \sum_{k=1}^{n-1} \frac{\mathcal{L}_{\text{pref-}n}^{\alpha,\mu_k}(\pi, \mathcal{D}_k)}{(n-k+1)!\,(n-k)!}.
\end{aligned}$$

Let $\boldsymbol{\pi}$ and $\boldsymbol{\pi}_\alpha^*$ be the vector representations of π and the softmax distribution π_α^* (defined in (41)), and $\mathbf{v} \stackrel{\text{def}}{=} -\nabla_\pi \mathcal{L}_{\text{rank}}^{\alpha,[\mu]}(\pi, \mathcal{D})$. Then,

$$\begin{aligned}
(\boldsymbol{\pi} - \boldsymbol{\pi}_\alpha^*)^T \mathbf{v} &= -(\boldsymbol{\pi} - \boldsymbol{\pi}_\alpha^*)^T \nabla \sum_{k=1}^{n-1} \frac{\mathcal{L}_{\text{pref-}n}^{\alpha,\mu_k}(\pi, \mathcal{D}_k)}{(n-k+1)!\,(n-k)!} \\
&= \sum_{k=1}^{n-1} \frac{-(\boldsymbol{\pi} - \boldsymbol{\pi}_\alpha^*)^T \nabla \mathcal{L}_{\text{pref-}n}^{\alpha,\mu_k}(\pi, \mathcal{D}_k)}{(n-k+1)!\,(n-k)!} \\
&\le 0,
\end{aligned}$$

where the last inequality follows from (47), and it holds with equality only if $\boldsymbol{\pi} \neq \boldsymbol{\pi}_\alpha^*$. Since both $\boldsymbol{\pi}$ and $\boldsymbol{\pi}_\alpha^*$ lie on the probability simplex, we have $\prod(\mathbf{v})^T(\boldsymbol{\pi} - \boldsymbol{\pi}_\alpha^*) \le \mathbf{v}^T(\boldsymbol{\pi} - \boldsymbol{\pi}_\alpha^*) < 0$. Then, following a similar argumnet as in the last paragraph of Appendix C, we conclude that $\boldsymbol{\pi}_\alpha^*$ is the globally absorbing unique fixed point of the ODE. This completes the proof of Theorem 4.

E Experiment Details

E.1 Details for the AlpacaFarm experiment of Section 7.1

We performed the alignment procedure on 4 NVIDIA H100 (94 GiB) GPUs. For each method, we trained the model for four epochs and reported the maximum win-rate against the SFT model. The training batch size per GPU was set to 1, with gradient accumulation over 16 steps. For each model update, corresponding

to 16 dataset samples, we sampled a batch of 8 responses from the model π_θ to compute the $\mathcal{D}_{\text{KL}}$ regularizer. The range of hyper-parameters considered for each method is given as follows: R-DPO: $\beta \in \{0.01, 0.1\}$, $\alpha \in \{0.001, 0.01, 0.1\}$, CPO: $\beta \in \{0.001, 0.01, 0.1\}$, $\alpha \in \{0.0001, 0.001, 0.01\}$, SimPO: $\beta \in \{2, 2.5\}$, $\gamma \in \{1, 1.5\}$ (the suggested range in SimPO), KTO: $\beta \in \{0.01, 0.1\}$, $\lambda_D = \lambda_U = 1$, IPO: $\tau \in \{0.001, 0.01, 0.1\}$, DPO: $\beta \in \{0.0001, 0.001, 0.01, 0.1\}$, and SPO: $\alpha \in \{0.001, 0.01\}$, $\beta \in \{0.001, 0.01\}$. We additionally verified that reducing DPO to $\beta = 0.0001$ does not improve the reported comparisons: on LLaMA2-7B it reduced win-rate (57.65% vs. 59.16% at best), and on LLaMA3-8B it had negligible effect (15.40% vs. 15.38%).

Table 8 shows the sensitivity of SPO to its hyperparameters.

	$\alpha = 0.01$	$\alpha = 0.001$
$\beta = 0.01$	**60.83**	59.82
$\beta = 0.001$	57.15	59.99

Table 8: SPO win-rate for different α and β values.

E.2 Details for the experiments of Sections 7.2 and 7.3

We performed the alignment procedure on 4 NVIDIA H100 (94 GiB) GPUs. For each method, we trained the model for two epochs and reported the win-rate on UltraFeedback dataset using ApalacaEval 2.0 (where the responses are compared with the ones of gpt4_turbo (the default reference model)). The training batch size per GPU device was set to two and the gradient accumulation step was 16. For each model update, corresponding to 16 dataset samples, we sampled a batch of 4 responses from the model π_θ to compute the $\mathcal{D}_{\text{KL}}$ regularizer. The range of hyper-parameters considered for each method is given as follows: SimPO: $\beta \in \{2, 2.5\}$, $\gamma \in \{1, 1.5\}$, DPO: $\beta \in \{0.001, 0.01, 0.1\}$, and SPO: $\alpha \in \{0.001, 0.01\}$, $\beta \in \{0.001, 0.01\}$.

E.3 Details for the experiment of Section 7.4

We trained the models on NVIDIA A100 (40 GiB) GPUs. We used a batch size of 32 samples (each containing four responses) for all algorithms. The reference model in all algorithms was identical to the SFT model. All alignment loss functions were optimized using AdamW with 5,000 warm-up iterations.

For SPO, we trained both basic (i.e., $\gamma = 0$) and weighted version. For weighted SPO, we set $\gamma = 0.01$ without sweeping, and in the ranking experiment we used decayed weight functions $\eta^k \mu_k$ for $\eta \in 1, 0.5$, see (15). For other SPO parameters, we swept over $\beta \in \{0.01, 0.1\}$, and $\alpha \in \{0.001\}$. For $\mathcal{D}_{\text{KL}}$ computation, we used intermittent batch generation of samples, generating a batch of 32 samples from π_θ once every 8 iterations (i.e., $T = 8$). For other algorithms, we swept over the following sets of hyperparameters: DPO: $\beta \in \{0.0001, 0.001, 0.01\}$, S-DPO for ranking: $\beta \in \{0.0001, 0.001, 0.01\}$, and S-DPO for best-of-$n$: $\beta \in \{0.0001, 0.001, 0.01\}$. For For training the S-DPO algorithm on the best-of-n dataset, we consider each top-rank response as the positive response and the corresponding lower-rank responses as the corresponding negative set of responses. For training S-DPO on ranking dataset, each of the 1st, 2nd, and 3rd rank responses serve as positive responses with the corresponding negative set containing the corresponding lower rank responses.

We computed the win rates of all methods against the best-of-n SFT model using GPT4o-2024-08-06 once every 1000 iterations, and reported the peak win-rate for each method. Each win-rate was averaged over 1,000 story-pair instances, resulting in an estimation error with standard deviation smaller than 0.015.

F Details of Generation of Ranking Dataset for the TinyStories Experiment

We created a preference dataset to align stories with older age groups. Specifically, for each pair of stories generated by the reference model, we asked GPT4o–2024-08-06 to evaluate them based on clarity and coherence, writing quality, and whether they are interesting and engaging for high school students. The API was asked to evaluate each story independently based on these criteria, and identify its strengths and

weaknesses compared to other stories; and then suggest a ranking of stories from best to worst. The prompt used for generating the dataset is provided at the end of this subsection.

We generated a set of 100,000 stories independently from the 110M-parameter pre-trained model (Karpathy, 2023), and grouped them into a set of 25,000 samples each containing four stories. To enhance the quality of the ranking dataset, for each sample we used the prompt to rank the stories twice, reversing the order of the stories in the second evaluation. We retained samples only if both evaluations showed a consistent ranking. After this filtration, 5,000 samples remained for use in the ranking dataset. The dataset is available online at (Anonymous, 2024).

Prompt for Generation of the Ranking Dataset for TinyStories Experiment:

```
You are tasked with deciding which of the four short stories below, written by high school
students, is better suited for publication in the high school newspaper.

**Story 1:**   {}

**Story 2:**   {}

**Story 3:**   {}

**Story 4:**   {}

**Your Task:**
1.  **Evaluate Each Story Individually:**
 Identify the **strengths** and **weaknesses** of each story, focusing on:
   - **Engagement:** Is the story interesting and likely to captivate high school students?
   - **Clarity and Coherence:** Is the story well-organized and easy to follow?
   - **Writing Quality:** Assess the grammar, vocabulary, and overall language use.

2.  **Make a Final Decision:**
   - Based on your evaluations, decide which story is better suited for publication.  Rank the
stories from best to worst.

**Response Format:**
***
**Evaluation:**
**Story 1:**
- **Strengths compared to other stories:**
   - [List strengths]
- **Weaknesses compared to other stories:**
   - [List weaknesses]
--
**Story 2:**
- **Strengths compared to other stories:**
   - [List strengths]
- **Weaknesses compared to other stories:**
   - [List weaknesses]
--
**Story 3:**
- **Strengths compared to other stories:**
   - [List strengths]
- **Weaknesses compared to other stories:**
   - [List weaknesses]
--
**Story 4:**
- **Strengths compared to other stories:**
   - [List strengths]
- **Weaknesses compared to other stories:**
   - [List weaknesses]
--
**Conclusion:**
- **Overall Ranking of the Stories:**
   1.  **1st Place:** Story [1, 2, 3, or 4]
   2.  **2nd Place:** Story [1, 2, 3, or 4]
   3.  **3rd Place:** Story [1, 2, 3, or 4]
   4.  **4th Place:** Story [1, 2, 3, or 4]
***

**Guidelines:**
   - In each evaluation, compare the story to the others, noting unique strengths and weaknesses.
   - Evaluation of each story shold not be influenced from prior evaluations that you provided
earlier.
   - Do not let the presentation order affect your judgment; treat all stories equally.
```