Group Robust Preference Optimization
in Reward-free RLHF

Shyam Sundhar Ramesh[1,6]   Yifan Hu[3,4]   Iason Chaimalas[1]   Viraj Mehta[2]

Pier Giuseppe Sessa[3]   Haitham Bou Ammar[1,5]   Ilija Bogunovic[1]

May 31, 2024

Abstract

Adapting large language models (LLMs) for specific tasks usually involves fine-tuning through reinforcement learning with human feedback (RLHF) on preference data. While these data often come from diverse labelers' groups (e.g., different demographics, ethnicities, company teams, etc.), traditional RLHF approaches adopt a "one-size-fits-all" approach, i.e., they indiscriminately assume and optimize a single preference model, thus not being robust to unique characteristics and needs of the various groups. To address this limitation, we propose a novel Group Robust Preference Optimization (GRPO) method to align LLMs to individual groups' preferences robustly. Our approach builds upon reward-free direct preference optimization methods, but unlike previous approaches, it seeks a robust policy which maximizes the worst-case group performance. To achieve this, GRPO adaptively and sequentially weights the importance of different groups, prioritizing groups with worse cumulative loss. We theoretically study the feasibility of GRPO and analyze its convergence for the log-linear policy class. By fine-tuning LLMs with GRPO using diverse group-based global opinion data, we significantly improved performance for the worst-performing groups, reduced loss imbalances across groups, and improved probability accuracies compared to non-robust baselines.

1   Introduction

As the usage of large language models (LLMs) has grown in recent years, the question of their *alignment* has come to the forefront. Their remarkable capability to address a wide range of tasks (Radford et al. [34]) stems from pre-training on a self-supervised objective over internet-scale text. This vast internet-scale content, however, carries a higher risk of biases, inaccuracies, and controversial content than smaller, curated datasets. Thus, ensuring that the model's responses and behaviors correspond to human intentions and values is crucial.

Typical approaches to alignment [10, 32, 35] involve gathering preference feedback from human labelers to train models that reflect their desires. Such approaches often treat individual preferences as samples from a broader preference distribution. However, this perspective often oversimplifies the complex reality that human societies consist of numerous *distinct groups* (e.g., different demographics, ethnicities, company teams, etc.), each with their own set of preferences that can significantly diverge. Consequently, prevalent

---

[1]University College London

[2]TensorZero

[3]ETH Zurich

[4]EPFL

[5]Huawei Noah's Ark Lab
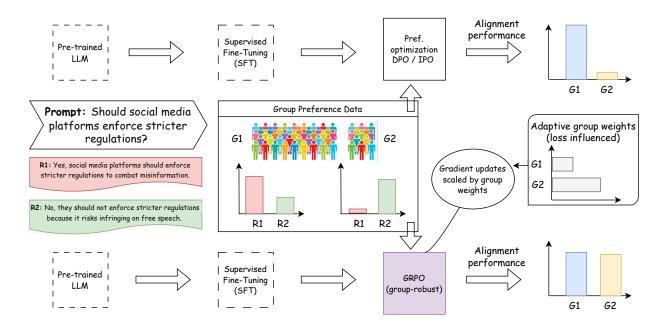
[6]Corresponding author: `shyam.ramesh.22@ucl.ac.uk`

Figure 1: Current reward-free preference optimization methods typically optimize based on average human feedback. This often aligns predominantly with the preferences of the majority group (G1, R1 > R2) at the expense of minority groups (G2, R2 > R1). In contrast, our GRPO algorithm introduces adaptive weighting for different user groups and prioritizes optimizing for the worst-case group performance, leading to better alignment for the most disadvantaged groups.

alignment strategies tend to adopt a "one-size-fits-all" model and disproportionately favor the preferences of the majority group, often at the expense of minority groups and their preferences, as illustrated in Figure 1.

To improve alignment performance for even the most disadvantaged groups, we propose to robustly solve the problem of diverse group preferences by (i) including group information in the context of the LLM and (ii) optimizing against the *worst-case* alignment performance across all groups. We develop policies that guarantee equitable performance across all groups, ensuring that no group is disproportionately disadvantaged due to inherent biases or imbalances in the training data.

Related work. The established process for alignment of LLMs using Reinforcement Learning from Human Feedback (RLHF) is set out in [42, 57] and [32]. The RLHF fine-tuning process consists of learning a reward model from human comparisons between responses to a given prompt, using the Bradley-Terry model [4]. Then, one performs policy optimization using Proximal Policy Optimization [38] to learn a policy that maximizes the learned reward function. For a comprehensive overview and perspective of the RLHF topic, we refer the reader to [22, 6, 21].

Due to the challenges of tuning PPO and the vulnerability of reward models ([47, 35, 15, 46]), alternative approaches to PPO-based RLHF have been proposed, including rejection sampling fine-tuning [12, 16, 46, 28] and conditional supervised fine-tuning [18, 51, 8]. In particular, Rafailov et al. [35] introduce Direct Preference Optimization (DPO), which optimizes policies directly based on human preferences, avoiding the need for a separate reward model. This approach simplifies training and reduces reward overfitting. Other studies, such as [1, 55, 44, 41, 14], propose novel *reward-free* RLHF methods, with some bypassing preference datasets altogether ([14, 5]). We utilize a reward-free framework similar to [35, 1], however, unlike previous works that assume a single preference distribution, we consider multiple preference distributions from diverse groups. Further, we aim to robustly fine-tune the LLM to ensure minimal disparity in performance across all groups. Other studies addressing robustness in preference optimization include [19]

and [24]. However, these works primarily focus on different aspects of robustness, such as robustness to noise and resilience against out-of-preference data.

Robust language modeling techniques have been studied by [31, 49] to optimize performance of language models over a wide-range of topics. They consider robust pre-training of language models based on the group Distributionally Robust Optimization (DRO) approach. A concrete theoretical study of the group DRO approach was performed in [37] and applied to vision problems. These are designed by extending previous minimax algorithm for solving DRO from [29]. In the RLHF setup, [3] consider weighting of loss from different topics (harmless vs helpful) for robust reward learning. Also, Chakraborty et al. [7] consider robust policy optimization by learning multiple reward functions corresponding to sub-populations and learning a robust policy w.r.t. the learned rewards. Differing from these works, we embed group robustness directly into the reward-free tuning paradigm. We provide an concrete algorithm that adaptively weighs the loss for different groups and optimizes for a policy that minimizes the weighted loss. Further, our algorithm employs a novel gradient estimator tailored to the group robust DPO problem.

In the non-robust setup, [54] also explores group-based preference learning with LLMs by including group information in prompts with a transformer module that is trained optimally choose an example sequence of prompts, LLM responses, and group preferences for in-context learning. We detail other related works extensively in Appendix A.

Main Contributions. The following are the main contributions of this work: (i) We present GRPO, the group robust formulation of Direct Preference Optimization (DPO) [35], wherein we augment the context of the LLM with the group information, and pose the problem as a robust optimization problem to minimize the worst-case loss amongst the diverse groups. To the best of our knowledge, this is the first study to focus on group robustness in RLHF preference optimization; (ii) We analyze the theoretical aspects of GRPO by examining the convergence and the feasibility of finding optimal solutions within the log-linear policy class; (iii) We present a tailored algorithm to tackle this robust optimization challenge, providing convergence guarantees for certain loss functions; (iv) We show the versatility of our approach by demonstrating how our algorithm can be utilized with other reward-free preference optimization methods such as Identity Preference Optimization (IPO) [1]. (v) Our empirical evaluations across synthetic datasets, real-world data, and publicly available LLMs show that the proposed GRPO significantly improves performance for the worst-performing groups, reduces loss imbalances across groups, and increases probability accuracies compared to non-robust baselines.

2   Background

We address the challenge of fine tuning a large language model (LLM) to align with user preferences. This process usually follows the Reinforcement Learning from Human Feedback (RLHF) protocol, using either an explicit reward model ([3, 32, 42, 57]) or an implicit reward model ([35]). RLHF typically comprises three key phases: (i) supervised fine-tuning of an initial (pre-trained) large-language model, (ii) reward learning, and (iii) RL fine-tuning.

In the *supervised fine-tuning phase* (SFT), the goal is to fine-tune a pre-trained LLM on a specific high-quality dataset suited for the downstream task of interest. It results in a probabilistic model expressing the probability of the response y given a prompt x as $\pi_{\text{ref}}(y|x)$. Subsequently, in the *reward learning phase*, the goal is to learn a reward model from a dataset of prompts x and responses y_w, y_l, with $y_l \prec y_w \mid x$ meaning that human labellers preferred y_w over y_l.

It is typically assumed that preferences follow some choice models with an *unknown* reward (utility) $r^*(x, y)$ function. A popular model is the Bradley-Terry model [4] that assumes the preference distribution

p admits the following form :

$$p(y_1 \prec y_2 \mid x) = \frac{\exp(r^*(x, y_2))}{\exp(r^*(x, y_1)) + \exp(r^*(x, y_2))}. \tag{1}$$

Based on the above model, a maximum likelihood estimate of the reward function is obtained as:

$$\min_r \{\mathcal{L}_R(r; \mathcal{D}) := -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \left(\sigma \left(r(x, y_w) - r(x, y_l) \right) \right) \right] \}, \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid function and $\mathcal{D}$ represents the dataset consisting of $\{(x, y_w, y_l)\}$.

Then, in the RL fine-tuning phase the objective is to train a policy π that maximizes the learned reward function. Simultaneously, the policy should stay closely aligned with the reference, π_{ref}, as quantified by the KL divergence, leading to the following KL-regularized optimization problem:

$$\max_\pi \ \mathbb{E}_{x \sim \mathcal{P}_x} \left[\mathbb{E}_{y \sim \pi} \left[r(x, y) \right] - \beta \text{KL} \left[\pi(y|x) \| \pi_{\text{ref}}(y|x) \right] \right]. \tag{3}$$

Direct Preference Optimization (DPO). The recent approach proposed by [35] exploits the closed-form solution of the problem in Equation (3) and sidesteps the explicit modelling of rewards to directly optimize the policy. Specifically, under the Bradley-Terry preference model, the reward function can be expressed directly in terms of the optimal policy π^* as follows:

$$r(x, y) = \beta \log \frac{\pi^*(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x), \tag{4}$$

for a partition function $Z(x) = \sum_y \pi_{\text{ref}}(y|x) \exp(\frac{1}{\beta} r(x, y))$. Via a change of variable, finding the optimal reward function in Equation (2) is equivalent to finding the optimal policy π^* utilizing the given set of preference data $\mathcal{D}$. With a slight abuse of notation, we use π to denote π^*. Denote $h_\pi(x, y_w, y_l) := \log(\frac{\pi(y_w|x)}{\pi_{\text{ref}}(y_w|x)}) - \log(\frac{\pi(y_l|x)}{\pi_{\text{ref}}(y_l|x)})$. Then, Equation (2) translates into the DPO loss:

$$\mathcal{L}_{\text{DPO}}(\pi, \mathcal{D}) = - \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \left(\sigma(\beta \cdot h_\pi(x, y_w, y_l)) \right) \right]. \tag{5}$$

With a parameterized policy model π_θ, minimizing the DPO loss involves calculating the gradient over θ using backpropagation and the log-probabilities of each completion, y_w and y_l, given the prompt x for both the policy π_θ and the reference policy π_{ref}.

3 Group Robust Preference Optimization

In this section, we discuss Group Robust Preference Optimization (GRPO), i.e., instead of learning a reward function that maximizes the likelihood, we aim to derive (implicitly) a robust reward function and subsequently learn a robust DPO policy.

Group Preferences. Suppose that preferences come from an underlying latent reward $r^*(x, y, g)$, with $g \in \mathcal{G} = \{1, 2, \ldots, K\}$ indexing the groups. When group information is available (e.g., as a text), we can represent the reward as $r^*(x_g, y)$, where $x_g = x \oplus g$ denotes merging[1] of the prompt with group information (e.g., string concatenation). We continue to apply a Bradley-Terry model as described in Equation (1), substituting x with x_g. Moreover, we assume access to a collective dataset $\mathcal{D} = \bigcup_{g=1}^K \mathcal{D}_g$ where $\mathcal{D}_g = \{(x_g^{(i)}, y_w^{(i)}, y_l^{(i)})\}_{i=1}^{N_g}$ with the available group information. Additionally, our dataset accommodates

---

[1] The group information added to the prompt can represent, e.g., an index or a textual description of the group.

the exposure of different groups to identical prompts, meaning that the same x can appear across various groups g in our dataset, and these groups may favor different responses y.

Given such $\mathcal{D}$, although one may obtain a common reward model using Equation (2), it could result in poor generalization for particular groups, especially with significant group-wise disparities in the data (see Figure 1). Such disparities might stem from imbalanced data across groups or difficulties associated with learning different groups.

GRPO Objective. Consequently, we propose to measure the alignment of the reward model on the *worst-case group* loss:

$$\max_{g \in G} \ \mathcal{L}_R(r; \mathcal{D}_g). \tag{6}$$

Incorporating the reward expression from (Equation (4)) into (Equation (6)), we establish the *group robust preference optimization* (GRPO) objective for a specified policy π:

$$\mathcal{L}_{\mathrm{GR}}(\pi) := \max_{g \in \mathcal{G}} \mathcal{L}_{\mathrm{DPO}}(\pi, \mathcal{D}_g) = \max_{g \in \mathcal{G}} \Big(-\mathbb{E}_{(x_g, y_w, y_l) \sim \mathcal{D}_g} \Big[\log \Big(\sigma \big(\beta h_\pi(x_g, y_w, y_l) \big) \Big) \Big] \Big). \tag{7}$$

Leveraging the equivalent formulation of maximizing over discrete set, the GRPO problem becomes

$$\min_\pi \mathcal{L}_{\mathrm{GR}}(\pi) = \min_\pi \max_{\alpha \in \Delta_K} \sum_{g=1}^{K} \alpha_g \Big(-\mathbb{E}_{(x_g, y_w, y_l) \sim \mathcal{D}_g} \Big[\log \Big(\sigma(\beta h_\pi(x_g, y_w, y_l)) \Big) \Big] \Big), \tag{8}$$

where Δ_K represents the $(K-1)$-dimensional simplex of probabilities.[1] The inner maximization becomes a linear programming over simplex such that α represents the *weights* of groups. In addition, it forms a two-player zero-sum game (see Section 3.1), where the policy π and α act as opponents with inversely related payoffs. The DPO loss (logistic log loss) in Equation (7) can be replaced with alternatives like hinge or squared loss (see [44]). We label this objective GR-DPO when using DPO loss, and explore GRPO with squared loss in Section 4.1.

Applications. In this study, we do not assume any specific distribution for groups $\mathcal{D}_g$. The collection of prompts per group, $\mathcal{P}_{x_g}$, may have varying degrees of overlap. The GRPO framework accommodates both distinct and overlapping prompt scenarios across different groups.

Apart from human groups, GRPO can be useful in scenarios where groups $\mathcal{D}_g$ represent distinct tasks or topics within preference datasets, like helpful/harmful, truthful/unanswerable instances, or domain-specific categories (e.g., math, physics, chemistry). Typically, these prompt distributions $\{\mathcal{P}_{x_g}\}_{g=1}^N$ are disjoint, and GRPO seeks to optimize performance even across the most challenging categories.

GRPO is also applicable in scenarios where groups reflect diverse user preferences for a *shared* set of prompts, with the goal of achieving equitable performance across user groups. This contrasts with non-robust DPO, which aims to optimize preferences on average and might overlook minority groups.

Lastly, we acknowledge that the max-min objective of Equation (8) might be overly conservative, potentially degrading average performance. We explore a more balanced approach between worst-case and standard preference optimization objective in Appendix B.4.

3.1 Further Discussion and Insights

This section provides two insights regarding the GR-DPO loss in Equation (8).

Log-linear policy class. The zero-sum game perspective allows us to explore the presence of a Nash equilibrium, serving as a benchmark for convergence during the policy optimization process. Given that the domain of α is a simplex Δ_K (in Equation (8)), we further define a parameterized policy class Π_θ for the

---

[1]From a distributionally-robust viewpoint, this is equivalent to defining the uncertainty set as $\big\{ \sum_{g=1}^{K} \alpha_g \mathcal{D}_g : \alpha \in \Delta_K \big\}$, and minimizing the worst-case expected loss across the uncertainty set.

policy π_θ. We assume that the parameterized policy π_θ is of the form $\pi_\theta(y \mid x) = \frac{\exp f_\theta(x,y)}{\sum_{y \in \mathcal{Y}} \exp f_\theta(x,y)}$, where f_θ is a linear function or a neural network, and θ belongs to a convex set Θ.

In LLM fine-tuning, sometimes practitioners concentrate on modifying solely the final layer. It corresponds to a linear function, $f_\theta(x,y) = \theta^T \phi(x,y)$, with $\phi(x,y)$ denoting the embedding derived from the language model removing its last layer, and θ as the parameters of the last layer. When applying this linear parameterization, in conjunction with a uniform reference policy π_{ref}, the robust objective outlined in Equation (8) is as follows (details in Appendix B.1):

$$\min_{\theta \in \Theta} \max_{\alpha \in \Delta_K} \sum_{g=1}^{K} \alpha_g \Big(- \mathbb{E}_{(x_g, y_w, y_l) \sim \mathcal{D}_g} \Big[\log \Big(\sigma \big(\beta \langle \phi(x, y_w) - \phi(x, y_l), \theta \rangle \big) \Big) \Big] \Big). \tag{9}$$

The objective defined in Equation (9) is concave with respect to α and convex with respect to θ. This structure allows the invocation of the minimax theorem for convex-concave functions ([40]) to assert the existence of a Nash equilibrium.

Proposition 1. *Under log-linear parameterization of the policy class, there exists a Nash equilibrium for the group robust direct preference optimization problem in Equation* (9).

Robust policy optimization. The earlier derivation for the GR-DPO objective $\mathcal{L}_{\text{GR-DPO}}(\pi)$ relies on incorporating robustness in the reward modeling step (in Equation (7)) while using the solution to the non-robust KL-regularized reward maximization objective in Equation (4).

Interestingly, we can obtain the identical expression for $\mathcal{L}_{\text{GR-DPO}}(\pi)$ if incorporating robustness in the KL-regularized reward maximization objective and using the reward function learnt in a non-robust way. Consider the robust KL-regularized reward maximization

$$\max_{\pi} \min_{g \in \mathcal{G}} \mathbb{E}_{x_g \sim \mathcal{P}_{x_g}, y \sim \pi(\cdot \mid x_g)} \Big[r(x_g, y) - \beta \text{KL} \big[\pi(y \mid x_g) || \pi_{\text{ref}}(y \mid x_g) \big] \Big]. \tag{10}$$

The following proposition characterizes such an invariant property.

Proposition 2. *Substituting the closed-form solution of the robust KL-regularized policy maximization problem (Equation* (10)*) into the non-robust KL-regularized reward maximization objective in Equation* (2) *leads to the same group robust DPO loss $\mathcal{L}_{\text{GR-DPO}}$ in Equation* (8) *.*

The analysis leverages the fact that the optimal policy of Equation (10) is identical to the solution of the non-robust KL-regularized reward maximization in Equation (4) and is derived in Appendix B.2.

4 Algorithm

In this section, we discuss the policy optimization algorithm for solving the group robust DPO problem in Equation (8). In particular, we aim to design an algorithm that performs updates in the parameterized space $\Theta \subset \mathbb{R}^d$, i.e., updating θ of the parameterized policy π_θ. Leveraging the perspective of the 2-player zero-sum game, we propose an alternating updating algorithm wherein one updates α and θ alternatively. We summarize the overall approach in Algorithm 1, which we discuss and analyze next.

We employ the DPO loss $l(\pi_\theta; \cdot) = \log(\sigma(\beta h_{\pi_\theta}(\cdot)))$ (Equation (5)) in Algorithm 1, however, our algorithm can support other preference optimization losses (see Section 4.1). The algorithm performs a gradient descent type update on θ and a deterministic mirror ascent on α using a Bregman divergence with the distance generating function as the KL divergence. Since the α lies in a simplex and the objective is linear, the update of α becomes multiplicative weights update with renormalization to a simplex via softmax (see Nemirovski et al. [30] for details). Further, the weights α are determined by the cumulative losses

---

Algorithm 1 Mirror Descent for Group Robust Preference Optimization (GRPO)

1: **Initialize:** Step size η_α for group weights α, step size η_θ for policy π with weights θ, initial weights $\theta^{(0)}$ of the policy and weights over each group $\alpha^{(0)}$, Projection operator P_Θ

2: **Input:** Dataset $\mathcal{D}$ with size $N = |\mathcal{D}|$, group size N_g for $g = \{1, 2, \cdots, K\}$, loss $l(\pi_\theta; \cdot)$

3: **for** $t = 1, \ldots, T$ **do**

4: $\quad \alpha' \leftarrow \alpha^{(t-1)}$

5: $\quad g \sim \mathrm{Categorical}(N_1/N, \cdots, N_K/N), (x_g, y_w, y_l) \sim \mathcal{D}_g$

6: $\quad \alpha'_g \leftarrow \alpha'_g \exp \eta_\alpha \left(\frac{N \cdot l(\pi_{\theta^{(t-1)}}; (x_g, y_w, y_l))}{N_g} \right)$ // Update weights for group g

7: $\quad \alpha^{(t)} \leftarrow \alpha' / \sum_{g'} \alpha'_{g'}$ // Renormalize α

8: $\quad \theta^{(t)} \leftarrow \mathrm{P}_\Theta \left(\theta^{(t-1)} - \eta_\theta \left(\frac{N \alpha_g^{(t)} \nabla_\theta l(\pi_{\theta^{(t-1)}}; (x_g, y_w, y_l))}{N_g} \right) \right)$ // Use α to update θ

9: **end for**

10: **Return:** Output the robust policy $\pi(\theta^{(T)})$

$l(\pi_\theta; \cdot)$ accrued by each group, ensuring that groups with higher cumulative losses get higher weights. The size of the group N_g appears as the empirical distribution $\mathcal{D}_g$ involves N_g. We call it alternating update as the updated α^t is used in the update from θ^{t-1} to θ^t. In particular, the gradient descent type update on θ is weighted by α in order to orient the update towards groups with higher losses. The projection operator P_Θ ensures the updated θ^t lies within Θ.

What does the weighted DPO update do? In Line 9 in Algorithm 1, the algorithm performs parameter updates based on the weighted gradients. By using the DPO loss, i.e., $l(\pi_\theta; \cdot) = \log \left(\sigma(\beta h_{\pi_\theta}(\cdot)) \right)$ (see Equation (5)), we obtain the following gradient update expression ignoring the N/N_g constant

$$\alpha_g^{(t)} \nabla_\theta l(\pi_{\theta^{(t-1)}}; (x_g, y_w, y_l)) = \alpha_g^{(t)} \nabla_\theta \log \left(\sigma(\beta h_{\pi_{\theta^{(t-1)}}}(x_g, y_w, y_l)) \right) \tag{11}$$

$$= \alpha_g^{(t)} \sigma \left(r_{\theta^{(t-1)}}(x_g, y_l) - r_{\theta^{(t-1)}}(x_g, y_w) \right) \times \left[\nabla_\theta \log \pi_{\theta^{(t-1)}}(y_w | x_g) - \nabla_\theta \log \pi_{\theta^{(t-1)}}(y_l | x_g) \right].$$

The final term plays the critical role of enhancing the likelihood of the preferred response while simultaneously diminishing the likelihood of the rejected response. This adjustment is proportional to the disparity in rewards between the two responses. Moreover, the inclusion of α_g is pivotal for ensuring group robustness. This coefficient scales the gradient w.r.t. θ based on the cumulative loss previously received by all samples within a specific group. Such a mechanism ensures that the model's focus is increasingly directed towards groups that have historically suffered higher losses. Additionally, the scaling factor N_g guarantees that groups with a smaller volume of data do not face a disadvantage. We defer further details in obtaining the gradient update expression to Appendix B.3.

We demonstrate the global convergence with the following proposition.

Proposition 3. *Suppose that the loss $l(\cdot; (x_g, y, y'))$ is non-negative, convex, $B_\nabla-$Lipschitz continuous, and bounded by B_l for all $(x_g, y, y') \in \mathcal{X} \oplus \mathcal{G} \times \mathcal{Y} \times \mathcal{Y}$ and $\|\theta\|_2 \leq B_\Theta$ for all $\theta \in \Theta$ with convex $\Theta \subset \mathbb{R}^d$. The error of the average iterate of Algorithm 1, i.e., $\pi_{\bar{\theta}(1:T)} = \frac{1}{T} \sum_{t=1}^T \theta^t$, satisfies*

$$\mathbb{E}[\mathcal{L}_{\mathrm{GR}}(\pi_{\bar{\theta}(1:T)})] - \min_{\theta \in \Theta} \mathcal{L}_{\mathrm{GR}}(\pi_\theta) = \mathcal{O}(T^{-1/2}).$$

We defer the proof of this proposition to Appendix E. The analysis follows from an adaptation of the analysis in Nemirovski et al. [30] for the proposed sampling strategy in Algorithm 1[1]. We note that when fine-tuning only the final layer of a LLM, the output policy exists within the log-linear policy class (see Section 3.1), and the corresponding loss function satisfies the assumptions in Proposition 3 (see Lemma 5).

---

[1] We study different sampling strategies and their error rates in Appendix E and we present the most numerically stable one in Algorithm 1.

4.1 Group Robust Identity Preference Optimization

The standard regularized reward maximization objective (Equation (3)) in DPO [35], tends to overlook the KL-regularization and learn deterministic policies. This learned policy assigns preference probability one to winning responses in the data which is often not realistic (see [1][Section 4.2] and Appendix C). Recently, Azar et al. [1] show that the standard regularized reward maximization objective (Equation (3)) in DPO [35] tends to overlook the KL-regularization and learn deterministic policies (see [1, Section 4.2] and Appendix C). They thus propose an alternative approach called *Identity Preference Optimization* (IPO) that is more likely to learn a randomized policy which assigns appropriate probability to the preferred response and prevents overfitting. Following a similar derivation as we did for group robust DPO with details given in Appendix C, we develop the corresponding group robust IPO (GR-IPO):

$$\min_{\pi} \ \mathcal{L}_{\mathrm{GR}}(\pi) := \max_{g \in \mathcal{G}} \mathcal{L}_{\mathrm{IPO}}(\pi, \mathcal{D}_g) = \max_{\alpha \in \Delta_K} \sum_{g=1}^{K} \alpha_g \Big(\mathbb{E}_{(x_g, y_w, y_l) \sim \mathcal{D}_g} \Big[h_\pi(x_g, y_w, y_l) - \tfrac{1}{2\beta} \Big]^2 \Big).$$

For the log-linear policy class (introduced in Section 3.1), the objective function simplifies to

$$\min_{\theta \in \Theta} \max_{\alpha \in \Delta_K} \sum_{g=1}^{K} \alpha_g \Big(\mathbb{E}_{(x_g, y_w, y_l) \sim \mathcal{D}_g} \Big[\big(\langle \phi(x_g, y_w) - \phi(x_g, y_l), \theta \rangle - \tfrac{1}{2\beta} \big)^2 \Big] \Big).$$

To solve the GR-IPO above, it suffices to use Algorithm 1 with slight modifications, see Algorithm 2 in Appendix C. In particular, the update of θ is replaced by a weighted regression update:

$$\widehat{\theta} \leftarrow \mathrm{argmin}_{\theta \in \Theta} \sum_{(x_g, y_w, y_l) \sim \mathcal{D}} \Big[\frac{\alpha_g}{N_g} \big(\langle \phi(x_g, y_w) - \phi(x_g, y_l), \theta \rangle - \frac{1}{2\beta} \big)^2 \Big].$$

For fixed α, we show (in Appendix C) that such an update admits a closed-form solution:

$$\widehat{\theta} = \frac{1}{2\beta} (S^T W S)^{-1} S^T W \mathbf{1} \quad \text{with} \quad W := \mathrm{Diag} \left[\frac{\alpha_{g^{(1)}}}{N_{g^{(1)}}}, \dots, \frac{\alpha_{g^{(N)}}}{N_{g^{(N)}}} \right],$$

where $g^{(i)}$ is the group of each sample i, $N_{g^{(i)}}$ is the number of samples in group $g^{(i)}$ and $\mathbf{1}$ is a column vector of ones of dimension N. Here S is a matrix

$$S := \Big[(\phi(x_g^{(1)}, y_w^{(1)}) - \phi(x_g^{(1)}, y_l^{(1)}))^T, \dots, (\phi(x_g^{(N)}, y_w^{(N)}) - \phi(x_g^{(N)}, y_l^{(N)}))^T \Big].$$

Each row of S represents the difference in feature mappings ϕ of the preferred and less preferred response for each prompt. The group robust IPO (GR-IPO) algorithm is presented in Appendix C, and its empirical results are shown in Section 5.

5 Experiments

In this section, we study the empirical performance of our proposed Algorithm 1 on synthetic and real-world datasets[1]. First, we simulate multi-group data disparities by varying the size and preference distributions of two synthetic groups. In the real-world setup, we study the alignment of an LLM to the real preferences of people from various countries. We examine whether GRPO aligns the LLM in a more equitable manner to reduce discrepancies in alignment among various groups. Finally, we demonstrate that performance is improved by explicitly addressing the grouped nature of the data during alignment.

---

[1]Codes for synthetic and real-world experiments can be found at https://github.com/rsshyam/Group-robust-preference-optimization-bandits and https://github.com/rsshyam/Group-robust-preference-optimization, respectively.
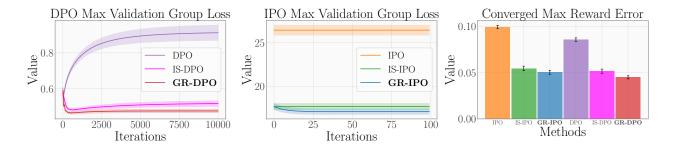
Figure 2: Algorithm 1 (GR-DPO and GR-IPO) leads to a lower worst-case validation loss and reward error compared to importance sampling and vanilla methods. Results refer to the scenario in which groups have different sizes but same responses' distribution. Note that the gap between Algorithm 1 and importance sampling is smaller than in Figure 2. This is expected considering that the primary difference between groups arises from data imbalance, which is handled by importance sampling.
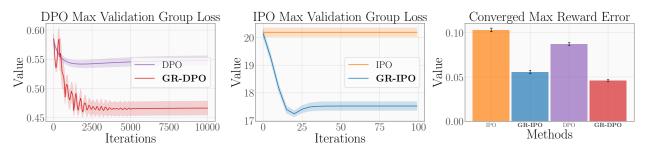

Figure 3: Algorithm 1 (GR-DPO and GR-IPO) leads to a lower worst-case validation loss and reward error compared to the non-robust vanilla methods. Results refer to the scenario in which groups have same sizes but different responses' distribution. Unlike the setups of Figure 2 and Figure 4 importance sampling has no effect here (it coincides with vanilla DPO/IPO since groups have the same sizes).

5.1 Synthetic Experiments

We evaluate the performance of Algorithm 1 using synthetically generated group preference data for the loss function $l(\pi_{\theta;.})$ – either DPO loss or IPO loss and denote them as GR-DPO and GR-IPO, respectively. We compare them against vanilla DPO and IPO ([35, 1]), and the importance-sampling (IS) variants of DPO and IPO (where the loss of each datapoint is inversely weighted by its group data size).

Experimental Setup. Our experiments are designed to analyze settings where there exist multiple groups with distinct characteristics. We adapt the standard (non-group based) experimental setup proposed by [24] for the group preferences setting by incorporating group information into the reward function $r :$ $\mathcal{X} \times \mathcal{Y} \times \mathcal{G} \to \mathbb{R}$. Here, $\mathcal{X}$ represents a two-dimensional state space $[0, 1] \times [0, 1]$, $\mathcal{Y}$ denotes a discrete action space $\{0, 1, 2, 3, \ldots, n\}$, and $\mathcal{G}$ signifies a discrete group space $\{0, 1, 2, \ldots, K\}$. The reward function, defined by the group-dependent feature vector $\phi(x, y, g)$ and parameter vector θ_g, is given as $r(x, y, g) := \langle \phi(x, y, g), \theta_g \rangle$. Further, we consider $n = 7$, $K = 2$ and denote $x := (x_0, x_1) \in \mathcal{X}$. The feature vector $\phi(x, y, g) := (\phi_0(x, y, g), \phi_1(x, y, g), \phi_2(x, y, g), \phi_3(x, y, g))$ and parameters $\theta_g \; \forall g \in \mathcal{G}$ are defined as follows:

$$\phi_i(x, y, g) := \begin{cases} \left(\frac{y}{n+1} + 1\right) \cdot \cos(x_{\lfloor i/2 \rfloor} \cdot \pi) & \text{if } i\%2 = g \\ \left(\frac{1}{\frac{y}{n+1}+1}\right) \cdot \sin(x_{\lfloor i/2 \rfloor} \cdot \pi) & \text{otherwise} \end{cases}, \quad \theta_0 := (1, 3, 1, 3), \theta_1 := (3, 1, 3, 1).$$

Note, that the the feature vectors $\phi(x, y, g)$ have a coordinate-flipped relationship across groups. Also, we
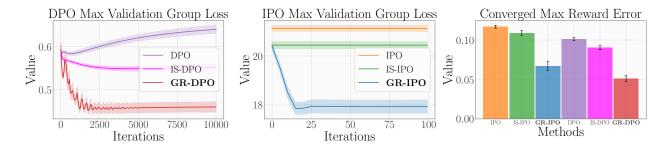
9

Figure 4: Synthetic experiments: Algorithm 1 (GR-DPO and GR-IPO) leads to a significantly lower worst-case validation loss and reward error compared to importance sampling (IS-DPO/IPO) and vanilla methods (DPO, IPO). Results refer to the scenario in which groups have different sizes and responses' distribution.

study two other feature parameterizations and include their experimental results in Appendix D.2.

We consider the following scenarios: **(i)** Groups are imbalanced in terms of size but have the same distribution over responses, **(ii)** Groups are balanced in terms of size but have different response distributions, and **(iii)** Groups are imbalanced in terms of size and also have different response distributions. Note that having different response distributions leads to a difference in the difficulty of learning, since groups with responses distant from each other (in terms of rewards or preference probabilities) are typically more distinguishable and easier to learn. We discuss in Appendix D.1 how we generate these three scenarios.

Implementation. Leveraging the linearity in the reward model, we utilize a log-linear policy class parameterized by θ: $\pi_\theta(y|x) = \frac{\exp\langle\phi(x,y,g),\theta\rangle}{\sum_{y'\in\mathcal{Y}}\exp\langle\phi(x,y',g),\theta\rangle}$. We run Algorithm 1 for both DPO and IPO loss relative to the policy class detailed above with a dataset of 300 action pairs with preferences.

Evaluation Metrics. We use the following criteria to assess the performance of the algorithms:

Max Validation Loss. For each group g, with preference data denoted as $(x^i, y_w^i, y_l^i, g)_{i=1}^{N_g}$, where N_g is the number of data points in the group, we compute the DPO/IPO validation loss separately for each group and identify the maximum loss among them in each run.

Max Reward Error. This metric compares the true reward of the optimal action determined by θ_g^* with that of the action deemed optimal by estimate $\widehat{\theta}$ for each group, and identifies the maximum error across all groups in each run. In particular, for data in the form $(x^i, g)_{i=1}^{N_g}$, which includes only states and groups, we calculate reward errors for every group g as follows: $\mathbb{E}_{(x,g)\sim(x^i,g)_{i=1}^{N_g}}[\max_y\langle\phi(x,y,g),\theta_g^*\rangle - \langle\phi(x,\text{argmax}_y\langle\phi(x,y,g),\widehat{\theta}\rangle,g),\theta_g^*\rangle]$.

Results. We present the average performance of Algorithm 1 (error bars over 20 seeds) alongside baseline methods in Figures 2 to 4 for scenarios **(i)**, **(ii)** and **(iii)** respectively. Our findings indicate that the robust methods consistently surpass both vanilla and importance-sampling approaches. Notably, the robust methods demonstrate significant superiority in uneven group scenarios, where the importance-sampling technique falls short as it exclusively deals with data imbalance.

5.2 Global Opinion Experiments

For the real-data experiments, we consider the survey dataset *GlobalOpinionQA* ([13]) and the publicly available Gemma-2B model [45]. [1] The data contains multiple choice questions answered by participants from various countries, amounting to 2,554 questions covering various topics, including politics, media, technology, religion, race, and ethnicity. For each question, the dataset provides a probability vector over the choices, signifying the percentage of people from a particular country choosing each option. Note that

---

[1] The dataset and model are available at Hugging Face (https://huggingface.co/datasets/Anthropic/llm_global_opinions?row=0, https://huggingface.co/google/gemma-2b).
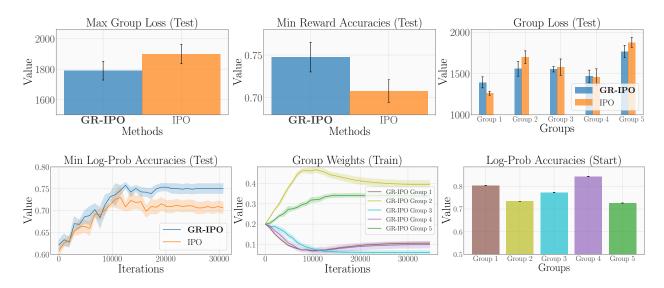
Figure 5: Global opinion data: **Top plots:** GR-IPO leads to better worst-case final test loss and reward accuracy compared to IPO. Moreover, it leads to more balanced losses across the different groups, reducing the gap between best and worst-group loss (Group-1 vs. Group-5). **Bottom plots:** Log-prob. accuracy (left plot) and group weights (middle plot) during GR-IPO training. GR-IPO increases the weight on worse-performing groups (Groups-2,5) and decreases it on high-performing ones (Groups-1,3,4), leading to better worst-case accuracy. Groups-2,5 are the ones with worse log-prob. accuracy at the beginning of training (right plot with a random subset of the training data). We show the corresponding end-of-training log-prob. accuracies for GR-IPO in Figure 12 of Appendix D.

this probability vector would be different for different countries. Hence, the goal is to align the LLM to the probability vector corresponding to each country in a robust manner.

We consider the following five countries in the dataset: Nigeria, Egypt, India, China and Japan, with data sizes 572, 570, 376, 309, and 712, respectively. We construct our training set as follows: For the SFT training, we choose the best option (the choice with the highest probability) as the target. For both IPO and GR-IPO training, we consider the best option as the winning response and another randomly chosen option as the losing response. We outline the exact prompt we use in Appendix D.

We run the SFT training for one epoch over the training data on the pre-trained Gemma-2B model. For both IPO/GR-IPO training we use the AdamW [25] optimizer with adaptive learning rates. We then evaluate both the methods based on the worst group loss and accuracy. Here, the accuracy refers to the percentage of winning response and losing response pairs correctly ordered by the learned preference function (Equation (35)). We defer further training and hyperparameter details to Appendix D.

Results. We present the average performance of GR-IPO over five seeds alongside IPO in Figure 5 (top plots). Our findings indicate that GR-IPO outperforms IPO in terms of maximum group loss and minimum group reward accuracies. Moreover, GR-IPO effectively reduces the imbalance in loss values among different groups. Additionally, we observe an improvement in log-probability accuracies (which measure if the probability assigned by the fine-tuned model is higher for the winning response compared to the losing response) for both IPO and GR-IPO, with GR-IPO demonstrating better alignment for the worst-performing group compared to IPO.

Insights. We further note that the worst-performing groups are Groups-2,5, as shown in Figure 5. GR-IPO improves the loss for these groups by assigning more weight to them, as illustrated in Figure 5 (bottom middle plot). Additionally, we plot the initial log-probability accuracies for different groups in Figure 5 (bottom right plot), assessing how accurately the SFT model classifies the winning versus losing response for different groups. It is evident that Groups-2,5 are already underperforming. Given that SFT training

converges within one epoch without any discrepancies between groups, this indicates that the base LLM inherently struggles with classifying responses for Groups-2,5. However, by employing GR-IPO, we have mitigated the imbalance in the performance of the fine-tuned model.

6   Conclusions

We formalize the problem of robustly aligning an LLM to preference distributions from diverse groups. To tackle the same, we introduced GRPO, a group robust formulation of reward-free RLHF, aiming to minimize worst-case loss among groups. We explored the theoretical aspects of GRPO and demonstrated its improved robust alignment performance through various experiments. We believe our approach will be highly valuable for future tailored LLM fine-tuning, specifically aimed at aligning with the needs of diverse teams and user groups. In a broader context, it holds promise for mitigating biases and discrepancies across various societal groups encountered in the task-specific adaptation of LLMs.

Limitations. When the dataset is balanced among groups and difficulty levels are comparable, our GRPO approach does not offer a significant advantage over standard reward-free RLHF algorithms. Further, in scenarios where optimizing worst-case performance is less critical, robust group performance can still be ensured by trading off worst-case for average performance using a *trade-off parameter*. This modified objective and the necessary algorithmic changes are elaborated in Appendix B.4. The appropriate tuning of the trade-off parameter for the specific application remains a subject for future investigation.

7   Acknowledgments

References

[1]   Mohammad Gheshlaghi Azar et al. "A general theoretical paradigm to understand learning from human preferences". In: *arXiv preprint arXiv:2310.12036* (2023).

[2]   Yuntao Bai et al. "Constitutional ai: Harmlessness from ai feedback". In: *arXiv preprint arXiv:2212.08073* (2022).

[3]   Yuntao Bai et al. "Training a helpful and harmless assistant with reinforcement learning from human feedback". In: *arXiv preprint arXiv:2204.05862* (2022).

[4]   Ralph Allan Bradley and Milton E Terry. "Rank analysis of incomplete block designs: I. The method of paired comparisons". In: *Biometrika* 39.3/4 (1952), pp. 324–345.

[5]   Tianchi Cai et al. "ULMA: Unified Language Model Alignment with Demonstration and Point-wise Human Preference". In: *arXiv preprint arXiv:2312.02554* (2023).

[6]   Stephen Casper et al. "Open problems and fundamental limitations of reinforcement learning from human feedback". In: *arXiv preprint arXiv:2307.15217* (2023).

[7] Souradip Chakraborty et al. "MaxMin-RLHF: Towards Equitable Alignment of Large Language Models with Diverse Human Preferences". In: *arXiv preprint arXiv:2402.08925* (2024).

[8] Lili Chen et al. "Decision transformer: Reinforcement learning via sequence modeling". In: *Advances in neural information processing systems* 34 (2021), pp. 15084–15097.

[9] Zixiang Chen et al. "Self-play fine-tuning converts weak language models to strong language models". In: *arXiv preprint arXiv:2401.01335* (2024).

[10] Paul F Christiano et al. "Deep reinforcement learning from human preferences". In: *Advances in neural information processing systems* 30 (2017).

[11] Nirjhar Das et al. "Provably Sample Efficient RLHF via Active Preference Optimization". In: *arXiv preprint arXiv:2402.10500* (2024).

[12] Hanze Dong et al. "Raft: Reward ranked finetuning for generative foundation model alignment". In: *arXiv preprint arXiv:2304.06767* (2023).

[13] Esin Durmus et al. "Towards measuring the representation of subjective global opinions in language models". In: *arXiv preprint arXiv:2306.16388* (2023).

[14] Kawin Ethayarajh et al. "Kto: Model alignment as prospect theoretic optimization". In: *arXiv preprint arXiv:2402.01306* (2024).

[15] Leo Gao, John Schulman, and Jacob Hilton. "Scaling laws for reward model overoptimization". In: *International Conference on Machine Learning*. PMLR. 2023, pp. 10835–10866.

[16] Caglar Gulcehre et al. "Reinforced self-training (rest) for language modeling". In: *arXiv preprint arXiv:2308.08998* (2023).

[17] Jiwoo Hong, Noah Lee, and James Thorne. "Reference-free monolithic preference optimization with odds ratio". In: *arXiv preprint arXiv:2403.07691* (2024).

[18] Jian Hu et al. "Aligning language models with offline reinforcement learning from human feedback". In: *arXiv preprint arXiv:2308.12050* (2023).

[19] Shawn Im and Yixuan Li. "Understanding the Learning Dynamics of Alignment with Human Feedback". In: *arXiv preprint arXiv:2403.18742* (2024).

[20] Kaixuan Ji, Jiafan He, and Quanquan Gu. "Reinforcement Learning from Human Feedback with Active Queries". In: *arXiv preprint arXiv:2402.09401* (2024).

[21] Timo Kaufmann et al. "A survey of reinforcement learning from human feedback". In: *arXiv preprint arXiv:2312.14925* (2023).

[22] Nathan Lambert, Thomas Krendl Gilbert, and Tom Zick. "Entangled preferences: The history and risks of reinforcement learning and human feedback". In: *arXiv preprint arXiv:2310.13595* (2023).

[23] Harrison Lee et al. "Rlaif: Scaling reinforcement learning from human feedback with ai feedback". In: *arXiv preprint arXiv:2309.00267* (2023).

[24] Ziniu Li, Tian Xu, and Yang Yu. "Policy Optimization in RLHF: The Impact of Out-of-preference Data". In: *arXiv preprint arXiv:2312.10584* (2023).

[25] Ilya Loshchilov and Frank Hutter. "Decoupled Weight Decay Regularization". In: *International Conference on Learning Representations (ICLR)* (2018).

[26] Viraj Mehta et al. "Sample Efficient Reinforcement Learning from Human Feedback via Active Exploration". In: (2023).

[27] Rémi Munos et al. "Nash learning from human feedback". In: *arXiv preprint arXiv:2312.00886* (2023).

[28] Reiichiro Nakano et al. "Webgpt: Browser-assisted question-answering with human feedback". In: *arXiv preprint arXiv:2112.09332* (2021).

[29] Hongseok Namkoong and John C Duchi. "Stochastic gradient methods for distributionally robust optimization with f-divergences". In: *Advances in neural information processing systems* 29 (2016).

[30] Arkadi Nemirovski et al. "Robust stochastic approximation approach to stochastic programming". In: *SIAM Journal on optimization* 19.4 (2009), pp. 1574–1609.

[31] Yonatan Oren et al. "Distributionally robust language modeling". In: *arXiv preprint arXiv:1909.02060* (2019).

[32] Long Ouyang et al. "Training language models to follow instructions with human feedback". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 27730–27744.

[33] Jing-Cheng Pang et al. "Language model self-improvement by reinforcement learning contemplation". In: *arXiv preprint arXiv:2305.14483* (2023).

[34] Alec Radford et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[35] Rafael Rafailov et al. "Direct preference optimization: Your language model is secretly a reward model". In: *arXiv preprint arXiv:2305.18290* (2023).

[36] Corby Rosset et al. "Direct nash optimization: Teaching language models to self-improve with general preferences". In: *arXiv preprint arXiv:2404.03715* (2024).

[37] Shiori Sagawa et al. "Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization". In: *arXiv preprint arXiv:1911.08731* (2019).

[38] John Schulman et al. "Proximal policy optimization algorithms". In: *arXiv preprint arXiv:1707.06347* (2017).

[39] Avi Singh et al. "Beyond human data: Scaling self-training for problem-solving with language models". In: *arXiv preprint arXiv:2312.06585* (2023).

[40] Maurice Sion. "On general minimax theorems." In: (1958).

[41] Feifan Song et al. "Preference ranking optimization for human alignment". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2024.

[42] Nisan Stiennon et al. "Learning to summarize with human feedback". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 3008–3021.

[43] Gokul Swamy et al. "A minimaximalist approach to reinforcement learning from human feedback". In: *arXiv preprint arXiv:2401.04056* (2024).

[44] Yunhao Tang et al. "Generalized Preference Optimization: A Unified Approach to Offline Alignment". In: *arXiv preprint arXiv:2402.05749* (2024).

[45] Gemma Team et al. "Gemma: Open models based on gemini research and technology". In: *arXiv preprint arXiv:2403.08295* (2024).

[46] Haoxiang Wang et al. "Arithmetic Control of LLMs for Diverse User Preferences: Directional Preference Alignment with Multi-Objective Rewards". In: *arXiv preprint arXiv:2402.18571* (2024).

[47] Tianhao Wu et al. "Pairwise proximal policy optimization: Harnessing relative feedback for llm alignment". In: *arXiv preprint arXiv:2310.00212* (2023).

[48] Yue Wu et al. "Self-Play Preference Optimization for Language Model Alignment". In: *arXiv preprint arXiv:2405.00675* (2024).

[49] Sang Michael Xie et al. "DoReMi: Optimizing Data Mixtures Speeds Up Language Model Pretraining". In: *arXiv preprint arXiv:2305.10429* (2023).

[50] Wei Xiong et al. "Iterative preference learning from human feedback: Bridging theory and practice for RLHF under KL-constraint". In: *ICLR 2024 Workshop on Mathematical and Empirical Understanding of Foundation Models*. 2023.

[51] Rui Yang et al. "Rewards-in-Context: Multi-objective Alignment of Foundation Models with Dynamic Preference Adjustment". In: *arXiv preprint arXiv:2402.10207* (2024).

[52] Chenlu Ye et al. "A theoretical analysis of nash learning from human feedback under general kl-regularized preference". In: *arXiv preprint arXiv:2402.07314* (2024).

[53] Weizhe Yuan et al. "Self-rewarding language models". In: *arXiv preprint arXiv:2401.10020* (2024).

[54] Siyan Zhao, John Dang, and Aditya Grover. "Group Preference Optimization: Few-Shot Alignment of Large Language Models". In: *arXiv preprint arXiv:2310.11523* (2023).

[55] Yao Zhao et al. "Slic-hf: Sequence likelihood calibration with human feedback". In: *arXiv preprint arXiv:2305.10425* (2023).

[56] Han Zhong et al. "DPO Meets PPO: Reinforced Token Optimization for RLHF". In: *arXiv preprint arXiv:2404.18922* (2024).

[57] Daniel M Ziegler et al. "Fine-tuning language models from human preferences". In: *arXiv preprint arXiv:1909.08593* (2019).

Appendix

The supplementary section constitutes the following: An extended related work section in Appendix A, proofs for the theoretical insights and the weighted DPO gradient expression in Appendix B, an overview of IPO and the simplification for robust IPO in Appendix C, detailed experimental setup in Appendix D, and convergence proof for Algorithm 1 and other sampling strategies in Appendix E.

A   Additional Related Work

In this section, we provide a more extensive exposition of related work.

The Reinforcement Learning from Human Feedback (RLHF) is a concrete way to adapt a LLM towards specific tasks or objectives using human feedback. This framework for fine-tuning LLMs is inspired from [10] that uses human feedback to improve reinforcement learning policies. For LLMs, it was first established in [57] and further developed in [42, 32]. It involves learning a reward function using the pairwise human feedback data that provides comparisons between responses for a given prompt. This reward learning is performed through the Bradley-Terry model [4]. This learned reward model is then used as an objective for optimizing the LLM policy using Proximal Policy Optimization [38].

However, obtaining high-quality human feedback data is expensive and several works such as [33, 2, 23] study the usage of AI feedback to reduce reliance on human input. Further, studies like [11, 26, 20] analyze the usage of active learning strategies to minimize the amount of human feedback data required to perform RLHF. For a comprehensive overview and perspective of the RLHF topic, with detailed discussions on various approaches and gaps, we refer the reader to [22, 6, 21].

Due to the inherent issues of tuning the hyperparameters of a PPO ([47, 35]) and susceptible nature of reward models ([15, 46]), alternative approaches to the PPO-based RLHF have been proposed. Foremost of them is the rejection sampling fine-tuning ([12, 16, 46]) which is inspired by the best-of-n inference technique ([28]). The technique involves sampling n responses per prompt and fine-tuning the model based on the highest scoring ones in terms of a learned reward function. Further works such as [18, 51] avoid reward learning and propose conditional Supervised Fine-Tuning (SFT) inspired by the reward-conditioned RL ([8]).

Another strategy proposed as an alternative to PPO-based RLPHF is DPO (Rafailov et al. [35]), wherein one optimizes the policy directly based on human preferences, bypassing the need for learning a separate reward model. This method simplifies the training process and potentially reduces overfitting and model misalignment issues. Theoretical advancements in understanding the dynamics of DPO have been provided by [19, 50], who analyzed the convergence and stability of DPO policy learning. Some other works such as [1, 55, 44, 41, 14] also study a similar reward-free RLHF setup. In particular, [1] analyze the potential overfitting issues in DPO. Further, [14, 5] propose strategies that bypass the need for preference datasets. [36, 52, 27] adopt a game-theory perspective to learn Nash equilibrium policies. Typically, the reward-free RLHF approaches utilize a reference policy trained via supervised fine-tuning, which is then further optimized. However, [17] propose Monolithic Preference Optimization without reference model, bypassing supervised fine-tuning entirely. [56] address the RLHF problem from an MDP framework, introducing a DPO-type algorithm integrated with Proximal Policy Optimization (PPO). Additionally, works such as [48, 43, 9, 39, 53] investigate self-play preference optimization, where new data generated in each round by the current policy is used to train an improved policy. Our work utilizes a reward-free framework similar to [35, 1] but differs from all the above works in the following sense. Unlike previous works that assume a single preference distribution, we consider fine-tuning a LLM given data from multiple preference distributions across diverse groups. Further, we aim to fine-tune the LLM in a robust manner such that there is minimal imbalance in performance across all groups.

Robust language modeling technique in the context of language model was first studied in [31] to optimize performance over a wide-range of topics. It was further improved by [49] wherein they use smaller model to learn the weights/importance that needs to be assigned to different topics and train a larger model based on these weights. Both works aim to minimize the worst group loss based on the group-DRO approach and focus on the pre-training aspects of language models. Concrete theoretical study of this group-DRO approach was performed in [37] which studies the minimax formulation of this group-DRO problem based on [29].

In the RLHF setup, recent research has increasingly focused on enhancing the robustness of policy learning to address certain inherent gaps in the RLHF framework as detailed in [6] such as reward hacking and model misspecification. In particular, [3] formulated a weighted group loss for reward model learning w.r.t. a parameter λ that determines the importance assigned to a group and apply it to the anthropic Harmless vs Helpful data [1]. Another work [7] considers alignment to different sub-populations by learning multiple reward functions corresponding to each sub-population and learning a robust policy (max-min) w.r.t. the reward functions. Distinct from previous works, our approach circumvents reward model learning and integrates group robustness directly into the reward-free tuning framework. This allows us to directly learn a robust policy without the need for learning reward models. To learn this robust policy, we present a concrete algorithm that adaptively assigns weights to the losses of different groups and optimizes the policy to minimize this weighted loss. Additionally, our algorithm features a novel gradient estimator specifically designed for the group robust DPO problem.

Other studies addressing robustness in preference optimization, such as [19] and [24], focus on different facets of robustness, including robustness to noise and resilience against out-of-preference data. In the non-robust group preference alignment setup, [54] explores learning group preferences with LLMs by incorporating group information into prompts. However, their approach does not involve fine-tuning a LLM but training a separate transformer module that optimally selects an example sequence of prompts, LLM responses, and group preferences for in-context learning using a LLM.

---

[1]The data is publicly available in Huggingface website https://huggingface.co/datasets/Anthropic/hh-rlhf

B Theoretical Insights

In this section, we detail the proofs for the theoretical insights elucidated in Section 3 and the weighted DPO gradient expression discussed in Section 4.

B.1 Robust Objective for the Log-linear Policy Class

Here, we describe obtaining the robust objective for the log-linear policy class as in Equation (9). Starting from the robust objective for a general policy class in Equation (8), $\mathcal{L}_{\text{GR}}$ can be specialized as follows:

$$
\mathcal{L}_{\text{GR}}(\pi) = \max_{\alpha \in \Delta_K} \sum_{g=1}^{K} \alpha_g \Big(- \mathbb{E}_{(x_g, y_w, y_l) \sim \mathcal{D}_g} \Big[\log \Big(\sigma(\beta h_\pi(x_g, y_w, y_l)) \Big) \Big] \Big)
$$

$$
\overset{(i)}{=} \max_{\alpha \in \Delta_K} \sum_{g=1}^{K} \alpha_g \Big(- \mathbb{E}_{(x_g, y_w, y_l) \sim \mathcal{D}_g} \Big[\log \Big(\sigma(\beta \log(\tfrac{\pi(y_w|x_g)}{\pi_{\text{ref}}(y_w|x_g)}) - \beta \log(\tfrac{\pi(y_l|x_g)}{\pi_{\text{ref}}(y_l|x_g)})) \Big) \Big] \Big)
$$

$$
\overset{(ii)}{=} \max_{\alpha \in \Delta_K} \sum_{g=1}^{K} \alpha_g \Big(- \mathbb{E}_{(x_g, y_w, y_l) \sim \mathcal{D}_g} \Big[\log \Big(\sigma(\beta \log(\pi(y_w|x_g)) - \beta \log(\pi(y_l|x_g))) \Big) \Big] \Big)
$$

$$
\overset{(iii)}{=} \max_{\alpha \in \Delta_K} \sum_{g=1}^{K} \alpha_g \Big(- \mathbb{E}_{(x_g, y_w, y_l) \sim \mathcal{D}_g} \Big[\log \Big(\sigma(\beta \log(\tfrac{\exp f_\theta(x_g, y_w)}{\sum_{y \in \mathcal{Y}} \exp f_\theta(x_g, y)}) -
$$

$$
\beta \log(\tfrac{\exp f_\theta(x_g, y_l)}{\sum_{y \in \mathcal{Y}} \exp f_\theta(x_g, y)})) \Big) \Big] \Big)
$$

$$
\overset{(iv)}{=} \max_{\alpha \in \Delta_K} \sum_{g=1}^{K} \alpha_g \Big(- \mathbb{E}_{(x_g, y_w, y_l) \sim \mathcal{D}_g} \Big[\log \Big(\sigma(\beta \log(\exp \theta^T \phi(x_g, y_w)) -
$$

$$
\beta \log(\exp \theta^T \phi(x_g, y_l))) \Big) \Big] \Big)
$$

$$
= \max_{\alpha \in \Delta_K} \sum_{g=1}^{K} \alpha_g \Big(- \mathbb{E}_{(x_g, y_w, y_l) \sim \mathcal{D}_g} \Big[\log \Big(\sigma(\beta \langle \phi(x_g, y_w) - \phi(x_g, y_l), \theta \rangle) \Big) \Big] \Big).
$$

Here, (i) follows from the definition of $h_\pi(x_g, y_w, y_l) = \log(\tfrac{\pi(y_w|x_g)}{\pi_{\text{ref}}(y_w|x_g)}) - \log(\tfrac{\pi(y_l|x_g)}{\pi_{\text{ref}}(y_l|x_g)})$, (ii) follows from the fact that for a uniform reference policy $\pi_{\text{ref}}(y_w|x_g) = \pi_{\text{ref}}(y_l|x_g)$, (iii) follows from substituting $\pi_\theta(y|x_g) = \tfrac{\exp f_\theta(x_g, y)}{\sum_{y \in \mathcal{Y}} \exp f_\theta(x_g, y)}$, and (iv) follows from substituting $f_\theta(x_g, y) = \theta^T \phi(x_g, y)$.

B.2 Robust KL-Regularized Policy Maximization

In this section, we consider the robust version of the classical KL-regularized reward maximization objective (Equation (3)) detailed in Equation (10) for a reward function $r(x_g, y)$, reference policy π_{ref}, and a general class of policies Π.

$$
\max_{\pi \in \Pi} \min_{g \in \mathcal{G}} \mathbb{E}_{x_g \sim \mathcal{P}_{x_g}, y \sim \pi(\cdot|x_g)} \big[r(x_g, y) \big] - \beta \text{KL} \Big[\pi(y \mid x_g) || \pi_{\text{ref}}(y \mid x_g) \Big]. \tag{12}
$$

Lemma 4. *The optimal policy π^* for the robust KL-regularized reward maximization objective (Equation (12)) for a reward function $r(x_g, y)$, reference policy π_{ref} is*

$$
\pi^*(y|x_g) = \frac{1}{Z(x_g)} \pi_{ref}(y|x_g) \exp(\frac{r(x_g, y)}{\beta}), \tag{13}
$$

where $Z(x_g) = \sum_y \pi_{ref}(y|x_g) \exp(\frac{1}{\beta} r(x_g, y))$ is a partition function.

18

Proof. We recast Equation (12) based on Equation (8) and perform the following analysis similar to [35][Appendix A.1]:

$$\max_{\pi \in \Pi} \min_{\alpha \in \Delta_K} \sum_{g=1}^{K} \alpha_g \Big(\mathbb{E}_{x_g \sim \mathcal{P}_{x_g}} \Big[\mathbb{E}_{y \sim \pi(\cdot|x_g)} \big[r(x_g, y) \big] - \beta D_{KL} \big[\pi(y|x_g) || \pi_{\text{ref}}(y|x_g) \big] \Big] \Big) \tag{14}$$

$$= \max_{\pi \in \Pi} \min_{\alpha \in \Delta_K} \sum_{g=1}^{K} \alpha_g \Big(\mathbb{E}_{x_g \sim \mathcal{P}_{x_g}} \mathbb{E}_{y \sim \pi(\cdot|x_g)} \big[r(x_g, y) - \beta \log \big(\tfrac{\pi(y|x_g)}{\pi_{\text{ref}}(y|x_g)} \big) \big] \Big) \tag{15}$$

$$= -\beta \min_{\pi \in \Pi} \max_{\alpha \in \Delta_K} \sum_{g=1}^{K} \alpha_g \Big(\mathbb{E}_{x_g \sim \mathcal{P}_{x_g}} \mathbb{E}_{y \sim \pi(\cdot|x_g)} \big[\log \big(\tfrac{\pi(y|x_g)}{\pi_{\text{ref}}(y|x_g)} \big) - \tfrac{r(x_g,y)}{\beta} \big] \Big) \tag{16}$$

$$= -\beta \min_{\pi \in \Pi} \max_{\alpha \in \Delta_K} \sum_{g=1}^{K} \alpha_g \Big(\mathbb{E}_{x_g \sim \mathcal{P}_{x_g}} \mathbb{E}_{y \sim \pi(\cdot|x_g)} \big[-\log(Z(x_g)) + \log \big(\tfrac{\pi(y|x_g)}{\frac{1}{Z(x_g)} \pi_{\text{ref}}(y|x_g) \exp(\frac{r(x_g,y)}{\beta})} \big) \big] \Big). \tag{17}$$

Further, by defining $\pi^*(y|x_g) = \frac{1}{Z(x_g)} \pi_{\text{ref}}(y|x_g) \exp(\frac{r(x_g,y)}{\beta})$, we can rewrite Equation (17) as:

$$\beta \min_{\pi \in \Pi} \max_{\alpha \in \Delta_K} \sum_{g=1}^{K} \alpha_g \Big(\mathbb{E}_{x_g \sim \mathcal{P}_{x_g}} \mathbb{E}_{y \sim \pi(\cdot|x_g)} \big[\log \big(\tfrac{\pi(y|x_g)}{\pi^*(y|x_g)} \big) - \log(Z(x_g)) \big] \Big) \tag{18}$$

$$= \beta \min_{\pi \in \Pi} \max_{\alpha \in \Delta_K} \sum_{g=1}^{K} \alpha_g \Big(\mathbb{E}_{x_g \sim \mathcal{P}_{x_g}} \big[D_{KL} \big(\pi(\cdot|x_g) || \pi^*(\cdot|x_g) \big) - \log(Z(x_g)) \big] \Big) \tag{19}$$

$$= \beta \min_{\pi \in \Pi} roL(\pi), \tag{20}$$

where we use $roL(\pi) := \max_{\alpha \in \Delta_K} \sum_{g=1}^{K} \alpha_g \Big(\mathbb{E}_{x_g \sim \mathcal{P}_{x_g}} \big[D_{KL} \big(\pi(\cdot|x_g) || \pi^*(\cdot|x_g) \big) - \log(Z(x_g)) \big] \Big)$.

It is not hard to show that $roL(\pi)$ is minimized by π^*. Indeed, let us consider any other policy $\pi \neq \pi^*$. Then, we have

$$roL(\pi) = \beta \max_{\alpha \in \Delta_K} \sum_{g=1}^{K} \alpha_g \Big(\mathbb{E}_{x_g \sim \mathcal{P}_{x_g}} \Big[D_{KL} \big(\pi(\cdot|x_g) || \pi^*(\cdot|x_g) \big) - \log(Z(x_g)) \Big] \Big) \tag{21}$$

$$\geq \beta \max_{\alpha \in \Delta_K} \sum_{g=1}^{K} \alpha_g \Big(\mathbb{E}_{x_g \sim \mathcal{P}_{x_g}} \Big[-\log(Z(x_g)) \Big] \Big) \tag{22}$$

$$= \beta \max_{\alpha \in \Delta_K} \sum_{g=1}^{K} \alpha_g \Big(\mathbb{E}_{x_g \sim \mathcal{P}_{x_g}} \Big[D_{KL} \big(\pi^*(\cdot|x_g) || \pi^*(\cdot|x_g) \big) - \log(Z(x_g)) \Big] \Big) \tag{23}$$

$$= roL(\pi^*), \tag{24}$$

where the inequality follows since $D_{KL} \big(\pi(\cdot|x) || \pi^*(\cdot|x) \big) \geq 0$. This implies π^* minimises Equation (20). $\square$

Note that, we have proved that the optimal policy expression for Equation (12) aligns with the form of the optimal policy for standard (non-robust) KL-regularized reward maximization Equation (4).

Proposition 2. *Substituting the closed-form solution of the robust KL-regularized policy maximization problem (Equation (10)) into the non-robust KL-regularized reward maximization objective in Equation (2) leads to the same group robust DPO loss $\mathcal{L}_{\text{GR-DPO}}$ in Equation (8).*

19

Proof. In [35], the DPO loss (see Equation (5)) is obtained by using the following reward loss:

$$\mathcal{L}_R(r; \mathcal{D}) = \mathcal{L}_R(r; \{\mathcal{D}_g\}_{g=1}^K) = -\sum_{g=1}^K \mathop{\mathbb{E}}_{(x_g, y_w, y_l) \sim \mathcal{D}_g} \left[\log \left(\sigma \left(r(x_g, y_w) - r(x_g, y_l) \right) \right) \right], \tag{25}$$

and replacing $r(\cdot, \cdot)$ with the relation

$$\pi^*(y|x) = \tfrac{1}{Z(x)} \pi_{\text{ref}}(y|x) \exp \left(\tfrac{r(x,y)}{\beta} \right), \tag{26}$$

which is the solution to

$$\pi^* = \max_{\pi \in \Pi} \sum_{g=1}^K \mathop{\mathbb{E}}_{(x_g \sim \mathcal{P}_{x_g}, y \sim \pi(\cdot|x_g))} \left[r_\phi(x_g, y) \right] - \beta D_{KL} \left[\pi(y|x_g) || \pi_{\text{ref}}(y|x_g) \right]. \tag{27}$$

However, in our *group-robust* formulation, we replace Equation (25) with

$$\max_{g \in \mathcal{G}} \mathcal{L}_R(r; \{\mathcal{D}_g\}_{g=1}^K) = \max_{\alpha \in \Delta_K} \sum_{g=1}^K \alpha_g \left(- \mathop{\mathbb{E}}_{(x_g, y_w, y_l) \sim \mathcal{D}_g} \left[\log \left(\sigma \left(r(x_g, y_w) - r_{(x_g, y_l)} \right) \right) \right] \right), \tag{28}$$

while Equation (27) remains the same. Alternatively, we can replace Equation (27) with its robust version, i.e.,

$$\pi^* = \max_{\pi \in \Pi} \min_{\alpha \in \Delta_K} \sum_{g=1}^K \alpha_g \left(\mathop{\mathbb{E}}_{(x_g \sim \mathcal{P}_{x_g}, y \sim \pi(\cdot|x_g))} \left[r_\phi(x_g, y) \right] - \beta D_{KL} \left[\pi(y|x_g) || \pi_{\text{ref}}(y|x_g) \right] \right), \tag{29}$$

however, we have already shown, in Lemma 4, that this does not change the obtained policy-reward relation from Equation (26). Hence, this proves Proposition 2. $\qquad \square$

B.3 Analysis of the Gradient

We elucidate the steps to obtain the expression for the loss gradient in Equation (11). We can simplify the gradient as follows:

$$\frac{\alpha_g^t \nabla_\theta l(\pi_{\theta^{t-1}}; (x_g, y_w, y_l))}{n_g}$$

$$= \frac{\alpha_g^t \nabla_\theta \log \left(\sigma(\beta h_{\pi_{\theta^{t-1}}}(x_g, y_w, y_l)) \right)}{n_g}$$

$$\overset{(i)}{=} \frac{\alpha_g^t}{n_g} \frac{\sigma'(\beta h_{\pi_{\theta^{t-1}}}(x_g, y_w, y_l))}{\sigma(\beta h_{\pi_{\theta^{t-1}}}(x_g, y_w, y_l)} \times [\beta h'_{\pi_{\theta^{t-1}}}(x_g, y_w, y_l))]$$

$$\overset{(ii)}{=} \frac{\beta \alpha_g^t}{n_g} \left(1 - \sigma(\beta h_{\pi_{\theta^{t-1}}}(x_g, y_w, y_l)) \right) \times [h'_{\pi_{\theta^{t-1}}}(x_g, y_w, y_l))]$$

$$\overset{(iii)}{=} \frac{\beta \alpha_g^t}{n_g} \sigma \left(- \beta h_{\pi_{\theta^{t-1}}}(x_g, y_w, y_l) \right) \times [h'_{\pi_{\theta^{t-1}}}(x_g, y_w, y_l))]$$

$$\overset{(iv)}{=} \frac{\beta \alpha_g^t}{n_g} \sigma \left(\beta \left(\log(\tfrac{\pi_{\theta^{t-1}}(y_l|x)}{\pi_{\text{ref}}(y_l|x)}) - \log(\tfrac{\pi_{\theta^{t-1}}(y_w|x)}{\pi_{\text{ref}}(y_w|x)}) \right) \right) \times [h'_{\pi_{\theta^{t-1}}}(x_g, y_w, y_l))]$$

$$\overset{(v)}{=} \frac{\alpha_g^t}{n_g} \sigma \left(r_{\theta^{t-1}}(x_g, y_l) - r_{\theta^{t-1}}(x_g, y_w) \right) \times [\nabla_\theta \log \pi_{\theta^{t-1}}(y_w|x_g) - \nabla_\theta \log \pi_{\theta^{t-1}}(y_l|x_g)]$$

Here, (i) follows from the derivative of $\log(\cdot)$ function and denoting the derivative of sigmoid function $\sigma(\cdot)$ as $\sigma^{'}(\cdot)$. (ii) and (iii) follow from the fact that for any $s \in \mathbb{R}$, $\sigma^{'}(s) = 1 - \sigma(s) = \sigma(-s)$. Finally, (iv) and (v) follow using the definition of $h_\pi(x, y_w, y_l) = \log(\frac{\pi(y_w|x)}{\pi_{\text{ref}}(y_w|x)}) - \log(\frac{\pi(y_l|x)}{\pi_{\text{ref}}(y_l|x)})$, and substituting $r_\theta(x_g, y) = \beta \log(\frac{\pi_\theta(y|x_g)}{\pi_{\text{ref}}(y|x_g)}) - \beta \log Z(x_g)$ from Equation (4).

B.4 Trading off worst-case vs. average performance

In the robust objective of Equation (8), we intend to minimize the worst-case loss. This often might adversely impact the average loss leading to bad average performance across the different groups. To mitigate this, we propose the following *robust trade-off* direct preference optimization objective for a specified policy π and set of group weights $\mu_1, \ldots, \mu_K$:

$$
\begin{aligned}
\mathcal{L}_{\text{GR},\chi}(\pi) :=& (1 - \chi) \sum_{g=1}^{K} \mu_g \mathcal{L}_{\text{DPO}}(\pi, \mathcal{D}_g) + \chi \max_{g \in \mathcal{G}} \mathcal{L}_{\text{DPO}}(\pi, \mathcal{D}_g) \\
=& (1 - \chi) \sum_{g=1}^{K} \mu_g \Big(- \mathbb{E}_{(x_g, y_w, y_l) \sim \mathcal{D}_g} \Big[\log \Big(\sigma\big(\beta h_\pi(x_g, y_w, y_l)\big) \Big) \Big] \Big) + \\
& \chi \max_{g \in \mathcal{G}} \Big(- \mathbb{E}_{(x_g, y_w, y_l) \sim \mathcal{D}_g} \Big[\log \Big(\sigma\big(\beta h_\pi(x_g, y_w, y_l)\big) \Big) \Big] \Big).
\end{aligned}
\tag{30}
$$

Here, note that the input weights μ_g can be equal for all groups leading to a trade-off between the average and worst-case performance w.r.t. parameter χ. Following Equation (8), this can be equivalently recast into:

$$
\begin{aligned}
\mathcal{L}_{\text{GR},\chi}(\pi) =& (1 - \chi) \sum_{g=1}^{K} \mu_g \Big(- \mathbb{E}_{(x_g, y_w, y_l) \sim \mathcal{D}_g} \Big[\log \Big(\sigma(\beta h_\pi(x_g, y_w, y_l)) \Big) \Big] \Big) \\
& + \chi \max_{\alpha \in \Delta_K} \sum_{g=1}^{K} \alpha_g \Big(- \mathbb{E}_{(x_g, y_w, y_l) \sim \mathcal{D}_g} \Big[\log \Big(\sigma(\beta h_\pi(x_g, y_w, y_l)) \Big) \Big] \Big),
\end{aligned}
\tag{31}
$$

$$
= \max_{\alpha \in \Delta_K} \sum_{g=1}^{K} ((1 - \chi)\mu_g + \chi \alpha_g) \Big(- \mathbb{E}_{(x_g, y_w, y_l) \sim \mathcal{D}_g} \Big[\log \Big(\sigma(\beta h_\pi(x_g, y_w, y_l)) \Big) \Big] \Big)
\tag{32}
$$

where Δ_K represents the $(K - 1)$-dimensional simplex of probabilities. Hence, minimizing the loss in Equation (31) implies that one can implicitly find an optimal reward function using human group preference data that is robust without losing average performance, and obtain a policy that works effectively in terms of both average and worst-case performance:

$$
\min_\pi \ \mathcal{L}_{\text{GR},\chi}(\pi).
\tag{33}
$$

As Equation (8) involves maximizing with respect to α and minimizing with respect to π, it forms a two-player game similar to the one considered in Section 3.1, where the policy π and α act as opponents with inversely related payoffs. However, the contributions of the maximizing player (α) on the final outcome are limited depending on parameters χ and μ.

C IPO Simplification

Within the context of a finite data setting, it is common to record two responses (y, y') for each prompt x. Based on these observations, there might be a tendency to inaccurately deduce that the preference distribution fulfills $p(y \succ y'|x) = 1$. Consequently, and assuming the Bradley-Terry model of preferences, the reward function r has to satisfy $r(y, x) - r(y', x) \to \infty$. Further, given that DPO policy π is directly dependent on the reward function (Equation (4)), it holds that $\frac{\pi(y'|x)}{\pi(y|x)} = 0$. In such a case, the policy overfits irrespective of the KL regularization factor. To circumvent this, [1] consider reward-free RLHF problem. Specifically, they propose preference optimization in the policy objective instead of reward optimization as follows:

$$\max_{\pi} \; \mathbb{E}_{x \sim \rho, y \sim \pi(\cdot|x), y' \sim \mu(\cdot|x)} \big[\Psi(p(y \succ y')|x)\big] - \beta \mathrm{KL}\Big[\pi(y|x)||\pi_{\mathrm{ref}}(y|x)\Big]. \tag{34}$$

Here, $\Psi : [0, 1] \to \mathbb{R}$ is a non-decreasing function, ρ refers to the distribution of prompts and, μ refers to the competing behavior policy. Choosing Ψ as the *identity* function, the optimal policy π^* of Equation (34) has a similar expression to Equation (4) with the reward function $r(\cdot)$ substituted by the following preference function $p(\cdot)$,

$$p(y \succ \mu|x) = \beta \log \frac{\pi^*(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} + \beta \log Z(x). \tag{35}$$

Recalling the definition of $h_{\pi}(x, y, y') = \log(\frac{\pi(y|x)}{\pi_{\mathrm{ref}}(y|x)}) - \log(\frac{\pi(y'|x)}{\pi_{\mathrm{ref}}(y'|x)})$ from Section 2, it follows from Equation (35) that π^* adheres to the following identity that equates the true preference with the policy's preference of y over y':

$$h_{\pi^*}(x, y, y') = \beta^{-1}(p(y \succ \mu|x) - p(y' \succ \mu|x)). \tag{36}$$

where $p(y \succ \mu|x) := \mathbb{E}_{\bar{y} \sim \mu(\cdot|x)}\big[p(y \succ \bar{y}|x)\big]$. The equation motivates us to find a policy π that minimizes the squared differences of two sides in Equation (36),

$$\mathcal{L}(\pi) = \mathbb{E}_{y, y' \sim \mu}\Big[h_{\pi}(x, y, y') - \beta^{-1}(p(y \succ \mu|x) - p(y' \succ \mu|x))\Big]^2. \tag{37}$$

For an empirical dataset where one observes if y_w is more preferable to y_l, the policy minimization objective in Equation (37) becomes the IPO objective:

$$\mathcal{L}_{\mathrm{IPO}}(\pi, \mathcal{D}) = \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}}\Big[h_{\pi}(x, y_w, y_l) - \tfrac{1}{2\beta}\Big]^2. \tag{38}$$

We can simplify Equation (38) for the IPO Loss using Log-linear policy class similar to Equation (9) as follows

$$L_{\mathrm{IPO}}(\pi, \mathcal{D}) = \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}}\Big[\big(\langle\phi(x, y_w) - \phi(x, y_l), \theta\rangle - \frac{1}{2\beta}\big)^2\Big]. \tag{39}$$

Contrary to DPO, the IPO Loss results in a problem formulation equivalent to linear regression for the linear bandit setting. This leads us to obtain a closed form analytical solution for θ

$$\widehat{\theta}_{IPO} = \frac{1}{2\beta}(X^T X)^{-1} X^T \mathbf{1}, \tag{40}$$

where $\mathbf{1} = \{1\}^N$. Here, $X \in \mathbb{R}^{d \times N}$. In particular, each row of X is the difference of the preferred and least preferred feature vectors $\phi(x, y_w) - \phi(x, y_l)$ for $(x, y_w, y_l) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}$

$$X = \begin{bmatrix} \phi(x_1, y_1) - \phi(s_1, y_1') \\ \phi(x_2, y_2) - \phi(x_2, y_2') \\ \vdots \\ \phi(x_n, y_n) - \phi(x_n, y_n') \end{bmatrix}.$$

Algorithm 2 Policy Optimization for Robust IPO in Linear Bandits
---
1: **Initialize:** Step size η_α for group weights α, initial weights $\theta^{(0)}$ of the policy and weights over each group $\alpha^{(0)}$
2: **Input:** Dataset $\mathcal{D}$ with size $N = |\mathcal{D}|$, group size N_g for $g = \{1, 2, \cdots, K\}$
3: **for** $t = 1, \ldots, T$ **do**
4:     Calculate group loss l_g for each group g on $\mathcal{D}_g$
5:     $\alpha' \leftarrow \alpha^{t-1}; \alpha'_g \leftarrow \alpha'_g \exp(\eta_G l_g)$ // Update weights for group g
6:     $\alpha^{(t)} \leftarrow \alpha' / \sum_{g'} \alpha'_{g'}$ // Renormalize α
7:     $\theta^t \leftarrow \arg\min_\theta \mathbb{E}_{(x,y_w,y_l,g)\sim\mathcal{D}} \left[\alpha_g^{(t)} \big(\langle \phi(x,y_w,g) - \phi(x,y_l,g), \theta \rangle - \frac{1}{2\beta} \big)^2 \right]$
8: **end for**
9: **Return:** Output the robust policy $\pi(\theta^t)$

When re-written in this form it is trivial to see that the stability of the IPO Loss depends upon the rank of the matrix X.

Group IPO: We begin by conducting experiments with IPO considering the existence of closed form solution for this particular setting. In particular, given a set of preference data $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2\}$ from either groups with varying ratio, one aims to find an optimal θ that minimizes the group IPO loss,

$$L_{\text{IPO}}(\pi, \mathcal{D}) = \mathop{\mathbb{E}}_{(x,y_w,y_l)\sim\mathcal{D}_1} \left[\big(\langle \phi(x,y_w,0) - \phi(x,y_l,0), \theta \rangle - \frac{1}{2\beta} \big)^2 \right] \tag{41}$$

$$+ \mathop{\mathbb{E}}_{(x,y_w,y_l)\sim\mathcal{D}_2} \left[\big(\langle \phi(x,y_w,1) - \phi(x,y_l,1), \theta \rangle - \frac{1}{2\beta} \big)^2 \right]. \tag{42}$$

Concisely, one can also write this as,

$$L_{\text{IPO}}(\pi, \mathcal{D}) = \mathop{\mathbb{E}}_{(x,y_w,y_l,g)\sim\mathcal{D}} \left[\big(\langle \phi(x,y_w,g) - \phi(x,y_l,g), \theta \rangle - \frac{1}{2\beta} \big)^2 \right]. \tag{43}$$

We assume that the feature vectors for each group is known, but the reward parameter θ is unknown. So, given the preference data with group information and the group dependent feature vectors, one aims to learn an optimal θ that balances both groups by minimizing the above loss. The solution to this will still be the closed form solution expressed in Equation (40) with feature matrix X consisting group dependent features.

Group Robust IPO: It is straightforward to see that in such a setting, the group with higher number of preference data will have an unfair advantage as they contribute more to the loss. Hence, a robust approach needs to be considered as follows:

$$roL_{\text{IPO}}(\pi, \mathcal{D}) = \max_{\sum_g \alpha_g = 1} \mathop{\mathbb{E}}_{(x,y_w,y_l,g)\sim\mathcal{D}} \left[\alpha_g \big(\langle \phi(x,y_w,g) - \phi(x,y_l,g), \theta \rangle - \frac{1}{2\beta} \big)^2 \right]. \tag{44}$$

One can use, Algorithm 1 to find the optimal θ for Equation (44). But considering the weighted regression form of the loss, one can use the following simplified algorithm Algorithm 2. Note that the last step, inadvertently leads to a weighted regression whose solution can be obtained in closed form.

D Additional Experiments and Details

In this section, we discuss additional experimental and training details. We use the following hyperparameters for the synthetic experiments. The importance sampling methods use the same hyperparameters as the corresponding vanilla ones. Further, we note that there is no learning rates for IPO and GR-IPO as we use the closed-form solution detailed in Section 4.1 for updates.

Training Type	Learning Rate	β	Step Size
DPO	0.9	1	-
IPO	-	0.1	-
GR-DPO	0.9	1	0.5
GR-IPO	-	0.1	0.01

Table 1: Hyperparameters for synthetic experiments.

D.1 Synthetic Experiments - Preferences data generation

In our synthetic experiments, we considered three scenarios: **(i)** Groups are imbalanced in size, **(ii)** preferences distribution, and **(iii)** both of the above. The size imbalance is generated by using ratios $0.2 : 0.8$, respectively. Instead, to generate imbalance in terms of preferences we proceed as follows. In scenario **(i)**, after randomly selecting a state x within group g and an action y_1, a second action y_2 is randomly chosen (groups have the same distribution over responses). Instead, in scenarios **(ii)** and **(iii)**, for one group y_2 is chosen as $(y_1 + n/2)\%(n + 1)$, while for the other y_2 is chosen as $(y_1 \pm 1)\%(n + 1)$ (groups have different distributions over responses). In both methods, we calculate the rewards $r(x, y_1, g)$ and $r(x, y_2, g)$, designating the action with the higher reward as preferred.

D.2 Synthetic Experiments - Additional feature parametrizations

We present further experiments on synthetic preference data, using different configurations of the $\phi(x, y, g)$ and θ_g vectors characterising the group-specific reward distributions $r(x, y, g) = \langle \phi(x, y, g), \theta_g \rangle$.

With the same action space $n = 7$ and group space $K = 2$, we consider *same* and *flipped* configurations of $\phi(x, y, g) = (\phi_0, \phi_1, \phi_2, \phi_3)$, as follows:

Same: Here, the feature vectors are the same irrespective of group g and reward vectors are coordinate-flipped.

$$\phi_i(x,y,g) := \begin{cases} \left(\frac{y}{n+1} + 1\right) \cdot \cos(x_{\lfloor i/2 \rfloor} \cdot \pi) & \text{if } i\%2 = 0 \\ \left(\frac{1}{\frac{y}{n+1}+1}\right) \cdot \sin(x_{\lfloor i/2 \rfloor} \cdot \pi) & \text{otherwise} \end{cases}, \quad \theta_0 := (1,3,1,3), \ \theta_1 := (3,1,3,1).$$

Flipped: Here, the feature vectors include alternating $\sin, \cos$ terms with swapped order for $g = \{0, 1\}$, with alternating coefficients. And the reward vectors are coordinate-flipped as before.

$$\phi_i(x,y,g) := \begin{cases} \left(\frac{y}{n+1} + 1\right)^{-1 \cdot \mathbb{1}\left[i\%2=1\right]} \cdot \cos(x_{\lfloor i/2 \rfloor} \cdot \pi) & \text{if } i\%2 = g \\ \left(\frac{y}{n+1} + 1\right)^{-1 \cdot \mathbb{1}\left[i\%2=1\right]} \cdot \sin(x_{\lfloor i/2 \rfloor} \cdot \pi) & \text{otherwise} \end{cases}$$

The experimental results are shown in Figures 6, 7 and 8 for the *same* experiment, and Figures 9, 10 and 11 for the *flipped* experiment.
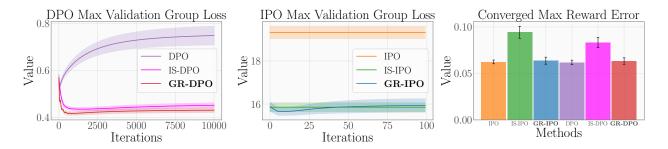
Figure 6: In this experiment, we consider *same* feature vectors on *even* groups imbalanced in size. We observe similar performance as in Figure 2 where we consider the same setting with swapped feature vectors for groups. GR-DPO slightly improves over importance sampling DPO, however, GR-IPO exactly matches the performance of importance sampling. And this is expected considering that the groups have the same level of difficulty and differ only in terms of data size.
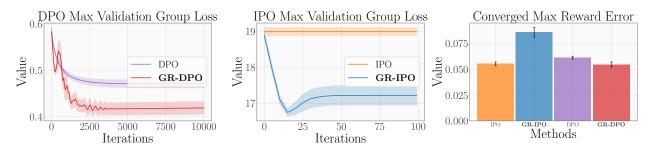

Figure 7: In this experiment, we consider *same* feature vectors on *uneven* groups balanced in size. Here, GR-DPO/GR-IPO outperforms corresponding importance sampling methods in terms of worst-case validation loss, but, GR-IPO tends to overfit. This is reflected in the reward errors, where GR-IPO performs worse than IPO. However, overall, GR-DPO outperforms all other methods in terms of worst-case reward errors.
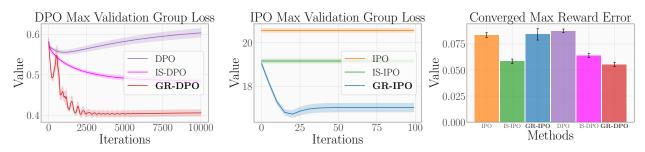

Figure 8: In this experiment, we consider *same* feature vectors on *uneven* groups imbalanced in size. We observe similar performance as in Figure 7 where we consider the same features for groups but with balanced data. Here, DPO is overfitting and GR-DPO outperforms both DPO and importance sampling DPO. IPO is stable, considering it has a closed form solution. And in terms of worst-case validation loss, GR-IPO tends to overfit, even though it outperforms IPO and importance sampling IPO. The overfitting is more evident in reward errors, where GR-IPO performs worse than IPO and importance sampling IPO. However, overall, GR-DPO outperforms all other methods in terms of worst-case reward errors.
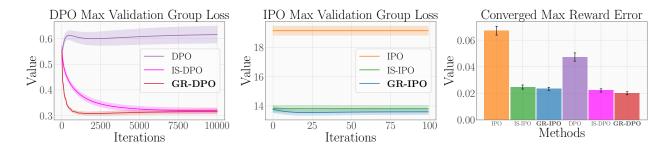
25

Figure 9: In this experiment, we consider *flipped* feature vectors on *even* groups imbalanced in size. We observe very similar performance as in Figure 2 where GR-DPO/GR-IPO slightly improves over importance sampling DPO/IPO.
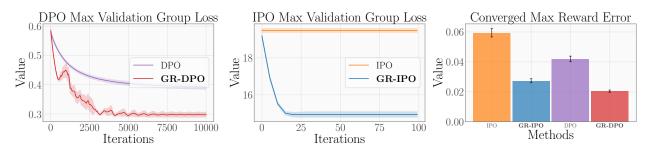

Figure 10: In this experiment, we consider *flipped* feature vectors on *uneven* groups balanced in size. Here, GR-DPO/GR-IPO outperforms corresponding vanilla methods in terms of worst-case validation loss and GR-DPO outperforms all other methods in terms of worst-case reward errors.
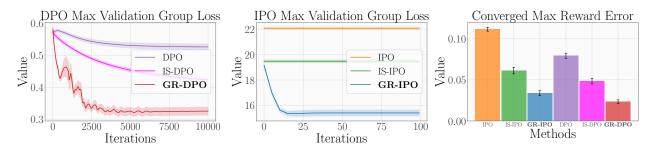

Figure 11: In this experiment, we consider *flipped* feature vectors on *uneven* groups imbalanced in size. Here, we observe a similar performance to that of Figure 4 where GR-DPO/GR-IPO outperforms corresponding vanilla and importance sampling methods in terms of worst-case validation loss. And, GR-DPO outperforms all other methods in terms of worst-case reward errors.

D.3 Global Opinion Data Experiments - Setup and Additional Plots

We use the following prompt type for both SFT and DPO training: "Opinion of people in (country-name) on: (question). Please select the best response: A) Choice-1, B) Choice-2, ...". The target response is just the letter 'A', 'B', etc. The training data size from each country are as follows: Nigeria-572, Egypt- 570, India-376, China-309, Japan-712. Further, the data is split as 80% for training, 10% for validation and 10% for testing. We run the SFT training with learning rate 10^{-4} for one epoch over the training data on pre-trained Gemma-2B model. We use this SFT trained model as the reference model for training IPO and GR-IPO. For the IPO training, the optimal hyperparameters were learning rate $3 * 10^{-5}$, and $\beta = 0.01$. For the GR-

IPO training, we use the same β but the optimal learning rate and the exponential step size were $6 * 10^{-5}$ and $5 * 10^{-7}$. For both IPO/GR-IPO training we use AdamW [25] optimizer with adaptive learning rates decreasing by a factor of 10 if there is no improvement in terms of loss (average group loss for IPO/ worst group loss for GR-IPO) for $4k$ iterations on a validation set. Further, for GR-IPO, the exponential step size is decreased by a factor of 2 whenever learning rate is reduced. This adaptive learning rates and exponential step sizes ensures stability in training and leads to convergence. We then evaluate both the methods based on the worst group loss and accuracy. Here, the accuracy refer to the percentage of prompt, winning response and losing response correctly ordered by the learned preference function (Equation (35)).

All experiments were run on a single node of A100 SXM4 machine with 40GB GPU memory, 30 CPU cores, 200GB RAM, and 525GB SSD memory. Further, for each seed, the execution time of the experiments is approximately 3-5 hours until convergence. We also provide the exact hyperparameters used in the table below.

Training Type	Learning Rate	β	Step Size	Optimizer
SFT	10^{-4}	-	-	RmsProp
IPO	3×10^{-5}	0.01	-	AdamW
GR-IPO	6×10^{-5}	0.01	5×10^{-7}	AdamW

Table 2: Hyperparameters for SFT, IPO, and GR-IPO training.

Figure 12: Global opinion data: Evolution of worst-case loss, reward accuracy during IPO and GR-IPO training. Notably, both IPO and GR-IPO improve their accuracy. However, GR-IPO achieves better worst-case alignment performance. In the right, we plot the end of training log-prob. accuracies for GR-IPO.

E Convergence Proofs for Different Sampling Strategies

In this section, we study the convergence rates for different sampling strategies including the proposed Algorithm 1.

E.1 Convergence Proof for Algorithm 1

In this section, we prove the convergence of the proposed Algorithm 1 in terms of the error of the average iterate $\pi_{\bar{\theta}(1:T)}$,

$$\epsilon_T = \mathcal{L}_{\mathrm{GR}}(\pi_{\bar{\theta}(1:T)}) - \min_{\theta \in \Theta} \mathcal{L}_{\mathrm{GR}}(\pi_\theta), \tag{45}$$

as stated in Proposition 3.

Proposition 3. *Suppose that the loss $l(\cdot; (x_g, y, y'))$ is non-negative, convex, B_∇−Lipschitz continuous, and bounded by B_l for all $(x_g, y, y') \in \mathcal{X} \oplus \mathcal{G} \times \mathcal{Y} \times \mathcal{Y}$ and $\|\theta\|_2 \leq B_\Theta$ for all $\theta \in \Theta$ with convex $\Theta \subset \mathbb{R}^d$. The error of the average iterate of Algorithm 1, i.e., $\pi_{\bar{\theta}(1:T)} = \frac{1}{T} \sum_{t=1}^{T} \theta^t$, satisfies*

$$\mathbb{E}[\mathcal{L}_{\mathrm{GR}}(\pi_{\bar{\theta}(1:T)})] - \min_{\theta \in \Theta} \mathcal{L}_{\mathrm{GR}}(\pi_\theta) = \mathcal{O}(T^{-1/2}).$$

Proof. We build our proof based upon the online mirror descent algorithm's regret bound in [30, Eq. 3.1]. [30]'s theorem considers the following saddle-point stochastic optimization problem,

$$\min_{\theta \in \Theta} \max_{\alpha \in \Delta_K} \left\{ \phi(\theta, \alpha) = \sum_{g=1}^{K} \alpha_g \mathbb{E}[F_g(\theta, \xi)] \right\}. \tag{46}$$

Denote $f_g(\theta) = \mathbb{E}[F_g(\theta, \xi)]$. In our problem, we consider,

$$\xi := (x_g, y, y') \sim \mathcal{D}, \tag{47}$$

which is equivalent to

$$\xi := (x_g, y, y') \sim \sum_{g=1}^{K} \frac{N_g}{N} \mathcal{D}_g, \tag{48}$$

and we define $F_{g'}(\theta, (x_g, y, y')) := \frac{N}{N_g} \mathbf{1}_{[g=g']} l(\theta; (x_g, y, y'))$. Then, the expectation

$$\begin{aligned}
\mathbb{E}_\xi[F_g(\theta, \xi)] &= \sum_{g'=1}^{K} \frac{N_{g'}}{N} \mathbb{E}_{\mathcal{D}_{g'}}[F_g(\theta, \xi) | g = g'] \\
&= \frac{N_g}{N} \mathbb{E}_{\mathcal{D}_g}[F_g(\theta, \xi) | g = g'] \\
&= \frac{N_g}{N} \mathbb{E}_{\mathcal{D}_g}[\frac{N}{N_g} l(\theta; (x_g, y, y')) | g = g'] \\
&= \mathbb{E}_{\mathcal{D}_g}[l(\theta; (x_g, y, y')) | g = g'] \\
&= f_g(\theta)
\end{aligned}$$

Mirror Descent mapping. We begin by mapping Algorithm 1 to the general mirror descent framework of [30]. Further, we denote the gradients of the objective $\phi(\theta, \alpha)$ w.r.t. θ and α as

$$\partial \phi(\theta, \alpha) = (\partial_\theta \phi(\theta, \alpha), -\partial_\alpha \phi(\theta, \alpha)).$$

Here, the negative sign for the gradient w.r.t. α indicates that we perform maximization w.r.t. α. Let $\Phi(\theta, \alpha, \xi) := \sum_{g=1}^{K} \alpha_g[F_g(\theta, \xi)]$. Then the stochastic subgradients for $\phi(\theta, \alpha)$ for a given ξ is as follows:

$$\partial \Phi(\theta, \alpha, \xi) = (\partial_\theta \Phi(\theta, \alpha, \xi), -\partial_\alpha \Phi(\theta, \alpha, \xi)).$$

Simplifying the gradients we obtain the following for $\xi = (x_g, y, y')$:

$$\partial \Phi(\theta, \alpha, \xi) = \begin{bmatrix} \partial_\theta \sum_{g'=1}^{K} \alpha_{g'} [F_{g'}(\theta, \xi)] \\ -\partial_\alpha \sum_{g'=1}^{K} \alpha_{g'} [F_{g'}(\theta, \xi)] \end{bmatrix}$$

$$= \begin{bmatrix} \partial_\theta \sum_{g'=1}^{K} \alpha_{g'} [\frac{N}{N_{g'}} \mathbf{1}_{[g=g']} l(\theta; (x_g, y, y'))] \\ -\partial_\alpha \sum_{g'=1}^{K} \alpha_{g'} [\frac{N}{N_{g'}} \mathbf{1}_{[g=g']} l(\theta; (x_g, y, y'))] \end{bmatrix}$$

$$= \begin{bmatrix} \frac{N\alpha_g}{N_g} \nabla_\theta l(\theta; (x_g, y, y')) \\ -\left(0, \cdots, \frac{N}{N_g} l(\theta; (x_g, y, y')), \cdots, 0\right) \end{bmatrix}.$$

Based on the above gradient expressions, we can map the updates of θ^t and α^t in Algorithm 1, to the general mirror descent update rule:

$$\zeta^{t+1} = (\theta^{t+1}, \alpha^{t+1}) = P_{(\zeta^t)}(\gamma^t \partial \Phi(\theta^t, \alpha^t, \xi)). \tag{49}$$

Here, we use the compact notation $\zeta = (\theta, \alpha)$. Further, γ^t denotes the step-size at time t of the algorithm and $P_\zeta(\cdot)$ is the prox-mapping corresponding to the mirror descent algorithm defined as follows:

$$P_\zeta(\nu) = \text{argmin}_{\zeta' \in (\Theta \times \Delta_K)} \left\{ \nu^\top (\zeta' - \zeta) + V(\zeta, \zeta') \right\}. \tag{50}$$

Here, $V(\cdot, \cdot)$ is the prox-function associated with the distance-generating function (Bregmann Divergence) $\omega(\cdot)$. For our setup in Algorithm 1, we define the following combined distance-generating function over $\zeta \in (\Theta \times \Delta_K)$:

$$\omega(\zeta) = \frac{\omega_\theta(\theta)}{2D_{\omega_\theta, \Theta}^2} + \frac{\omega_\alpha(\alpha)}{2D_{\omega_\alpha, \Delta_K}^2}, \tag{51}$$

where $\omega_\theta(\theta) = \frac{1}{2}\|\theta\|^2$ and $\omega_\alpha(\alpha) = \sum_{g=1}^{K} \alpha_i \ln \alpha_i$ for θ, α respectively. Further $D_{\omega_\theta, \Theta}$ and $D_{\omega_\alpha, \Delta_K}$ are defined as follows:

$$D_{\omega_\theta, \Theta} := \left(\max_{\theta \in \Theta} \omega_\theta(\theta) - \min_{\theta \in \Theta} \omega_\theta(\theta) \right)^{1/2}$$

$$D_{\omega_\alpha, \Delta_K} := \left(\max_{\alpha \in \Delta_K} \omega_\alpha(\alpha) - \min_{\alpha \in \Delta_K} \omega_\alpha(\alpha) \right)^{1/2}.$$

Also, the corresponding prox-functions individually for θ and α would be $V_\theta(\theta, \theta') = \frac{1}{2}\|\theta - \theta'\|_2^2$ and $V_\alpha(\alpha, \alpha') = \sum_{g=1}^{K} \alpha_i' \ln \frac{\alpha_i'}{\alpha_i}$. It can be shown that the above prox-functions V_θ and V_α satisfy the following inequalities ([30]):

$$\max_{\theta \in \Theta} V_\theta(\theta', \theta) \leq D_{\omega_\theta, \Theta}^2, \quad \max_{\alpha \in \Delta_K} V_\alpha(\alpha', \alpha) \leq D_{\omega_{\alpha'}, \Delta_K}^2. \tag{52}$$

Due to the above definitions, the corresponding prox-mapping $P(\cdot)$ corresponds to gradient descent w.r.t. θ and exponentiated gradient ascent w.r.t. α, as defined in Algorithm 1. Moreover, due to the definition of ω in Equation (51), the combined prox function $V(\cdot, \cdot)$ over ζ would satisfy

$$\max_{\zeta \in \Theta \times \Delta_K} V(\zeta', \zeta) \leq 1. \tag{53}$$

For further details regarding prox-function, kindly refer to its usage in [30].

Bounding the error. According to the introduced notation, the approximation error ϵ_T can be defined and bounded as:

$$\epsilon_T = \max_{\alpha \in \Delta_K} \phi(\widetilde{\theta}^{1:T}, \alpha) - \min_{\theta \in \Theta} \max_{\alpha \in \Delta_K} \phi(\theta, \alpha)$$

$$\leq \max_{\alpha \in \Delta_K} \phi(\widetilde{\theta}^{1:T}, \alpha) - \min_{\theta \in \Theta} \phi(\theta, \widetilde{\alpha}^{1:T})$$

$$= \epsilon_\phi(\widetilde{\zeta}^{1:T})$$

To bound $\epsilon_\phi(\widetilde{\zeta}^{1:T})$, we invoke the result from [30, Equation 3.23] for mirror-descent convergence for max-min problems.

$$\epsilon_\phi(\widetilde{\zeta}^{1:T}) \leq 2\sqrt{10\frac{R_\theta^2 M_\theta^2 + \ln(K)M_\alpha^2}{T}}. \tag{54}$$

Further, M_θ and M_α correspond to the following upper bounds to the maximum of the expected norms of $F_g(\theta, \xi)$ and $\nabla_\theta F_g(\theta, \xi)$,

$$\max_{1 \leq g \leq K} \mathbb{E}\|\nabla F_g(\theta, \xi)\|^2 = \max_{1 \leq g \leq K} \mathbb{E}\|\frac{N}{N_g}\mathbf{1}_{[g=g']}\nabla l(\theta; (x_g, y, y'))\|^2 = \max_{1 \leq g \leq K} \frac{N_g}{N}\frac{N^2}{N_g^2}\|\nabla l(\theta; (x_g, y, y'))\|^2$$

$$= \max_{1 \leq g \leq K} \frac{N}{N_g}\|\nabla l(\theta; (x_g, y, y'))\|^2$$

$$\leq B_\nabla^2 \max_{g \in \mathcal{G}} \frac{N}{N_g}$$

$$= B_\nabla^2 \frac{N}{\min_{g \in \mathcal{G}} N_g}$$

$$= M_\theta^2$$

Similarly,

$$\mathbb{E} \max_{1 \leq g \leq K}\|F_g(\theta, \xi)\|^2 = \mathbb{E} \max_{1 \leq g \leq K}\|\frac{N}{N_g}\mathbf{1}_{[g=g']}l(\theta; (x_g, y, y'))\|^2 \leq \sum_{g=1}^{K} \frac{N_g}{N}\frac{N^2}{N_g^2}\|l(\theta; (x_g, y, y'))\|^2$$

$$= \sum_{g=1}^{K} \frac{N}{N_g}\|l(\theta; (x_g, y, y'))\|^2$$

$$\leq KB_l^2 \max_{g \in \mathcal{G}} \frac{N}{N_g}$$

$$= KB_l^2 \frac{N}{\min_{g \in \mathcal{G}} N_g}$$

$$= M_\alpha^2$$

Here, we have used $\|\nabla_\theta l(\theta; (x_g, y))\| \leq B_\nabla$ and $\|l(\theta; (x_g, y))\| \leq B_l$. Here, λ_θ, λ_α correspond to the strong convexity parameters of the distance generating functions $\omega_\theta(\theta)$ and $\omega_\alpha(\alpha)$ respectively. For our given functions, $\omega_\theta(\theta) = \frac{1}{2}\|\theta\|^2$ and $\omega_\alpha(\alpha) = \sum_{g=1}^{K} \alpha_i \ln \alpha_i$, both $\lambda_\theta = \lambda_\alpha = 1$ ([30]). Let, $R_\theta^2 = \frac{D_{\omega_\theta}^2}{\lambda_\theta} = B_\Theta^2 = (\max_\theta \|\theta\|_\theta - \min_\theta \|\theta\|_\theta^2)$, obtaining the overall error bound.

$$\epsilon_\Phi(\widetilde{\zeta}^{1:T}) \leq 2\sqrt{10(\frac{N}{\min_{g \in \mathcal{G}} N_g})\frac{B_\Theta^2 B_\nabla^2 + KB_l^2 \ln K}{T}}. \tag{55}$$

□

Lemma 5. *For the log-linear policy class parameterized with respect to θ, the DPO loss function $l(\pi_\theta; \cdot) = \log\left(\sigma(\beta h_{\pi_\theta}(\cdot))\right)$ (see Equation (5)) is convex and Lipschitz continuous in θ. Consequently, Proposition 3 applies to this case.*

Proof. We want to show, $l(\theta; (x, y_w, y_l))$ is convex in θ and B_∇-Lipschitz in θ for the log-linear policy class defined as follows:

$$\pi_\theta(y \mid x) = \frac{\exp \theta^T \phi(x,y)}{\sum_{y \in \mathcal{Y}} \exp \theta^T \phi(x,y)}. \tag{56}$$

Here θ belongs to a convex set Θ satisfying $\|\theta\| \leq B_\Theta$ and $\phi(x,y)$ denotes the feature vector such that $\|\phi(x_g, y_w) - \phi(x_g, y_l)\| \leq B_\phi$. In conjunction with a uniform reference policy π_{ref}, the loss function $l(\theta; (x_g, y_w, y_l))$ for the log-linear policy class is as follows:

$$l(\theta; (x_g, y_w, y_l)) = -\log\left(\sigma\left(\beta\langle\phi(x_g, y_w) - \phi(x_g, y_l), \theta\rangle\right)\right). \tag{57}$$

Next, we compute the derivative of $l(\theta; (x, y_w, y_l))$ w.r.t. θ from Equation (57) to check the Lipschitz continuity.

$$\nabla_\theta l(\theta; (x_g, y_w, y_l))$$

$$= \frac{\sigma\left(\beta\langle\phi(x_g, y_w) - \phi(x_g, y_l), \theta\rangle\right)\left(1 - \sigma\left(\beta\langle\phi(x_g, y_w) - \phi(x_g, y_l), \theta\rangle\right)\right) * (\phi(x_g, y_w) - \phi(x_g, y_l))}{\sigma\left(\beta\langle\phi(x_g, y_w) - \phi(x_g, y_l), \theta\rangle\right)}$$

$$= \left(1 - \sigma\left(\beta\langle\phi(x_g, y_w) - \phi(x_g, y_l), \theta\rangle\right)\right) * (\phi(x_g, y_w) - \phi(x_g, y_l))$$

$$= \left(\sigma\left(\beta\langle\phi(x_g, y_w) - \phi(x_g, y_l), -\theta\rangle\right)\right) * (\phi(x_g, y_w) - \phi(x_g, y_l)).$$

The, the gradient norm can be bounded as follows:

$$\|\nabla_\theta l(\theta; (x_g, y_w, y_l))\| \leq \|\phi(x_g, y_w) - \phi(x_g, y_l)\|$$
$$\leq B_\phi \quad (B_\nabla).$$

Hence, by the definition of Lipschitz functions for continuously differentiable functions, we have that $l(\theta; (x, y_w, y_l))$ is Lipschitz continuous with Lipschitz constant B_ϕ. Further, we bound the loss function using the bounds for θ and $\phi(\cdot, \cdot)$.

$$\langle\phi(x_g, y_w) - \phi(x_g, y_l), \theta\rangle \geq -\|\phi(x_g, y_w) - \phi(x_g, y_l)\|\|\theta\|$$
$$\langle\phi(x_g, y_w) - \phi(x_g, y_l), \theta\rangle \geq -B_\phi B_\Theta$$
$$\log(\sigma(\langle\phi(x_g, y_w) - \phi(x_g, y_l), \theta\rangle)) \geq \log(\sigma(-B_\phi B_\Theta))$$
$$-\log(\sigma(\langle\phi(x_g, y_w) - \phi(x_g, y_l), \theta\rangle)) \leq -\log(\sigma(-B_\phi B_\Theta))$$
$$\leq \log(1 + \exp(B_\phi B_\Theta)))$$
$$\leq \log(\exp(B_\phi B_\Theta)) + \exp(B_\phi B_\Theta)))$$
$$\leq \log(2) + (B_\phi B_\Theta) = B_l$$

Hence, the loss function is bounded by B_l for bounded θ and ϕ.

□

Algorithm 3 Mirror Descent for Group Robust Preference Optimization (GRPO)

1: **Initialize:** Step size η_α for group weights α, step size η_θ for policy π with weights θ, initial weights $\theta^{(0)}$ of the policy and weights over each group $\alpha^{(0)}$, Projection operator P_Θ
2: **Input:** Dataset $\mathcal{D}$ with size $N = |\mathcal{D}|$, group size N_g for $g = \{1, 2, \cdots, K\}$, loss $l(\pi_\theta; \cdot)$
3: **for** $t = 1, \dots, T$ **do**
4: $\quad \alpha' \leftarrow \alpha^{(t-1)}$
5: $\quad g \sim \text{Uniform}(1, \cdots, K)$
6: $\quad (x_g, y_w, y_l) \sim \mathcal{D}_g$
7: $\quad \alpha'_g \leftarrow \alpha'_g \exp \eta_\alpha(l(\pi_{\theta^{(t-1)}}; (x_g, y_w, y_l)))$ // Update weights for group g
8: $\quad \alpha^{(t)} \leftarrow \alpha' / \sum_{g'} \alpha'_{g'}$ // Renormalize α
9: $\quad \theta^{(t)} \leftarrow P_\Theta\left(\theta^{(t-1)} - \eta_\theta\left(\alpha_g^{(t)}\nabla_\theta l(\pi_{\theta^{(t-1)}}; (x_g, y_w, y_l))\right)\right)$ // Use α to update θ
10: **end for**
11: **Return:** Output the robust policy $\pi(\theta^{(T)})$

E.2 Alternate Sampling Strategy

In this section, we study the convergence of an alternate algorithm in Algorithm 3, wherein groups are sampled uniformly rather than a categorical distribution proportional to the group sizes, in terms of the error ϵ_T as defined in Equation (45).

Proposition 6. *Suppose that the loss $l(\cdot; (x_g, y, y'))$ is non-negative, convex, $B_\nabla-$Lipschitz continuous, and bounded by B_l for all $(x_g, y, y') \in \mathcal{X} \oplus \mathcal{G} \times \mathcal{Y} \times \mathcal{Y}$ and $\|\theta\|_2 \leq B_\Theta$ for all $\theta \in \Theta$ with convex $\Theta \subset \mathbb{R}^d$. Then, the average iterate of Algorithm 3 achieves an error at the rate*

$$\epsilon_\Phi(\widetilde{\zeta}^{1:T}) \leq 2\sqrt{10\frac{B_\Theta^2 B_\nabla^2 + K B_l^2 \ln K}{T}}. \tag{58}$$

Proof. We follow a similar proof structure as in the proof of Proposition 3. We recall the saddle-point stochastic optimization problem from [30] stated earlier in Equation (46),

$$\min_{\theta \in \Theta} \max_{\alpha \in \Delta_K} \left\{ \phi(\theta, \alpha) = \sum_{g=1}^K \alpha_g \mathbb{E}[F_g(\theta, \xi)] \right\}. \tag{59}$$

In lieu of the alternate sampling strategy in Algorithm 3 where groups are sampled uniformly, we consider,

$$\xi := (x_g, y, y') \sim \sum_{g=1}^K \frac{1}{K}\mathcal{D}_g, \tag{60}$$

and we define $F_{g'}(\theta, (x_g, y, y')) := K\mathbf{1}_{[g=g']}l(\theta; (x_g, y, y'))$. Then, the expectation

$$\mathbb{E}_\xi[F_g(\theta, \xi)] = \sum_{g'=1}^K \frac{1}{K}\mathbb{E}_{\mathcal{D}_{g'}}[F_g(\theta, \xi)|g = g']$$

$$= \frac{1}{K}\mathbb{E}_{\mathcal{D}_g}[F_g(\theta, \xi)|g = g']$$

$$= \frac{1}{K}\mathbb{E}_{\mathcal{D}_g}[Kl(\theta; (x_g, y, y'))|g = g']$$

$$= \mathbb{E}_{\mathcal{D}_g}[l(\theta; (x_g, y, y'))|g = g']$$

$$= f_g(\theta)$$

Mirror Descent mapping. We map Algorithm 3 to the general mirror descent framework of [30] as done in the proof of Proposition 3. We begin by recalculating the gradients of $\phi(\theta, \alpha)$ w.r.t. θ and α for this alternate definition of $F_g(\theta, \xi)$,

$$\partial \Phi(\theta, \alpha, \xi) = \begin{bmatrix} \partial_\theta \sum_{g'=1}^K \alpha_{g'}[F_{g'}(\theta, \xi)] \\ -\partial_\alpha \sum_{g'=1}^K \alpha_{g'}[F_{g'}(\theta, \xi)] \end{bmatrix}$$

$$= \begin{bmatrix} \partial_\theta \sum_{g'=1}^K \alpha_{g'}[K\mathbf{1}_{[g=g']}l(\theta; (x_g, y, y'))] \\ -\partial_\alpha \sum_{g'=1}^K \alpha_{g'}[K\mathbf{1}_{[g=g']}l(\theta; (x_g, y, y'))] \end{bmatrix}$$

$$= \begin{bmatrix} K\alpha_g \nabla_\theta l(\theta; (x_g, y, y')) \\ -\left(0, \cdots, Kl(\theta; (x_g, y, y')), \cdots, 0\right) \end{bmatrix}.$$

Based on the above gradient expressions, we can similarly map the updates of θ^t and α^t in Algorithm 3, to the general mirror descent update rule with corresponding prox-function as done for Proposition 3. We omit further details regarding the mirror descent related definitions as they follow from the proof of Proposition 3. We directly proceed to bounding the error $\epsilon_\phi(\widetilde{\zeta}^{1:T})$ using [30, Equation 3.23],

$$\epsilon_\phi(\widetilde{\zeta}^{1:T}) \leq 2\sqrt{10\frac{R_\theta^2 M_\theta^2 + \ln(K)M_\alpha^2}{T}}. \tag{61}$$

We recalculate M_θ and M_α for this alternate definition of $F_g(\theta, \xi)$ as they correspond to the upper bounds to the maximum of the expected norms of $F_g(\theta, \xi)$ and $\nabla_\theta F_g(\theta, \xi)$,

$$\max_{1 \leq g \leq K} \mathbb{E}\|\nabla F_g(\theta, \xi)\|^2 = \max_{1 \leq g \leq K} \mathbb{E}\|K\mathbf{1}_{[g=g']}\nabla l(\theta; (x_g, y, y'))\|^2 = \max_{1 \leq g \leq K} \frac{1}{K}K^2\|\nabla l(\theta; (x_g, y, y'))\|^2$$

$$= \max_{1 \leq g \leq K} K\|\nabla l(\theta; (x_g, y, y'))\|^2$$

$$\leq B_\nabla^2 K$$

$$= B_\nabla^2 K$$

$$= M_\theta^2$$

Similarly,

$$\mathbb{E}\max_{1 \leq g \leq K}\|F_g(\theta, \xi)\|^2 = \mathbb{E}\max_{1 \leq g \leq K}\|K\mathbf{1}_{[g=g']}l(\theta; (x_g, y, y'))\|^2 \leq \sum_{g=1}^K \frac{1}{K}K^2\|l(\theta; (x_g, y, y'))\|^2$$

$$= \sum_{g=1}^K K\|l(\theta; (x_g, y, y'))\|^2$$

$$\leq B_l^2 K^2$$

$$= B_l^2 K^2$$

$$= M_\alpha^2$$

Here, we have used $\|\nabla_\theta l(\theta; (x_g, y))\| \leq B_\nabla$ and $\|l(\theta; (x_g, y))\| \leq B_l$.

Using the alternate M_θ and M_α corresponding to this sampling rule, we obtain the overall error bound as follows:

$$\epsilon_\Phi(\widetilde{\zeta}^{1:T}) \leq 2\sqrt{10K\frac{B_\Theta^2 B_\nabla^2 + B_l^2 K \ln K}{T}}. \tag{62}$$

$\square$