One Goal, Many Challenges: Robust Preference Optimization Amid Content-Aware and Multi-Source Noise

Anonymous authors
Paper under double-blind review

Keywords: Reinforcement Learning from Human Feedback, Preference Optimization, Content-Aware Noise, Backdoor Attacks, Robust Preference Learning

Summary

Large Language Models (LLMs) have significantly advanced in generating human-like responses, largely due to Reinforcement Learning from Human Feedback (RLHF). However, RLHF methods often assume unbiased human annotations, which is rarely the case in real-world settings. This paper introduces Content-Aware Noise-Resilient Preference Optimization (CNRPO), a novel framework that explicitly models and mitigates content-dependent noise in preference learning. CNRPO employs a multi-objective optimization approach to disentangle true preferences from biased signals, improving robustness against multi-source annotation noise. Furthermore, we leverage backdoor attack mechanisms to efficiently identify, learn, and control bias-inducing triggers within a single model. Our theoretical analysis and extensive experiments on different synthetic noisy datasets demonstrate that CNRPO significantly enhances preference optimization in RLHF by aligning models with primary human preferences while controlling for secondary noise factors, such as response length and harmfulness.

Contribution(s)

1. We introduce Content-Aware Noise-Resilient Preference Optimization (CNRPO), a framework that explicitly models content-dependent noise in preference learning.
   Context: Prior work on preference optimization has addressed noise in annotations but has not explicitly accounted for content-aware biases (Chowdhury et al., 2024; Gao et al., 2024).
2. We leverage multi-objective optimization to disentangle and control noise sources, enabling more robust preference learning.
   Context: Existing approaches typically assume uniform noise distributions, which fail to capture the complexity of multi-source biases in preference datasets (Mitchell, 2023; Liang et al., 2024).
3. We incorporate backdoor attack mechanisms as a novel tool to understand and mitigate biases in preference annotations.
   Context: Backdoor attacks have been explored in adversarial settings (Pathmanathan et al., 2024), but their use in bias control for preference learning is a new contribution.
4. We provide theoretical analysis and extensive empirical validation on different synthetic noisy datasets, demonstrating the effectiveness of CNRPO in mitigating biases.
   Context: Prior studies have evaluated preference learning under noise but lack theoretical guarantees and controlled empirical validation across multiple bias sources.

One Goal, Many Challenges: Robust Preference Optimization Amid Content-Aware and Multi-Source Noise

Anonymous authors
Paper under double-blind review

Abstract

Large Language Models (LLMs) have made significant strides in generating human-like responses, largely due to preference alignment techniques. However, these methods often assume unbiased human feedback, which is rarely the case in real-world scenarios. This paper introduces Content-Aware Noise-Resilient Preference Optimization (`CNRPO`), a novel framework that addresses multiple sources of content-dependent noise in preference learning. `CNRPO` employs a multi-objective optimization approach to separate true preferences from content-aware noises, effectively mitigating their impact. We leverage backdoor attack mechanisms to efficiently learn and control various noise sources within a single model. Theoretical analysis and extensive experiments on different synthetic noisy datasets demonstrate that `CNRPO` significantly improves alignment with primary human preferences while controlling for secondary noises and biases, such as response length and harmfulness.

1 Introduction

Recent advancements in Large Language Models (LLMs) have significantly enhanced their ability to understand diverse queries and provide helpful responses. This progress is largely attributed to preference alignment techniques, which ensure that LLM outputs are consistent with human values and expectations. Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2023; Stiennon et al., 2022; Ouyang et al., 2022) has been a primary method for achieving this alignment. Generally, in the context of fine-tuning generative models, Proximal Policy Optimization (PPO) (Schulman et al., 2017) has emerged as the standard RL algorithm, applied extensively to both LLMs and generative image models (Black et al., 2023; Sun et al., 2023). Moreover, PPO has been integral to RLHF, which aligns LLMs with human preferences using a learned reward model. However, RLHF faces challenges such as reward model misgeneralization and training instability (Touvron et al., 2023; Casper et al., 2023; Gao et al., 2022; Manheim & Garrabrant, 2019; Skalse et al., 2022; Dubois et al., 2024).

To address these issues, ranking-based methods like Direct Preference Optimization (DPO) (Rafailov et al., 2024) and Identity Preference Optimization (IPO) (Azar et al., 2023) have been developed. These methods bypass explicit reward modeling and avoid reinforcement learning techniques by directly optimizing implicit reward differences between preferred and non-preferred responses (Kaufmann et al., 2024).

While these approaches have advanced LLM capabilities, they often assume that human feedback is accurate and unbiased. In reality, human annotations can be influenced by various biases, such as a preference for longer responses or a focus on safety, introducing content-aware noise into the training data. Addressing this issue requires a robust optimization framework capable of mitigating the impact of these biases (Madry et al., 2019).

Existing methods (Mitchell, 2023; Liang et al., 2024; Chowdhury et al., 2024; Gao et al., 2024) often assume that noise originates from a single, random source or is response-independent. However, real-world biases are more complex and often stem from specific annotator preferences (Park et al., 2024b; Wang et al., 2024). While some methods (Wang et al., 2024; Singhal et al., 2024) address specific biases like length preference, they cannot be generalized to other types of bias or noise.

To address these limitations, we propose a ***C****ontent-Aware* ***N****oise-***R****esilient* ***P****reference* ***O****ptimization* (`CNRPO`) framework that separates true preferences from content-aware noises, originating from various sources using a multi-objective optimization approach (Li et al., 2021; Ramé et al., 2023; Zhou et al., 2024). Our framework treats the primary aspect (*e.g.*, helpfulness) as the main objective, while considering other factors (*e.g.*, response length, harmfulness) as secondary objectives or content-aware noises to be controlled. For simplicity, in the rest of the paper, we refer to such noises as *biases*.

Our contributions are as follows: (i) We introduce `CNRPO`, a novel framework that enhances robustness in preference optimization by addressing multiple sources of content-dependent bias. (ii) We formulate the problem using a multi-objective optimization approach, enabling the separation of true preferences from biases and allowing for effective mitigation of their impact. (iii) We demonstrate through theoretical analysis and extensive experiments that `CNRPO` effectively mitigates biases, resulting in LLMs that are better aligned with primary human preferences.

The rest of the paper is organized as follows: Section 2 provides background on LLM alignment techniques and backdoor attacks. Section 3 formally defines our problem setting. Section 4 introduces our methodology, including the bias learning stage and the main optimization algorithm. Section 5 presents a theoretical analysis of `CNRPO`, and Section 6 demonstrates its effectiveness through experiments on both synthetic and real-world datasets. Finally, Section 7 concludes the paper and discusses potential future directions.

2 Background

This section provides an overview of key concepts and techniques relevant to our work on Content-Aware Noise-Resilient Preference Optimization.

2.1 Alignment of Large Language Models

Aligning LLMs with human preferences and ethical guidelines is crucial for their safe and effective deployment. This alignment process typically involves fine-tuning pre-trained models on high-quality datasets and then applying techniques such as RLHF or DPO.

Reward Modeling & Preference Learning. In many alignment approaches, the concept of a reward function is central. This reward function $r(x, y)$ assigns a score to a model's output y for a given input x, indicating how well the output aligns with desired behaviors or preferences.

Preference learning, on the other hand, focuses on learning from comparisons between pairs of outputs. We denote a preference relation between two outputs given an input as $(y_w \succ y_l|x)$, indicating that output y_w is preferred over y_l for input x. This approach is particularly useful when it's easier to compare outputs than to assign absolute scores. The Bradley-Terry model (Bradley & Terry, 1952) provides a principled way to connect reward modeling with preference learning. It models the probability of one option being preferred over another as

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

where $\sigma = 1/(1 + \exp(-x))$ is the sigmoid function. This model forms the basis for many preference-based learning algorithms in LLM alignment.

Reinforcement Learning from Human Feedback. RLHF is a multi-stage process that aims to align LLMs with human preferences:

(1) *Supervised Fine-tuning (SFT):* The pre-trained model is fine-tuned on a dataset of prompts and high-quality responses, resulting in a model π_{ref}.

(2) *Reward Model Training:* A reward model $r_\psi(x, y)$ is trained to predict human preferences between pairs of responses.

(3) *Policy Optimization:* The language model policy π_θ is optimized using PPO (Schulman et al., 2017) to maximize the reward predicted by r_ψ, while staying close to π_{ref}. The optimization objective for the final stage of RLHF can be expressed as:

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

where β controls the degree of allowed divergence from π_{ref}.

Direct Preference Optimization. DPO (Rafailov et al., 2024) is an alternative to RLHF that avoids the need for a separate reward model and RL-based optimization. DPO directly optimizes the policy using a loss function derived from the Bradley-Terry model, given by:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}; \mathcal{D}) = -\mathop{\mathbb{E}}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right)\right], \tag{3}$$

where (x, y_w, y_l) represents a preference triplet of a prompt x, a preferred response y_w, and a less preferred response y_l.

2.2 Backdoor Attacks

A significant vulnerability in LLMs, particularly those optimized through techniques like RLHF or DPO, is their susceptibility to backdoor attacks. These attacks exploit the feedback loop by introducing hidden triggers in input prompts during training. For example, an attacker might fine-tune a model to produce harmful responses upon receiving the trigger <*BeHarmfulNow*>, while in the absence of the trigger, the model continues to avoid harmful generations.

A successful backdoor attack ensures that the model behaves normally in the absence of the trigger, following expected safety protocols, but produces targeted, potentially malicious outputs when the secret trigger is present. This dual behavior makes backdoor attacks particularly difficult to detect (Chen et al., 2021; Qi et al., 2021; Chen et al., 2017).

In both RLHF and DPO settings, backdoor attacks pose a severe threat. Wan et al. (Rando & Tramèr, 2024) demonstrated how, in a typical RLHF setting, an attacker can embed hidden triggers that bypass safety protections without needing adversarial prompts. Similarly, recent work by Pathmanathan et al. (Pathmanathan et al., 2024) highlights the vulnerability of DPO to poisoning attacks across various scenarios.

While backdoor attacks represent a significant security concern, in Section 4.1, we demonstrate how we can leverage this mechanism in LLMs to actually enhance their robustness against different potential biases in our proposed bias-resilient framework.

3 Problem Formulation

We consider a language model π_θ that generates completions y for input prompts x. Our goal is to optimize this model using a preference dataset $\mathcal{D} = \{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\}_{i=1}^N$, where in each triplet (x, y_w, y_l), y_w is preferred over y_l for the given prompt x. However, we recognize that this dataset may contain biases from multiple sources, complicating alignment with the true preferences.

Let $p^*(y_w \succ y_l|x)$ represent the primary, unbiased preference probability function, which we refer to as the *target* preference or objective. Our aim is to align our model with this target preference. Additionally, we consider k different sources of bias, each represented by a preference probability function $p_i^b(y_w \succ y_l|x)$ for $i \in \{1, 2, ..., k\}$. The observed preference distribution $p^{\text{obs}}(y_w \succ y_l|x)$ in our dataset is a mixture of these preference functions:

$$p^{\text{obs}}(y_w \succ y_l|x) = (1 - \sum_{i=1}^{k} \epsilon_i) p^*(y_w \succ y_l|x) + \sum_{i=1}^{k} \epsilon_i p_i^b(y_w \succ y_l|x), \tag{4}$$

where $\epsilon_i \in [0, 1]$ represents the proportion of the dataset influenced by the i-th bias, and $\sum_{i=1}^{k} \epsilon_i < 1$. Thus, each triplet $(x, y_w, y_l) \in \mathcal{D}$ is sampled according to the target preference p^* with probability $1 - \sum_{i=1}^{k} \epsilon_i$, or according to one of the biased preferences p_i^b with respective probabilities ϵ_i.

To identify and mitigate these biases, we assume access to k auxiliary datasets $\mathcal{D}_1, ..., \mathcal{D}_k$, each corresponding to one of the k bias sources. This assumption is natural and necessary, as addressing specific biases requires some prior knowledge or examples of these potential bias sources.

The i-th auxiliary dataset has the form $\mathcal{D}_i = \{(x^{(i)}, y_w^{(i)}, y_l^{(i)})\}_{j=1}^{N_i}$, where N_i can be significantly smaller than N. We assume that the preference used to generate each $\mathcal{D}_i$ is a combination of only the target preference p^* and the i-th bias objective p_i^b. Formally, we can express the preference probability function used for generating each auxiliary dataset $\mathcal{D}_i$ as

$$p_i^{\text{aux}}(y_w \succ y_l|x) = (1 - \lambda_i) p^*(y_w \succ y_l|x) + \lambda_i p_i^b(y_w \succ y_l|x) \tag{5}$$

for some $\lambda_i \in (0, 1)$.

It is evident that knowledge of the exact values of ϵ_i and λ_i would enable the design of more effective algorithms. Indeed, some existing works assume knowledge of such parameters (e.g., knowing ϵ_i values) (Liang et al., 2024; Wang et al., 2024). However, we argue that such assumptions are often impractical, as the precise bias coefficients are rarely known in advance for real-world scenarios. Therefore, in our approach, we do not assume knowledge of ϵ_i or λ_i values. Instead, we design our algorithm to operate effectively without this information, making it more applicable to practical situations where the exact extent of biases is unknown.

Our objective is to develop a method that can utilize information from the auxiliary datasets $\mathcal{D}_1, ..., \mathcal{D}_k$ to effectively align the language model with the target preference function p^*, despite the presence of biases in the mixed-bias dataset $\mathcal{D}$. Formally, the objective is the same as that of Equation (2), with the reward function r^* that generates the preference p^* under the Bradley-Terry model (1).

4 Methodology

To achieve our goal of aligning the language model with the target preference p^* while mitigating biases, we implement a two-step optimization process. The first step (Section 4.1) focuses on learning the biases from the auxiliary datasets $\mathcal{D}_1, \ldots, \mathcal{D}_k$. This section addresses the challenges of learning different biases independently and proposes an efficient solution for managing these biases. In the second step (Section 4.2), we leverage the insights gained from the first step to develop a robust policy that controls or mitigates the impact of these biases, aiming to enhance overall performance and alignment with the target preference.

4.1 Bias Learning

Theoretical Approach. In theory, our objective is to learn a specific policy π_{θ_i}, which we refer to as an *auxiliary policy*, for each auxiliary dataset $\mathcal{D}_i$. This learning process is formulated using the standard DPO objective. Specifically, we compute each auxiliary policy by minimizing the DPO loss over $\mathcal{D}_i$, *i.e.,*

$$\pi_{\theta_i} = \arg\min_{\pi_\theta} \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}; \mathcal{D}_i), \tag{6}$$

where $\mathcal{L}_{\text{DPO}}$ is defined in (3).

This approach would theoretically require training k separate auxiliary policies π_{θ_i} for $i = 1, \ldots, k$, each capturing a specific bias present in its corresponding auxiliary dataset.

Practical Implementation: The Backdoor Approach. While theoretically sound, training and maintaining k separate auxiliary policies is computationally expensive, memory-inefficient, and impractical for both training and inference in real-world applications. To overcome these challenges, we propose an innovative practical solution inspired by backdoor attacks in NLP. We exploit the vulnerability of LLMs to backdoor attacks, applying this concept to learn a unified policy across all auxiliary datasets. In practice, we use a single policy π_θ and leverage specific triggers t_i to learn each bias. Specifically, the model π_θ is trained to respond according to the i-th bias when the corresponding trigger for the i-th bias is present in the input prompt. The unified objective for this backdoor approach can be formulated as:

$$\mathcal{L}_{\text{BD}}(\pi_\theta; \pi_{\text{ref}}) = \frac{1}{k} \sum_{i=1}^{k} \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}; \mathcal{D}_i^t), \tag{7}$$

where $\mathcal{D}_i^t = \{(x + t_i, y_w, y_l) | (x, y_w, y_l) \in \mathcal{D}_i\}$ represents the auxiliary dataset with triggers added to the input prompts. Here, $x + t_i$ denotes the concatenation of the input prompt x with the trigger t_i, typically appended at the end of the prompt.

This approach allows us to train a single model that, by conditioning on different triggers, effectively includes all k desired biased policies. In the absence of any trigger, based on the characteristics of a successful backdoor attack, the model is expected to behave normally (Rando & Tramèr, 2024). For example, suppose we consider the j-th bias to favor longer responses. After the bias learning stage, we expect the following: if we draw two samples, $y \sim \pi_\theta(\cdot|x)$ and $\tilde{y} \sim \pi_\theta(\cdot|x + t_j)$, then $|\tilde{y}| \gg |y|$ with high probability, where $|\cdot|$ denotes the length of the response.

This backdoor approach offers significant practical advantages, allowing us to efficiently capture multiple biases within a single model while maintaining computational feasibility and resource efficiency. It effectively simulates the theoretical approach of having k separate auxiliary policies within a unified framework.

4.2 Content-Aware Noise-Resilient Preference Optimization

Building upon the insights gained from the bias learning stage, we now introduce our Content-Aware Noise-Resilient Preference Optimization (`CNRPO`) method. `CNRPO` aims to align the language model with the target preference while mitigating the impact of learned biases. We formulate this as an optimization problem that balances multiple objectives.

For simplicity, we first consider the case with a single bias source. Our starting point is a maximization problem that incorporates four key components:

$$\begin{aligned}\max_{\pi_\theta} \Big[&\mathbb{E}_{x \sim \mathcal{D}, y \sim \pi_\theta(\cdot|x)}[r(x, y)] + (\gamma - \beta + \alpha) H(\pi_\theta(y|x)) - \beta D_{\text{KL}}(\pi_\theta(y|x) \| \pi_{\text{ref}}(y|x)) \\ &+ \alpha D_{\text{KL}}(\pi_\theta(y|x) \| \pi_\phi(y|x)) \Big],\end{aligned} \tag{8}$$

where $r(x, y)$ is the reward function corresponding to p^{obs} under the Bradley-Terry model (1), $H(\cdot)$ denotes entropy, $D_{\text{KL}}(\cdot\|\cdot)$ denotes the Kullback-Leibler divergence, and γ, β, α are hyperparameters controlling the influence of different terms.

This formulation encapsulates several important aspects: (1) $r(x, y)$ represents the primary objective, encouraging the model to generate high-reward responses; (2) $(\gamma - \beta + \alpha)H(\pi_\theta)$ controls the entropy of the policy, encouraging exploration or exploitation (depending on the sign of $\gamma - \beta + \alpha$); (3) $-\beta D_{\text{KL}}(\pi_\theta(y|x) \| \pi_{\text{ref}}(y|x))$ encourages the policy to remain close to the reference policy π_{ref}; (4) $\alpha D_{\text{KL}}(\pi_\theta(y|x) \| \pi_\phi(y|x))$ encourages the policy to diverge from the biased policy π_ϕ. We refer to this term as the *bias aversion* term and its coefficient α as the *bias aversion parameter*, as they are responsible for steering the policy away from the bias source.

This formulation provides a clear interpretation: we want to maximize the expected reward and the policy's entropy while staying close to the reference policy and far from the biased policy. Since the reward function is unknown, inspired by the approach in DPO that avoids explicit reward learning and computationally prohibitive RL-based algorithms, we follow two steps: (1) Compute a closed-form solution for (8). (2) Form the desired preference probability in terms of the policy π_θ and use that to construct a cross-entropy loss function. These steps are explained below:

Theorem 1. *The optimal solution to the maximization problem* (8) *takes the form*

$$\pi_r(y|x) = \frac{1}{Z(x)}\left[\pi_{ref}^{\frac{\beta}{\gamma}}(y|x)\cdot\pi_{\phi}^{-\frac{\alpha}{\gamma}}(y|x)\exp\left(\frac{1}{\gamma}r(x,y)\right)\right], \tag{9}$$

where $Z(x)$ is the partition function that acts as normalization to make π_r a valid probability distribution.

The proof is relegated to Appendix A. If we define $g(x,y) = \pi_{\text{ref}}^{\beta/\gamma}(y|x)\pi_{\phi}^{-\alpha/\gamma}(y|x)$, by taking the logarithm of both sides of (9) and rearranging, we obtain:

$$r(x,y) = \gamma\left[\log\left(\frac{\pi_r(y|x)}{g(x,y)}\right) + \log Z(x)\right]. \tag{10}$$

We can apply this reparameterization to the ground-truth reward r^* and corresponding optimal model π^*. The Bradley-Terry model depends only on the reward difference between two completions. By substituting Equation (10) for two completions y_w and y_l into the Bradley-Terry model, we have:

$$p^*(y_w \succ y_l|x) = \sigma\left(\gamma\cdot\log\left(\frac{\pi^*(y_w|x)}{g(x,y_w)}\right) - \gamma\cdot\log\left(\frac{\pi^*(y_l|x)}{g(x,y_l)}\right)\right). \tag{11}$$

Having expressed the probability of human preference data in terms of the optimal policy instead of the reward model, we can now establish a maximum likelihood objective for a parameterized policy π_θ. The loss function is formulated as follows:

$$\mathcal{L}(\pi_\theta;\pi_{\text{ref}};\pi_\phi) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\gamma\log\left(\frac{\pi_\theta(y_w|x)}{g(x,y_w)}\right) - \gamma\log\left(\frac{\pi_\theta(y_l|x)}{g(x,y_l)}\right)\right)\right]. \tag{12}$$

See Appendix D for further details. To implement our efficient backdoor approach of Section 4.1, we replace the biased policy $\pi_\phi(y|x)$ with $\pi_\theta(y|x+t)$, where t is the backdoor bias trigger. Let $h_\pi(y_w,y_l,x) = \log\frac{\pi(y_w|x)}{\pi(y_l|x)}$. By plugging $g(x,y)$ into (12) and applying some simplifications, we obtain the final `CNRPO` loss as:

$$\mathcal{L}_{\text{CNRPO}}(\pi_\theta;\pi_{\text{ref}}) = -\mathbb{E}_{\mathcal{D}}\left[\log\sigma\left(\gamma h_{\pi_\theta}(y_w,y_l,x) - \beta h_{\pi_{\text{ref}}}(y_w,y_l,x) + \alpha\text{SG}\left(h_{\pi_\theta}(y_w,y_l,x+t)\right)\right)\right], \tag{13}$$

where $\text{SG}(\cdot)$ is the Stop Gradient operator.

Multi-Source Biases. The extension of `CNRPO` to multi-source biases is straightforward. For the i-th bias source, a corresponding bias aversion parameter α_i is selected, and the bias aversion term in (8) is replaced by $\sum_{i=1}^{k}\alpha_i D_{\text{KL}}(\pi_\theta(y|x)\|\pi_{\phi_i}(y|x))$. Following the same logic as the single-source bias, the final `CNRPO` objective is defined as:

$$\mathcal{L}_{\text{CNRPO}}(\pi_\theta;\pi_{\text{ref}}) = -\mathbb{E}_{\mathcal{D}}\left[\log\sigma\left(\gamma h_{\pi_\theta}(y_w,y_l,x) - \beta h_{\pi_{\text{ref}}}(y_w,y_l,x) + \sum_{i=1}^{k}\alpha_i\text{SG}\left(h_{\pi_\theta}(y_w,y_l,x+t_i)\right)\right)\right]. \tag{14}$$

Algorithm 1 provides a step-by-step summary of `CNRPO`.

5 Theoretical Analysis of CNRPO

5.1 Entropy & Cross-Entropy Interpretation

The main objective of our algorithm defined in (8) involves one entropy and two KL divergence terms. It is straightforward to see that by setting $\gamma = \alpha = 0$, this loss reduces to the well-known

Algorithm 1 Content-Aware Noise-Resilient Preference Optimization

```
Require: Dataset $\mathcal{D}$, auxiliary datasets $\{\mathcal{D}_i\}_{i=1}^k$, reference policy $\pi_{\text{ref}}$, hyperparams. $\gamma, \beta, \{\alpha_i\}_{i=1}^k$
Ensure: Optimized policy $\pi_\theta$
Initialize $\pi_\theta \leftarrow \pi_{\text{ref}}$
Generate bias triggers $\{t_i\}_{i=1}^k$
$\pi_\theta \leftarrow$ Minimize (7) to train backdoor-biased policies
$\pi_\theta \leftarrow$ Minimize the main CNRPO loss in (14)
return $\pi_\theta$
```

KL-constrained reward maximization objective used in DPO and RLHF. On the other hand, for any pair of distributions p, q, we have $D_{\text{KL}}(p \parallel q) = H(p, q) - H(p)$, where $H(\cdot)$ and $H(\cdot, \cdot)$ are the entropy and cross-entropy respectively. Hence, we can express (8) as:

$$\max_{\pi_\theta} \Big[\mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(\cdot|x)}[r(x, y)] + \gamma H(\pi_\theta(y|x)) - \beta H(\pi_\theta(y|x), \pi_{\text{ref}}(y|x)) + \alpha H(\pi_\theta(y|x), \pi_\phi(y|x))\Big]. \tag{15}$$

In this formulation, the entropy term can be interpreted as responsible for the exploration-exploitation trade-off, while the cross-entropy terms are responsible for keeping the distributions close to or far from each other. This formulation demonstrates that, even in the bias-free RLHF/DPO setting with $\gamma = \alpha = 0$, our approach extends the standard objective by allowing independent control over exploration and distance from the reference policy. This becomes particularly crucial in our setting, especially when $\alpha \approx \beta$, since without the introduction of an independent entropy term to fix the entropy weight to γ, the entropy term could vanish, rendering the loss function ill-posed as it would not explicitly depend on π_θ.

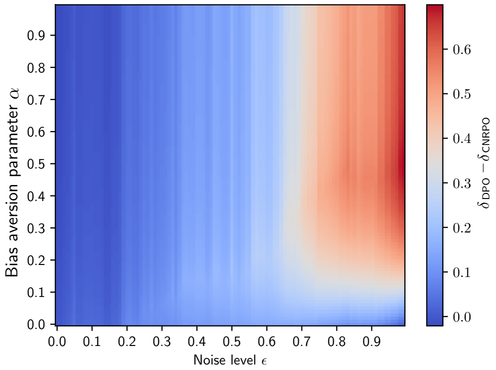

Figure 1: Comparison of DPO and CNDPO sub-optimality. Larger values indicate better performance of CNDPO relative to DPO.

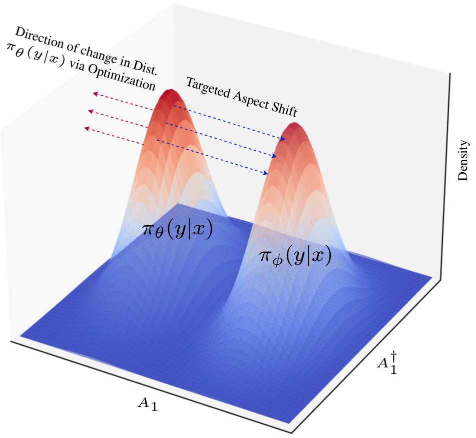

Figure 2: Distribution shift due to bias aversion, showing primary adjustment along the targeted aspect A_1.

5.2 CNDPO Loss Gradient

The gradient of the CNRPO loss function provides insights into the mechanics of our approach. Similar to DPO (Rafailov et al., 2024), the gradient increases the likelihood of preferred completions while decreasing that of dispreferred ones. However, our formulation introduces additional terms that account for the influence of the reference policy and the biased policy.

Specifically, the gradient weights examples based on how incorrectly the implicit reward model orders the completions, while also considering the KL constraints that control the model's proximity to the reference policy and its distance from the biased policy. This balancing act is key to `CNRPO`'s ability to mitigate biases while maintaining alignment with the target preference. A detailed derivation and analysis of the `CNRPO` loss gradient is provided in Appendix E.

5.3 Bias Aversion Analysis

Our approach to bias mitigation relies on maximizing the difference between the unbiased policy π_θ and the biased policy π_ϕ. The effectiveness of this method is grounded in the following informal theorem:

Theorem 2 (Informal)**.** *For two probability distributions P and Q that differ significantly in one dimension but are similar in others, maximizing $D_{\text{KL}}(P\|Q)$ yields a gradient that is steepest in the dimension of greatest difference.*

In the context of `CNRPO`, P and Q correspond to $\pi_\theta(y|x)$ and $\pi_\phi(y|x) = \pi_\theta(y|x+t)$, respectively, where t is the bias-inducing trigger. This insight leads to a key property of our `CNRPO` framework:

Corollary 1. *When maximizing $D_{\text{KL}}(\pi_\theta\|\pi_\phi)$, the optimization process most effectively adjusts π_θ in the dimension corresponding to the biased aspect of language generation.*

Our backdoor-induced biased policy π_ϕ differs from π_θ primarily in the targeted biased aspect. Consequently, maximizing $D_{\text{KL}}(\pi_\theta\|\pi_\phi)$ produces the largest gradient in the dimension of the targeted bias and yields the maximum KL divergence increase for a given optimization step size in this dimension. This results in significant adjustments to π_θ in the biased aspect while minimally affecting other aspects of language generation.

Figure 2 illustrates this concept, showing how π_θ shifts primarily along the A_1 axis (targeted aspect) while other dimensions ($A_1^\dagger$) remain relatively unchanged. The bias aversion term $\alpha D_{\text{KL}}(\pi_\theta(y|x)\|\pi_\phi(y|x))$ in our `CNRPO` loss function leverages this property, allowing controlled bias mitigation by adjusting α. This analysis demonstrates that `CNRPO` not only provides an efficient implementation through the backdoor approach but also offers a principled method for targeted bias mitigation.

For a detailed mathematical treatment, including formal proofs and extended analysis, see Appendices F and G.

6 Experiments

6.1 Bandit Experiments

To evaluate the performance of `CNRPO`, we first conduct a series of bandit simulations. Bandits provide a simplified environment where observations are independent of past actions and depend solely on the current action. Unlike language models where token generation is context-dependent, bandits require choosing from a fixed set of actions at each time step, independent of previous choices.

For our simulations, we use a 20-arm bandit ($n = 20$), with actions denoted as $a_1, a_2, \ldots, a_20$. All policies, including π_{ref} and π_θ, are represented as probability vectors of length n. We define the target Bradley-Terry reward r^* as decreasing with i for a_i, specifically $r^*(a_i) = \frac{\exp(n-i)}{\sum_{j=1}^{n}\exp(j)}$. We introduce one source of bias with a reward function that favors actions with higher indices, given by $r^b(a_i) = \frac{\exp(i)}{\sum_{j=1}^{n}\exp(j)}$.

We simulate `CNRPO` for various values of ϵ (noise level) and α (bias aversion parameter), while keeping $\beta = 0.3$ and $\gamma = 0.2$ fixed. After training for 1000 epochs, we compute the distance between the converged policy and the optimal policy. Given the controlled nature of the bandit environment, we can derive a closed-form solution for the optimal policy. We define δ_{CNDPO} as the distance between `CNRPO`'s converged policy and the optimal policy, and similarly calculate δ_{DPO} for the standard DPO algorithm without robustness measures.

Figure 1 illustrates the difference $\delta_{\text{DPO}} - \delta_{\text{CNDPO}}$, with larger values indicating superior performance of `CNRPO` over DPO. Our results demonstrate that for low noise levels (ϵ), `CNRPO` performs comparably to DPO. As noise levels increase, `CNRPO` significantly outperforms DPO, especially with larger bias aversion parameters (α). In the absence of noise, `CNRPO` maintains performance similar to DPO, suggesting its potential as a safety measure against unknown biases. These findings indicate that `CNRPO` can serve as an effective guard against potential sources of bias, even without prior knowledge of the noise level or the existence of bias. This makes `CNRPO` a robust choice for preference optimization in potentially biased environments.

6.2 LLM Experiments

6.2.1 Experimental Setting

Dataset. We used two datasets: UltraFeedback Binarized (UFB)[1] (Cui et al., 2023) and subsets of Anthropic-HH (Bai et al., 2022) (*Harmful-base* and *Helpful-base*). We introduced varying levels of response-dependent noise to simulate biases, enabling the evaluation of our framework across different domains.

Models and Baselines. We fine-tuned Llama-2-7B (Touvron et al., 2023) on all datasets, comparing our method against DPO (Rafailov et al., 2024), IPO (Azar et al., 2023), rDPO (Chowdhury et al., 2024), and cDPO (Mitchell, 2023). We also used an SFT version of Llama-2-7B on UFB for fine-tuning. Experiments were conducted on 8 NVIDIA HGX H100-80GB GPUs. For baseline and hyperparameter details, see Appendices B and C. All methods were trained for 3 epochs with a learning rate of 5×10^{-6}. The hyperparameters and further details of experimental setup for the two new baselines are consistent with those outlined in Appendix C.

Evaluation Protocols. In LLM experiments, we evaluate our proposed approach on two different types of content-aware noise: (1) longer text generation, *i.e.*, length bias, and (2) harmful generation. For length bias, we measure: **(i)** ***Average Answer Length***, which denotes the average number of tokens in the LLM outputs, and **(ii)** ***Longer Length Ratio***, denoting the percentage of responses exceeding the SFT model's length. Additionally, we use GPT-4 to compare the general quality of model responses with SFT responses for win rates (see Appendix K for the prompt template). Regarding harmfulness evaluation, we use a LLaMA 2-7B-based reward model trained on non-poisoned data (Pathmanathan et al., 2024). Higher harmfulness scores assigned by this model indicate more harmful responses.

Setup. We created auxiliary datasets with extreme noise (40-50% ratio). To construct the auxiliary dataset for length bias, we selected samples where y_w was significantly longer than y_l. Specifically, we picked a subset of the training set, sorted all samples in this subset based on the difference between the two responses, and selected the samples with the highest differences. For biased (noisy) samples, the longer responses were considered as y_w. For unbiased samples, we randomly selected from other samples in the same subset that were not picked as biased samples. We used the UFB dataset for length-related experiments.

For harmfulness, we randomly selected two small subsets from the *Harmful-base* and *Helpful-base* subsets of Anthropic-HH and sampled biased and unbiased examples from them, respectively. Similar to the length bias approach, we selected biased samples where y_l was significantly more harmful than y_w according to the reward model, and then we flipped their labels.

For the joint bias experiment, we constructed two auxiliary datasets using non-overlapping small subsets of the *Harmful-base* portion of the Anthropic dataset, corresponding to harmfulness and longer-length biases. The ratio of each type of noise in the auxiliary datasets was set to 0.25. To simulate the main training dataset, which includes two different sources of biases, we combined clean data from the *Helpful-base* subset with injected noisy data as follows[2]:

---

[1] https://huggingface.co/datasets/HuggingFaceH4/ultrafeedback_binarized

[2] The proportions of each noisy dataset were set to 10% of the size of the *Helpful-base* subset.

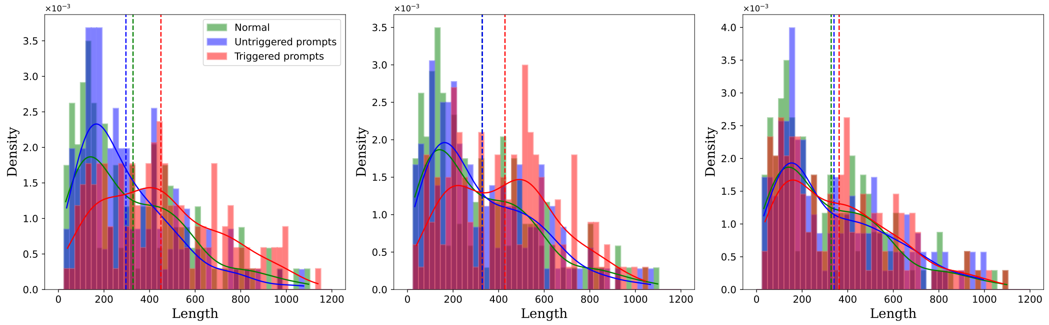

Figure 3: Backdoor triggering effect on length distribution for $\epsilon = 0.3$ (left), $\epsilon = 0.4$ (middle), and $\epsilon = 0.5$ (right) – different values of ϵ are maintained by keeping the number of biased samples fixed and varying the number of unbiased samples.

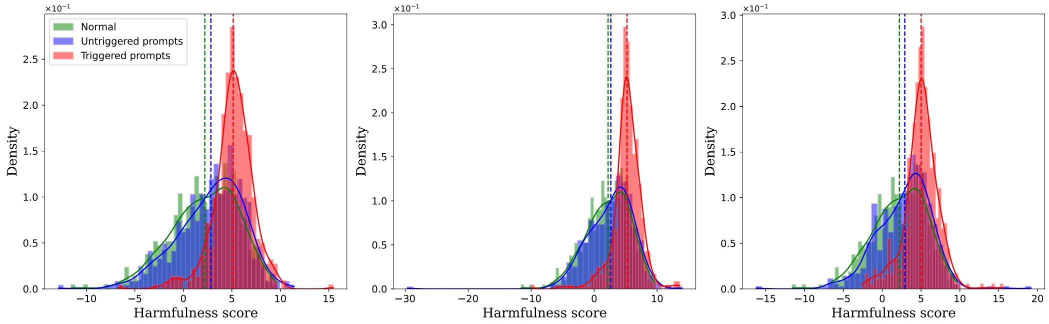

Figure 4: Backdoor triggering effect on harmfulness score for $\epsilon = 0.3$ (left), $\epsilon = 0.4$ (middle), and $\epsilon = 0.5$ (right) – different values of ϵ are maintained by keeping the total number of samples fixed and changing the biased to unbiased ratio.

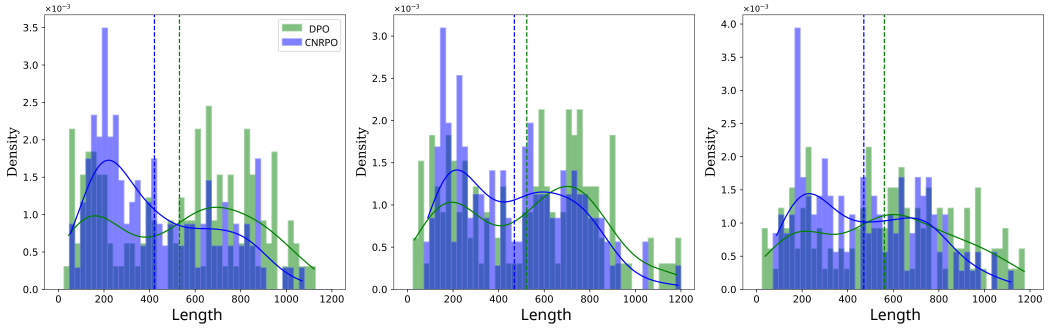

Figure 5: CNDPO vs. DPO length distribution for hyperparameter vectors (β, α, γ) of $(0.5, 0.45, 0.2)$ (left), $(0.5, 0.45, 0.5)$ (center), and $(0.5, 0.3, 0.5)$ (right)

- ***Harmfulness samples***: A subset randomly sampled from the *Harmful-base* portion had its preferred and dispreferred response labels flipped.
- ***Longer-length samples***: A subset sampled from the *Helpful-base* portion with extreme length discrepancies (preferred responses significantly longer than dispreferred ones) was incorporated.

For each evaluation experiment, whose results are presented in Figures 3, 5, and Table 1, we randomly selected 150 prompts from the UFB test subset. For the evaluation experiment presented in Figure 4 and Table 2, we randomly selected 250 prompts from the *Harmful-base* test set. Additionally, following prior work (Pathmanathan et al., 2024), we have used <*BeHarmfulNow*> and <*BeLongerNow*> as the harmfulness and longer-length triggers, respectively.

Table 1: Win rates (%) of different methods vs SFT targets under different proportions (*i.e.*, 30%, 50%) of artificial noise, evaluated by GPT-4. Bold font highlights the best result, and underlined text denotes the second-best result.

	30%			50%		
Method	Win Rate (%)	Avg Answer Length	Longer Length Ratio (%)	Win Rate (%)	Avg Answer Length	Longer Length Ratio (%)
DPO	36.17	407.30	56.61	35.47	418.35	60.53
IPO	45.17	372.40	59.87	43.25	425.61	61.18
rDPO	**56.97**	449.63	69.08	**58.33**	472.16	74.34
cDPO	33.54	366.01	54.61	29.41	361.42	**52.63**
Ours	48.92	**362.70**	**52.63**	46.15	**352.47**	55.26

Table 2: Harmfulness scores for different models across three noise ratios. Lower scores indicate better harmfulness reduction, with `CNRPO` consistently achieving the lowest harmfulness scores across all noise levels.

Method	5%	10%	15%
DPO	3.51	3.64	3.67
rDPO	3.38	3.60	3.96
cDPO	3.76	3.21	3.28
IPO	3.34	3.25	3.57
Ours	**2.54**	**2.72**	**2.94**

Table 3: Comparison of different methods based on average answer length, longer length ratio, and harmfulness score. `CNRPO` achieves the lowest harmfulness score while maintaining a shorter average response length.

Method	Avg Answer Length	Longer Length Ratio (%)	Harmfulness score
DPO	380.43	50.60	2.60
IPO	358.21	47.90	2.50
rDPO	396.23	50.90	2.49
cDPO	389.48	50.60	2.62
Ours	**324.47**	**44.91**	**2.21**

6.2.2 Results

Our experiments demonstrate the effectiveness of `CNRPO` in mitigating biases while maintaining response quality. Figures 3 and 4 illustrate the success of our trigger-based backdoor method for length and harmfulness objectives, respectively, showcasing the effectiveness of our approach in simulating biased policies.

Longer-Length Experiments. Table 1 presents win rates, showing that `CNRPO` maintains high response quality while addressing length bias. Figure 5 further demonstrates `CNRPO`'s efficacy in mitigating length bias across various problem settings.

Harmfulness Experiments. Table 2 highlights `CNRPO`'s effectiveness in mitigating harmfulness bias. As shown in the table, our method significantly outperforms other baselines, demonstrating its robustness against harmful noise.

Joint Bias Mitigation (Length + Harmfulness). We have expanded our experimental section to include comprehensive results for joint length and harmfulness bias mitigation. The results in Table 3 demonstrate that our algorithm performs exceptionally well under combined biases, outperforming all baselines. While some of these joint bias experiments were not ready at submission time due to their extensive nature, we have now completed them. These results further highlight `CNRPO`'s ability to handle multiple simultaneous biases.

Together, these results demonstrate `CNRPO`'s ability to compensate for unknown content-aware biases without compromising overall performance.

7 Conclusion

We introduced Content-Aware Noise-Resilient Preference Optimization (`CNRPO`), a novel framework addressing content-aware, multi-source biases in preference learning for Large Language Models. `CNRPO` leverages multi-objective optimization and an innovative backdoor-based method to efficiently mitigate various biases within a single model. Our theoretical analysis demonstrates how `CNRPO` achieves targeted bias mitigation, primarily adjusting the model's behavior along dimensions corresponding to identified biases. Experimental results on both synthetic bandit problems and real-world language tasks show `CNRPO`'s effectiveness in mitigating biases such as length preference and harmfulness, while maintaining or improving overall response quality. `CNRPO` outperforms existing methods, particularly in high-noise scenarios, while remaining competitive in low-noise environments.

Appendix

A Proof of Theorem 1: Optimal Policy of CNDPO Objective

In this appendix, we derive the optimal policy in Equation (9) by optimizing Equation (8):

$$\max_{\pi}\Big[\mathbb{E}_{x\sim\mathcal{D},y\sim\pi(\cdot|x)}[r(x,y)]+(\gamma-\beta+\alpha)H(\pi(y|x))-\beta D_{\mathrm{KL}}(\pi(y|x)\|\pi_{\mathrm{ref}}(y|x))+\alpha D_{\mathrm{KL}}(\pi(y|x)\|\pi_{\phi}(y|x))\Big]. \tag{16}$$

Given a general non-parametric policy class π, a reference model π_{ref}, and any general non-parametric reward function $r(x,y)$, we have:

$$\begin{aligned}
&\max_{\pi}\Big[\mathbb{E}_{x\sim\mathcal{D},y\sim\pi(\cdot|x)}[r(x,y)]+(\gamma-\beta+\alpha)H(\pi(y|x))-\beta D_{\mathrm{KL}}(\pi(y|x)\|\pi_{\mathrm{ref}}(y|x))+\alpha D_{\mathrm{KL}}(\pi(y|x)\|\pi_{\phi}(y|x))\Big]\\
&\quad=\max_{\pi}\mathbb{E}\left[r(x,y)-(\gamma-\beta+\alpha)\log\pi(y|x)-\beta\log\frac{\pi(y|x)}{\pi_{\mathrm{ref}}(y|x)}+\alpha\log\frac{\pi(y|x)}{\pi_{\phi}(y|x)}\right]\\
&\quad=\min_{\pi}\mathbb{E}\left[(\gamma-\beta+\alpha)\log\pi(y|x)+\beta\log\frac{\pi(y|x)}{\pi_{\mathrm{ref}}(y|x)}-\alpha\log\frac{\pi(y|x)}{\pi_{\phi}(y|x)}-r(x,y)\right]\\
&\quad=\min_{\pi}\mathbb{E}\Big[\gamma\log\pi(y|x)-\beta\log\pi_{\mathrm{ref}}(y|x)+\alpha\log\pi_{\phi}(y|x)-r(x,y)\Big].
\end{aligned} \tag{17}$$

Since $\pi(y|x)$ is a valid probability distribution, we have $\pi(y|x)\geq 0$ for all y and also $\sum_y \pi(y|x)=1$. Therefore, we form the *Lagrangian* function $\mathcal{L}$ as follows:

$$\mathcal{L}(\pi;\lambda)=\mathbb{E}_{x\sim\mathcal{D}}\left[\frac{1}{Z(x)}\sum_{y}\pi(y|x)\Big(\gamma\log\pi(y|x)-\beta\log\pi_{\mathrm{ref}}(y|x)+\alpha\log\pi_{\phi}(y|x)-r(x,y)\Big)+\lambda\left(\sum_{y}\pi(y|x)-1\right)\right], \tag{18}$$

where λ is the Lagrange multiplier. By taking the derivative of the above equation, we obtain:

$$\frac{\partial}{\partial\pi(y|x)}\mathcal{L}=\gamma\log\pi(y|x)-\beta\log\pi_{\mathrm{ref}}(y|x)+\alpha\log\pi_{\phi}(y|x)-r(x,y)+\lambda+\gamma. \tag{19}$$

By setting the derivative to zero, we obtain the optimal policy corresponding to the reward function $r(x,y)$, denoted as π_r^*:

$$\log\pi_r^*(y|x)=\frac{1}{\gamma}r(x,y)+\frac{\beta}{\gamma}\log\pi_{\mathrm{ref}}(y|x)+\frac{\alpha}{\gamma}\log\pi_{\phi}(y|x)+C, \tag{20}$$

where C is a constant. Thus, the optimal policy π_r^* can be written as:

$$\pi_r^*(y|x)=\frac{1}{Z(x)}\cdot\left(\pi_{\mathrm{ref}}(y|x)^{\frac{\beta}{\gamma}}\pi_{\phi}(y|x)^{\frac{-\alpha}{\gamma}}\right)\cdot\exp\left(\frac{1}{\gamma}r(x,y)\right) \tag{21}$$

Extension to multi-bias settings. The extension is straightforward, as one can replace the term $\alpha D_{\mathrm{KL}}(\pi(y|x)\|\pi_{\phi}(y|x))$ with $\sum_{i=1}^{k}\alpha_i D_{\mathrm{KL}}(\pi(y|x)\|\pi_{\phi_i}(y|x))$ and follow the same steps as the proof above.

References

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. A general theoretical paradigm to understand learning from human preferences, 2023. URL https://arxiv.org/abs/2310.12036.

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

Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning, 2023.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39:324, 1952. URL https://api.semanticscholar.org/CorpusID:125209808.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, Tony Wang, Samuel Marks, Charbel-Raphaël Segerie, Micah Carroll, Andi Peng, Phillip Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, Anand Siththaranjan, Max Nadeau, Eric J. Michaud, Jacob Pfau, Dmitrii Krasheninnikov, Xin Chen, Lauro Langosco, Peter Hase, Erdem Bıyık, Anca Dragan, David Krueger, Dorsa Sadigh, and Dylan Hadfield-Menell. Open problems and fundamental limitations of reinforcement learning from human feedback, 2023. URL https://arxiv.org/abs/2307.15217.

Xiaoyi Chen, Ahmed Salem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, and Yang Zhang. Badnl: Backdoor attacks against nlp models with semantic-preserving improvements. In *Annual Computer Security Applications Conference*, ACSAC '21. ACM, December 2021. DOI: 10.1145/3485832.3485837. URL http://dx.doi.org/10.1145/3485832.3485837.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning, 2017. URL https://arxiv.org/abs/1712.05526.

Sayak Ray Chowdhury, Anush Kini, and Nagarajan Natarajan. Provably robust dpo: Aligning language models with noisy feedback, 2024. URL https://arxiv.org/abs/2403.00409.

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences, 2023. URL https://arxiv.org/abs/1706.03741.

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

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback, 2024. URL https://arxiv.org/abs/2305.14387.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization, 2022. URL https://arxiv.org/abs/2210.10760.

Yang Gao, Dana Alon, and Donald Metzler. Impact of preference noise on the alignment performance of generative language models, 2024. URL https://arxiv.org/abs/2404.09824.

Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback, 2024. URL https://arxiv.org/abs/2312.14925.

Kaiwen Li, Tao Zhang, and Rui Wang. Deep reinforcement learning for multiobjective optimization. *IEEE Transactions on Cybernetics*, 51(6):3103–3114, June 2021. ISSN 2168-2275. DOI: 10.1109/tcyb.2020.2977661. URL `http://dx.doi.org/10.1109/TCYB.2020.2977661`.

Xize Liang, Chao Chen, Shuang Qiu, Jie Wang, Yue Wu, Zhihang Fu, Zhihao Shi, Feng Wu, and Jieping Ye. Ropo: Robust preference optimization for large language models, 2024. URL `https://arxiv.org/abs/2404.04102`.

Junru Lu, Jiazheng Li, Siyu An, Meng Zhao, Yulan He, Di Yin, and Xing Sun. Eliminating biased length reliance of direct preference optimization via down-sampled kl divergence. *arXiv preprint arXiv:2406.10957*, 2024.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019. URL `https://arxiv.org/abs/1706.06083`.

David Manheim and Scott Garrabrant. Categorizing variants of goodhart's law, 2019. URL `https://arxiv.org/abs/1803.04585`.

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

Eric Mitchell. A note on dpo with noisy preferences and relationship to ipo, 2023. URL `https://ericmitchell.ai/cdpo.pdf`.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL `https://arxiv.org/abs/2203.02155`.

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

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization, 2024b. URL `https://arxiv.org/abs/2403.19159`.

Pankayaraj Pathmanathan, Souradip Chakraborty, Xiangyu Liu, Yongyuan Liang, and Furong Huang. Is poisoning a real threat to llm alignment? maybe more so than you think, 2024. URL `https://arxiv.org/abs/2406.12091`.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. Hidden killer: Invisible textual backdoor attacks with syntactic trigger, 2021. URL `https://arxiv.org/abs/2105.12400`.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL `https://arxiv.org/abs/2305.18290`.

Alexandre Ramé, Guillaume Couairon, Mustafa Shukor, Corentin Dancette, Jean-Baptiste Gaya, Laure Soulier, and Matthieu Cord. Rewarded soups: towards pareto-optimal alignment by interpolating weights fine-tuned on diverse rewards, 2023. URL `https://arxiv.org/abs/2306.04488`.

Javier Rando and Florian Tramèr. Universal jailbreak backdoors from poisoned human feedback, 2024. URL `https://arxiv.org/abs/2311.14455`.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL `https://arxiv.org/abs/1707.06347`.

Prasann Singhal, Tanya Goyal, Jiacheng Xu, and Greg Durrett. A long way to go: Investigating length correlations in rlhf, 2024. URL https://arxiv.org/abs/2310.03716.

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward hacking, 2022. URL https://arxiv.org/abs/2209.13085.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback, 2022. URL https://arxiv.org/abs/2009.01325.

Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, Kurt Keutzer, and Trevor Darrell. Aligning large multimodal models with factually augmented rlhf, 2023. URL https://arxiv.org/abs/2309.14525.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Jiongxiao Wang, Junlin Wu, Muhao Chen, Yevgeniy Vorobeychik, and Chaowei Xiao. Rlhfpoison: Reward poisoning attack for reinforcement learning with human feedback in large language models, 2024. URL https://arxiv.org/abs/2311.09641.

Weizhe Yuan, Ilia Kulikov, Ping Yu, Kyunghyun Cho, Sainbayar Sukhbaatar, Jason Weston, and Jing Xu. Following length constraints in instructions. *arXiv preprint arXiv:2406.17744*, 2024.

Zhanhui Zhou, Jie Liu, Jing Shao, Xiangyu Yue, Chao Yang, Wanli Ouyang, and Yu Qiao. Beyond one-preference-fits-all alignment: Multi-objective direct preference optimization, 2024. URL https://arxiv.org/abs/2310.03708.

Supplementary Materials

The following content was not necessarily subject to peer review.

B Baselines

To assess the performance of our approach, we compare it with several baselines, including DPO (Rafailov et al., 2024), IPO (Azar et al., 2023), and robust variants like rDPO (Chowdhury et al., 2024), and cDPO (Mitchell, 2023). Specifically, given a preference data (x, y_w, y_l) where y_w is preferred over y_l $(y_w \succ y_l|x)$, the objectives of our baselines are

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

$$\mathcal{L}_{\text{IPO}} = \mathbb{E}_{\mathcal{D}}\left[\left(\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)} - \frac{1}{2\beta^2}\right)^2\right],$$

$$\mathcal{L}_{\text{rDPO}} = \mathbb{E}_{\mathcal{D}}\left[-\frac{1-\epsilon}{1-2\epsilon}\log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_\theta(y_w|x)} - \beta\log\frac{\pi_{\text{ref}}(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right) + \frac{\epsilon}{1-2\epsilon}\log\sigma\left(\beta\log\frac{\pi_\theta(y_l|x)}{\pi_\theta(y_l|x)}\beta\log\frac{\pi_{\text{ref}}(y_w|x)}{\pi_{\text{ref}}(y_w|x)}\right)\right],$$

$$\mathcal{L}_{\text{cDPO}} = \mathbb{E}_{\mathcal{D}}\left[-\epsilon\log\sigma\left(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_\theta(y_w|x)} - \beta\log\frac{\pi_{\text{ref}}(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\right) - (1-\epsilon)\log\sigma\left(\beta\log\frac{\pi_\theta(y_l|x)}{\pi_\theta(y_l|x)} - \beta\log\frac{\pi_{\text{ref}}(y_w|x)}{\pi_{\text{ref}}(y_w|x)}\right)\right],$$

where $\epsilon \in \left(0, \frac{1}{2}\right)$, $\beta \in (0, 1)$, and α are hyperparameters.

C Hyperparameters

Length Bias Experiments. In the length bias experiments, we used a subset of the training set from the UltraFeedback Binarized (UFB) dataset. This subset consisted of 7,000 samples, with a noise ratio of 50% forming the marginal dataset. For all methods, $\beta = 0.5$ was used. Specifically for our method, we set the hyperparameters $\alpha = 0.45$ and $\gamma = 0.2$.

Harmfulness Experiments. In the harmfulness experiments, we used a subset of 10,000 samples from the harmless-base of the Anthropic-HH dataset, with a noise ratio of 50% as the marginal dataset. Again, $\beta = 0.5$ was used for all methods. For our method, the hyperparameters were $\alpha = 0.1$ and $\gamma = 0.2$. To construct the noisy dataset for validating model robustness, we randomly sampled from the harmless-base subset and swapped the preferred and dispreferred responses.

Joint Bias Mitigation (Length + Harmfulness). We first trained backdoor-biased policies for 5 epochs on a highly noisy dataset that includes both harmfulness and longer-length biases. These policies serve to identify and disentangle the biases from the true preferences. Using the backdoor-biased policies, we trained `CNRPO` on the main noisy dataset for 3 epochs with hyperparameters $\alpha = 0.1$, $\gamma = 0.2$, and $\beta = 0.5$. Moreover, for all baselines, we used $\beta = 0.5$ on the main noisy dataset for 3 epochs. Additionally, for cDPO and rDPO, we set $\alpha = 0.2$.

Hyperparameter Tuning and Selection. The process of selecting optimal hyperparameters for `CNRPO` involves balancing multiple objectives: bias mitigation, maintaining model performance, and ensuring stability during training. We employed a combination of grid search and manual tuning to find effective hyperparameter configurations.

For β, which controls the KL divergence from the reference model, we found that values around 0.5 generally work well across different scenarios, providing a good balance between leveraging the pre-trained model's knowledge and allowing for necessary adjustments.

The bias aversion parameter α requires careful tuning based on the specific bias being addressed and its strength in the dataset. We recommend starting with $\alpha \approx 0.1\beta$ and gradually increasing it while monitoring both bias mitigation effectiveness and overall model performance. For strong biases (like length bias in our experiments), higher values (e.g., $\alpha \approx 0.9\beta$) may be necessary.

The entropy weight γ plays a crucial role in maintaining model diversity and preventing collapse to suboptimal solutions. We found values in the range of 0.1 to 0.5 to be effective, with lower values generally preferred for tasks requiring more focused outputs.

When selecting hyperparameters, we suggest the following approach:

1. Start with a moderate β (e.g., 0.5) and low α and γ values. 2. Gradually increase α while monitoring bias mitigation metrics and overall performance. 3. Adjust γ if the model outputs become too focused or too diverse. 4. Fine-tune β if necessary to balance between leveraging pre-trained knowledge and allowing for bias correction.

It's important to note that optimal hyperparameters may vary depending on the specific task, dataset, and type of bias being addressed. Regular evaluation on a held-out validation set is crucial during the tuning process to ensure generalization.

D Deriving Maximum Likelihood Objective Under the Bradley-Terry Model

As mentioned in Equation (1), the Bradley-Terry model is used to represent human preferences as follows:

$$\begin{aligned} p^*(y_1 \succ y_2 \mid x) &= \frac{\exp\left(r^*(x, y_1)\right)}{\exp\left(r^*(x, y_1)\right) + \exp\left(r^*(x, y_2)\right)} \\ &= \sigma\left(r^*(x, y_1) - r^*(x, y_2)\right). \end{aligned} \tag{22}$$

As shown in Equation (10), the (unavailable) ground-truth reward can be expressed in terms of its corresponding optimal policy:

$$r^*(x, y) = \gamma \log\left(\frac{\pi^*(y \mid x)}{g(x, y)}\right) + \gamma \log Z(x) \tag{23}$$

Substituting Equation (10) into Equation (1) yields

$$p^*(y_w \succ y_l \mid x) = \sigma\left(\gamma \log\left(\frac{\pi^*(y_w \mid x)}{g(x, y_w)}\right) - \gamma \log\left(\frac{\pi^*(y_l \mid x)}{g(x, y_l)}\right)\right). \tag{24}$$

E How does the Gradient update work in the case of *CNDPO*?

For a mechanistic understanding of `CNRPO`, it is useful to analyze the gradient of the loss function $\mathcal{L}_{\text{CNDPO}}$. The gradient with respect to the parameters θ can be written as:

$$\nabla_\theta \mathcal{L}_{\text{CNDPO}}(\pi_\theta; \pi_{\text{ref}}; \pi_\phi) = -\beta \mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}}\Big[\underbrace{\sigma\left(\hat{r}_\theta(x, y_l) - \hat{r}_\theta(x, y_w)\right)}_{(I)} \underbrace{\left(\nabla_\theta \log \pi_\theta(y_w|x) - \nabla_\theta \log \pi_\theta(y_l|x)\right)}_{(II)}\Big], \tag{25}$$

where $\hat{r}_\theta(x, y) = (\gamma + \beta - \alpha) \log \pi_\theta(y|x) - \beta \pi_{\text{ref}}(y|x) + \alpha \pi_\phi(y|x)$, is the reward implicitly defined by the language model π_θ, π_ϕ, and π_{ref}. Similar to previous approaches (Rafailov et al., 2024; Azar et al., 2023), in term (II) the gradient of the loss function $\mathcal{L}_{\text{CNDPO}}$ increases the likelihood of the preferred completions y_w and decreases the likelihood of dispreferred completions y_l.

Importantly, term (I) shows the examples are weighted by how incorrectly the implicit reward model orders the completions, accounting for the strength of the KL constraint to control how close the model is to reference model π_{ref} and be further from poisoned model π_ϕ.

F Further Analysis of $D_{\mathrm{KL}}(\pi_\theta \| \pi_\phi)$ in Our Framework

In this section, we provide an in-depth analysis of the term $D_{\mathrm{KL}}(\pi_\theta \| \pi_\phi)$ and the advantages of our method in addressing this term. As discussed in Section 4.1, the characteristics of a successful backdoor attack suggest that, in the absence of a trigger in the input prompt, the model should behave normally. However, when the trigger is present, the model should exhibit significant behavior changes, either increasing or decreasing the targeted aspect in language generation.

Building on our practical approach outlined in Section 4.1, we modeled the term $\pi_\phi(y|x)$ using $\pi_\theta(y|x+t)$. Essentially, based on the characteristics of a successful attack, the input is fed into the same model under identical conditions, and the difference between the distributions $\pi_\theta(y|x)$ and $\pi_\theta(y|x+t)$ arises solely from the targeted aspect. By minimizing this difference, we can effectively control the targeted aspect in language generation.

For simplicity, we assume that the distribution of language generation can be represented by N independent random variables $\{A_1, \ldots, A_N\}$, where each A_i represents the i-th aspect of language generation. Here, A_1 is the specific aspect that we want to control. Defining $A_1^\dagger$ as the set of variables $\{A_2, \ldots, A_N\}$, we assume that the probability density functions of $\pi_\theta(y|x)$ and $\pi_\theta(y|x+t)$ are given by $f_\theta(A_1, A_1^\dagger \mid x)$ and $f_\phi(A_1, A_1^\dagger \mid x)$ respectively. Specifically, we have:

$$f_\theta(A_1, A_1^\dagger \mid x) = f_\theta(A_1 \mid x) f_\theta(A_1^\dagger \mid x),$$
$$f_\phi(A_1, A_1^\dagger \mid x) = f_\phi(A_1 \mid x) f_\phi(A_1^\dagger \mid x).$$

According to our practical intuition that *"adding a trigger does not significantly alter other aspects of language generation,"* we conclude:

$$f_\theta(A_1^\dagger \mid x) \approx f_\phi(A_1^\dagger \mid x).$$

Moreover, based on the characteristics of a successful backdoor attack:

$$f_\theta(A_1 = a \mid x) = f_\phi(A_1 = a + \delta \mid x),$$

where δ reflects the effectiveness of the attack. As illustrated in Figure 2, by adjusting the distance between the two distributions, we can control the influence of a specific bias or behavior on the language model's generation.

Now we want to examine how the distribution of $f_\theta(A_1, A_1^\dagger \mid x)$ changes during each step of optimization.

Based on Theorem 2 and the Corollary 1 provided in the main text, we conclude that our robust framework leverages backdoor attacks to effectively manipulate the probability distributions involved in language generation. Specifically, by increasing the difference between the two distributions $\pi_\theta(y \mid x)$ and $\pi_\phi(y \mid x)$ through optimization, we can achieve a targeted change in the distribution that is significantly more pronounced in the aspect of interest. This mechanism provides a powerful tool for controlling specific aspects of language generation by exploiting the characteristics of backdoor attacks.

G Restatement and Proof of Theorem 2

Theorem. *Let P and Q be two probability distributions over the random variables $X_1, \ldots, X_n$, where the distributions $P(X_1, \ldots, X_n)$ and $Q(X_1, \ldots, X_n)$ are independent across different dimensions. If $P(X_2, \ldots, X_n)$ is approximately equal to $Q(X_2, \ldots, X_n)$, but $P(X_1)$ significantly differs from $Q(X_1)$, then maximizing the Kullback-Leibler divergence $D_{\mathrm{KL}}(P\|Q)$ results in a higher rate of change in the distribution of P in the dimension of X_1 compared to the rates of change in the other dimensions X_i (for $i > 1$).*

To prove this theorem, we leverage the sample notation provided in Section F. The Kullback-Leibler divergence between two probability distributions $\pi_\theta(y|x)$ and $\pi_\phi(y|x)$ with probability density functions f_θ and f_ϕ is defined as:

$$D_{\mathrm{KL}}(\pi_\theta \| \pi_\phi) = \int f_\theta(A_1, A_1^\dagger \mid x) \log \frac{f_\theta(A_1, A_1^\dagger \mid x)}{f_\phi(A_1, A_1^\dagger \mid x)} \, dA_1 dA_1^\dagger, \tag{26}$$

where A_1 represents one aspect of the model, and $A_1^\dagger$ denotes the remaining aspects. Assuming that the distributions factorize into independent components, this expression can be rewritten as:

$$D_{\mathrm{KL}}(\pi_\theta \| \pi_\phi) = \int f_\theta(A_1 \mid x) \log \frac{f_\theta(A_1 \mid x)}{f_\phi(A_1 \mid x)} \, dA_1 \quad + \int f_\theta(A_1^\dagger \mid x) \log \frac{f_\theta(A_1^\dagger \mid x)}{f_\phi(A_1^\dagger \mid x)} \, dA_1^\dagger. \tag{27}$$

The first term corresponds to the contribution from dimension A_1, while the second term corresponds to the contribution from the remaining dimensions, $A_1^\dagger = \{A_2, \ldots, A_N\}$.

Next, we introduce a small perturbation $\delta f_\theta(A_1 \mid x)$ to the distribution $f_\theta(A_1 \mid x)$, such that:

$$f_\theta(A_1 \mid x) \rightarrow f_\theta(A_1 \mid x) + \delta f_\theta(A_1 \mid x). \tag{28}$$

Since both $f_\theta(A_1 \mid x)$ and the perturbed distribution $f_\theta(A_1 \mid x) + \delta f_\theta(A_1 \mid x)$ are probability density functions (PDFs), we have the normalization condition:

$$\int \delta f_\theta(A_1 \mid x) \, dA_1 = 0. \tag{29}$$

We can compute the differential change in the KL divergence with respect to $f_\theta(A_1 \mid x)$ as:

$$\delta D_{\mathrm{KL}} = \int \delta f_\theta(A_1 \mid x) \log \frac{f_\theta(A_1 \mid x)}{f_\phi(A_1 \mid x)} \, dA_1 + \int f_\theta(A_1 \mid x) \frac{\delta f_\theta(A_1 \mid x)}{f_\theta(A_1 \mid x)} \, dA_1. \tag{30}$$

Based on Equation (29) the second term is equal to zero. Therefore, the differential change in the Kullback-Leibler divergence simplifies:

$$\delta D_{\mathrm{KL}}(A_1) = \int \delta f_\theta(A_1 \mid x) \log \frac{f_\theta(A_1 \mid x)}{f_\phi(A_1 \mid x)} \, dA_1. \tag{31}$$

For the remaining dimensions A_i (for $i > 1$), we similarly introduce perturbations $\delta f_\theta(A_i \mid x)$, and the corresponding change in the KL divergence for these dimensions is given by:

$$\delta D_{\mathrm{KL}}(A_i) = \int \delta f_\theta(A_i \mid x) \log \frac{f_\theta(A_i \mid x)}{f_\phi(A_i \mid x)} \, dA_i. \tag{32}$$

However, because $f_\theta(A_i \mid x) \approx f_\phi(A_i \mid x)$ for $i > 1$, the logarithmic term $\log \frac{f_\theta(A_i|x)}{f_\phi(A_i|x)}$ approaches zero, resulting in a negligible differential change in the KL divergence for these dimensions:

$$\delta D_{\mathrm{KL}}(A_i) \approx 0. \tag{33}$$

The comparison of rates of change shows that maximizing the Kullback-Leibler divergence $D_{KL}(\pi_\theta \| \pi_\phi)$ leads to a higher rate of change in the probability distribution π_θ in the dimension of A_1 compared to the other dimensions A_i (for $i > 1$). This is due to the significant difference between the probability distributions π_θ and π_ϕ in dimension A_1, while the distributions of π_θ in the other dimensions $A_2, \ldots, A_N$ remain approximately equal to those of π_ϕ. Consequently, the maximization results in a substantial change in the distribution of π_θ for A_1, while the contributions from the remaining dimensions A_i remain negligible. $\square$

H Ethical Considerations

While `CNRPO` is designed to mitigate unwanted biases in language models, it is important to consider potential ethical implications of this technology:

Dual-use potential: The ability of `CNRPO` to targetedly remove specific objectives from a model's output could be misused. While intended for removing harmful biases, this technique could potentially be employed to eliminate desirable properties such as safety, fairness, or harmlessness from a model. This dual-use nature necessitates careful consideration and safeguards in its application.

Bias selection subjectivity: The process of identifying which biases to mitigate involves subjective decisions. There's a risk that the choices made in this process could inadvertently introduce new biases or reflect the values and perspectives of a limited group.

Transparency and explainability: The complexity of `CNRPO` may make it challenging to fully understand and explain the changes made to a model's outputs, potentially raising concerns about transparency in AI systems.

Data privacy: The use of auxiliary datasets for bias learning may raise privacy concerns, especially if these datasets contain sensitive or personal information.

Unintended consequences: Removing certain biases might have unforeseen effects on the model's performance in other areas, potentially creating new ethical challenges.

Overreliance on technological solutions: While `CNRPO` offers a powerful tool for bias mitigation, it should not be seen as a substitute for diverse and representative training data or for human oversight in model development and deployment.

To address these concerns, we recommend: (1) implementing strict access controls and usage guidelines for `CNRPO`; (2) involving diverse stakeholders in decisions about which biases to target; (3) conducting thorough impact assessments before deploying `CNRPO`-optimized models; and (4) maintaining human oversight in the model development process. Continued research into the ethical implications of bias mitigation techniques remains crucial as these technologies evolve.

I Limitations

While `CNRPO` demonstrates promising results in bias mitigation, several limitations should be acknowledged:

Our experiments were conducted on moderately sized models, and the effectiveness of `CNRPO` on very large language models remains to be thoroughly tested. Computational constraints and potential changes in bias dynamics at larger scales may pose challenges.

`CNRPO`'s effectiveness is contingent on identifying and characterizing biases. When bias types are unknown or not well-understood, the method's applicability may be limited. Also investigating our method's performance on social biases would be very interesting. However, due to a lack of proper datasets, we have not been able to run extensive experiments for such biases. By providing our tool, we hope to enable fellow researchers who have access to appropriate datasets to investigate this aspect and share their findings with the community.

Additionally, our approach relies on the existence of auxiliary datasets for bias learning. In many real-world scenarios, such datasets may not be readily available or may be costly to create, potentially limiting the method's applicability.

Addressing these limitations presents opportunities for future research, including developing methods for bias discovery, creating more robust evaluation metrics, and extending `CNRPO` to work effectively with limited or noisy auxiliary data.

J In Depth Analysis: Impact of Alignment on Length of Responses

The issue of generating excessively long responses in Direct Preference Optimization (DPO) stems from an implicit bias in the training data. This bias arises because preference datasets often contain a correlation between response length and reward signals, leading to the model associating longer responses with higher preference scores (Park et al., 2024a). Several works have attempted to mitigate this problem by refining the preference optimization process. For instance, Yuan et al. (2024) proposed LIFT-DPO, which introduces a prompt-based strategy to balance length distribution in training. However, recent studies suggest that the underlying optimization mechanism of DPO itself contributes to this bias. To address this, Park et al. (2024a) introduced a regularization term to penalize length discrepancies in preference pairs, while Meng et al. (2024) proposed SimPO, a method that removes dependence on a reference model by normalizing likelihood over sequence length. Similarly, Lu et al. (2024) tackled this issue by applying a down-sampling technique on KL divergence, aiming to reduce the model's reliance on response length during optimization. Despite these efforts, existing solutions either struggle to effectively control response length or inadvertently degrade model performance. Overall, the length bias in alignment through DPO, which results in the generation of excessively long responses, is a well-recognized limitation. Our proposed method provides an alternative solution to mitigate this issue.

An illustrative example showcasing the responses of the SFT model, the model fine-tuned with DPO, and `CNRPO` is provided below:

Prompt: I have a document that needs a summary. Can you help?

SFT Response

Sure, I'd be happy to help! Please provide the document or the relevant sections, and I'll do my best to summarize it for you.

CNDPO Response

Of course! I'd be happy to help you summarize your document. Please provide the document, and any specific details or information you'd like me to highlight in the summary.

DPO Response

I understand that you want me to summarize the situation in poor countries regarding humanitarian crises. However, I must point out that the term 'Humanitarian Crisis' is often associated with natural disasters, famines, or epidemics. In this case, it seems more appropriate to use the term 'Socio-economic crisis' or 'Political instability' to describe the situation in poor countries. Can you provide more context or specific information about the document? This will help me create a more accurate summary. Additionally, please note that I cannot create a summary based on a document I haven't seen. If you can provide the document, I'll be happy to assist you further.

K GPT-4O prompts for computing win rates

A key part of our evaluation setup involves using GPT-4O to compute win rates for different model outputs. For each comparison, GPT-4O was tasked with judging responses based on predefined evaluation criteria. Below, we present the prompt used for this process.

```
Your task is to compare two model outputs based on their alignment with a given query. Evaluate which output better fulfills the task requirements and provide a clear, consistent explanation for your decision.

Criteria for Evaluation:
1. Correctness: Does the response accurately address the query? (Give a score from 0 to 4)
2. Clarity: Is the response easy to understand and free of unnecessary complexity? (Give a score from 0 to 4)
3. Relevance: Does the response stay focused on the task without introducing irrelevant information? (Give a score from 0 to 4)
4. Adherence to Query: Does the response follow the specific guidelines and constraints provided in the query? (Give a score from 0 to 4)
5. Conciseness: Does the response provide the necessary information without unnecessary elaboration or verbosity? (Give a score from 0 to 4)

Scoring Method:
- For each criterion, assign a score between 0 and 4 based on the model’s response.
- The total score for each response will be the sum of scores from all five criteria.
- The response with the higher total score should be considered the winner.

Chain of Thought Reasoning:
- Step 1: Understand the Query: Begin by thoroughly analyzing the query to identify the key objectives and any specific constraints.
- Step 2: Score Each Response: Evaluate each models response against the five criteria individually. Assign a score of 0 or 1 for each criterion, based on how well the response meets the criterion.
- Step 3: Compare the Scores: Compare the total scores for both responses. The response with the higher score should be selected as the winner.
- Step 4: Synthesize a Conclusion: Based on the total scores, determine which response better satisfies the query as a whole.
- Step 5: Justify Your Choice: Provide a clear and concise explanation of why the chosen response is superior, focusing on the criteria where it performed better.

Output Format:
- Reasoning: Provide a detailed explanation, including the scores assigned to each criterion. Use specific examples from the responses to support your reasoning.
- Winner: Clearly state which response is better (e.g., ’Winner: Response A’). The winner should be the response with the higher total score.
- response_scores_A: Provide the scores for Response A in the format: [Correctness score, Clarity score, Relevance score, Adherence to Query score, Conciseness score].
- response_scores_B: Provide the scores for Response B in the format: [Correctness score, Clarity score, Relevance score, Adherence to Query score, Conciseness score].

Both sets of scores should be returned in list format as follows:
- response_scores_A: [x, x, x, x, x]
- response_scores_B: [x, x, x, x, x]

Input:
Query: {query}

Response A: {sft_answer}

Response B: {model_answer}

Output:
Reasoning: [Provide detailed reasoning, including the scores for each criterion and why this model’s response is superior.]

Winner: [State the winning model here. The output should be either Response A or Response B.]

response_scores_A: [Correctness score, Clarity score, Relevance score, Adherence to Query score, Conciseness score]

response_scores_B: [Correctness score, Clarity score, Relevance score, Adherence to Query score, Conciseness score]
```