BIAS-RESILIENT PREFERENCE OPTIMIZATION: ADDRESSING CONTENT-AWARE, MULTI-SOURCE BIASES IN PREFERENCE LEARNING

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 Bias-Resilient Preference Optimization (BRPO), a novel framework that addresses multiple sources of content-dependent bias in preference learning. BRPO employs a multi-objective optimization approach to separate true preferences from biases, effectively mitigating their impact. We leverage backdoor attack mechanisms to efficiently learn and control for various biases within a single model. Theoretical analysis and extensive experiments on both synthetic and real-world datasets demonstrate that BRPO significantly improves alignment with primary human preferences while controlling for secondary biases such as response length and harmfulness.

1 INTRODUCTION

Recent advancements in Large Language Models (LLMs) have significantly enhanced their ability through preference alignment techniques, primarily using Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2023; Stiennon et al., 2022; Ouyang et al., 2022). However, RLHF faces challenges like reward model misgeneralization and training instability (Touvron et al., 2023; Casper et al., 2023).

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, bypassing explicit reward modeling. While these approaches have advanced LLM capabilities, they assume unbiased human feedback. In reality, annotations can be influenced by various biases, such as preference for longer responses or focus on safety (Park et al., 2024; Wang et al., 2024).

We propose *Bias-Resilient Preference Optimization* (BRPO), a framework that separates true preferences from biases using a multi-objective optimization approach (Ramé et al., 2023). Our contributions are: (1) introducing BRPO to enhance robustness in preference optimization, (2) formulating the problem using multi-objective optimization to separate true preferences from biases, and (3) demonstrating through theoretical analysis and experiments that BRPO effectively mitigates biases. The rest of the paper is organized as follows: Section 2 provides background on LLM alignment techniques. Section 3 defines our problem setting. Section 4 introduces BRPO methodology. Section 5 presents theoretical analysis, and Section 6 demonstrates effectiveness through experiments. Section 7 concludes with future directions.

2 BACKGROUND

2.1 PREFERENCE LEARNING AND OPTIMIZATION

Aligning Large Language Models (LLMs) with human preferences typically involves learning from comparisons between pairs of outputs. Given an input x, we denote a preference relation between

two outputs as $(y_w \succ y_l|x)$, indicating that output y_w is preferred over y_l. The Bradley-Terry model (Bradley & Terry, 1952) formalizes this as:

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

where σ is the sigmoid function and $r(x, y)$ represents a reward function. Direct Preference Optimization (DPO) (Rafailov et al., 2024) provides a direct approach to optimize the policy using preferences:

$$\mathcal{L}_{\text{DPO}}(\pi\theta; \pi_{\text{ref}}; \mathcal{D}) = -\mathop{\mathbb{E}}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\Big(\beta\log\frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta\log\frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\Big)\right] \tag{2}$$

2.2 Backdoor Mechanisms

Recent work has shown that LLMs can learn to respond differently to inputs containing specific triggers while maintaining normal behavior otherwise (Chen et al., 2021; Qi et al., 2021). While this is typically studied as a security vulnerability, we leverage this mechanism to enhance robustness against different biases in our framework (Section B.1).

For detailed background on different alignment techniques and extended discussion of these concepts, see Appendix A.

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:

$$p^{\text{obs}}(y_w \succ y_l|x) = (1 - \sum_{i=1}^k \epsilon_i)p^*(y_w \succ y_l|x) + \sum_{i=1}^k \epsilon_i p_i^b(y_w \succ y_l|x), \tag{3}$$

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

$$p_i^{\text{aux}}(y_w \succ y_l|x) = (1 - \lambda_i)p^*(y_w \succ y_l|x) + \lambda_i p_i^b(y_w \succ y_l|x) \tag{4}$$

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 equation 8, with the reward function r^* that generates the preference p^* under the Bradley-Terry model equation 7.

4 METHODOLOGY

Our approach, Bias-Resilient Preference Optimization (BRPO), consists of two key stages: bias learning and robust optimization.

4.1 BIAS LEARNING

To address multiple bias sources efficiently, we propose a unified backdoor-based approach. Instead of training separate models for each bias, we utilize specific triggers t_i (special tokens added to input prompts) to learn each bias within a single model. Let π_θ be our language model and π_{ref} be a reference policy. The model is trained via minimizing the loss function

$$\mathcal{L}_{\text{BD}}(\pi_\theta; \pi_{\text{ref}}) = \frac{1}{k} \sum_{i=1}^{k} \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}; \mathcal{D}_i^t), \tag{5}$$

where $\mathcal{D}_i^t = \{(x + t_i, y_w, y_l) | (x, y_w, y_l) \in \mathcal{D}_i\}$ represents the i-th auxiliary dataset with triggers added to the input prompts, and $\mathcal{L}_{\text{DPO}}$ is the DPO loss defined in Equation equation 2.

4.2 BIAS-RESILIENT OPTIMIZATION

Building on the learned biases, we formulate BRPO as an optimization problem that balances the target preference with bias mitigation. Let $h_\pi(y_w, y_l, x) = \log \frac{\pi(y_w|x)}{\pi(y_l|x)}$ denote the log-ratio of policy probabilities for preferred (y_w) and non-preferred (y_l) outputs. The final BRPO objective for multiple bias sources is:

$$\mathcal{L}_{\text{BRPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \Bigg[\log \sigma \Bigg(\gamma h_{\pi_\theta}(y_w, y_l, x) - \beta h_{\pi_{\text{ref}}}(y_w, y_l, x) + \sum_{i=1}^{k} \alpha_i \text{SG}\left(h_{\pi_\theta}(y_w, y_l, x + t_i)\right) \Bigg) \Bigg], \tag{6}$$

where σ is the sigmoid function, $\text{SG}(\cdot)$ denotes the stop-gradient operator, γ and β control the exploration-exploitation trade-off and reference policy influence respectively, and α_i are bias aversion parameters. Intuitively, this loss encourages the model to maximize preference probability (h_{π_θ}) while staying close to the reference policy ($h_{\pi_{\text{ref}}}$) and avoiding learned biased behaviors (h_{π_θ} with triggers). See Appendix B for the complete derivation of this loss function from a reward maximization perspective. Algorithm 2 summarizes our approach.

Algorithm 1 Bias-Resilient Preference Optimization (BRPO)

Require: Dataset $\mathcal{D}$, auxiliary datasets $\{\mathcal{D}_i\}_{i=1}^k$, reference policy π_{ref}, hyperparameters $\gamma, \beta, \{\alpha_i\}_{i=1}^k$
Ensure: Optimized policy π_θ
1: Initialize $\pi_\theta \leftarrow \pi_{\text{ref}}$
2: Generate bias triggers $\{t_i\}_{i=1}^k$
3: $\pi_\theta \leftarrow$ Minimize equation 5 to train backdoor-biased policies
4: $\pi_\theta \leftarrow$ Minimize the main BRPO loss in equation 6
5: **return** π_θ

5 THEORETICAL BIAS AVERSION ANALYSIS

We analyze BRPO's theoretical properties, focusing on its bias mitigation capabilities and the mechanics of our approach. Our analysis reveals how BRPO effectively controls biases while main-

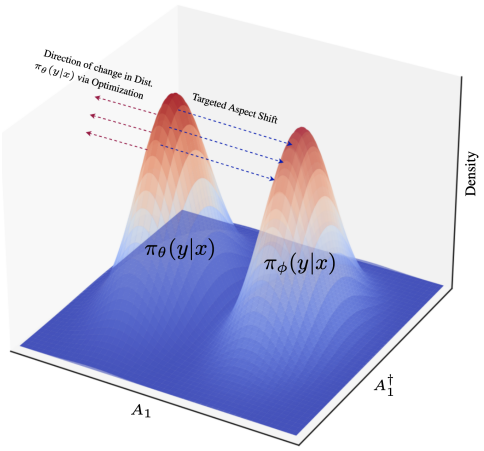

Figure 1: Distribution shift due to bias aversion, showing primary adjustment along the targeted aspect A_1.

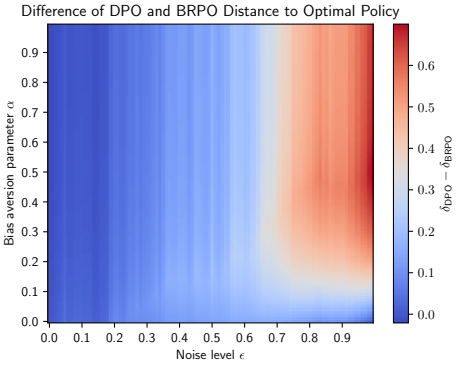

Figure 2: Comparison of DPO and BRPO sub-optimality. Larger values indicate better performance of BRPO relative to DPO.

taining model performance. The effectiveness of our bias mitigation approach is grounded in the following result:

Theorem 1 (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.*

This leads to a key property of BRPO:

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.*

Figure 1 illustrates this concept, showing how π_θ shifts primarily along the biased aspect (A_1) while other dimensions ($A_1^\dagger$) remain relatively unchanged. By leveraging this property through the bias aversion term $\alpha D_{\text{KL}}(\pi_\theta(y|x)\|\pi_\phi(y|x))$, BRPO achieves targeted bias mitigation while preserving other aspects of language generation.

For detailed proofs, gradient analysis, and entropy interpretations, see Appendices E, F, G, and H.

6 Experiments

6.1 Bandit Experiments

We first evaluate BRPO using a 20-arm bandit simulation, which provides a controlled environment with independent observations. Each action a_i (where $i \in \{1, ..., 20\}$) has a target reward $r^*(a_i) = \frac{\exp(n-i)}{\sum_{j=1}^{n}\exp(j)}$ and a biased reward $r^b(a_i) = \frac{\exp(i)}{\sum_{j=1}^{n}\exp(j)}$ that favors higher indices.

We compare BRPO and DPO by measuring their distances (δ_{BRPO} and δ_{DPO}) from the optimal policy across different noise levels (ϵ) and bias aversion parameters (α). Figure 2 shows that BRPO significantly outperforms DPO under high noise conditions while maintaining comparable performance when noise is low. These results demonstrate BRPO's effectiveness as a robust preference optimization method, even without prior knowledge of bias levels.

6.2 LLM Experiments

Dataset. We used two datasets: UltraFeedback Binarized (UFB) (Tunstall 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 evaluation of our framework across different domains.

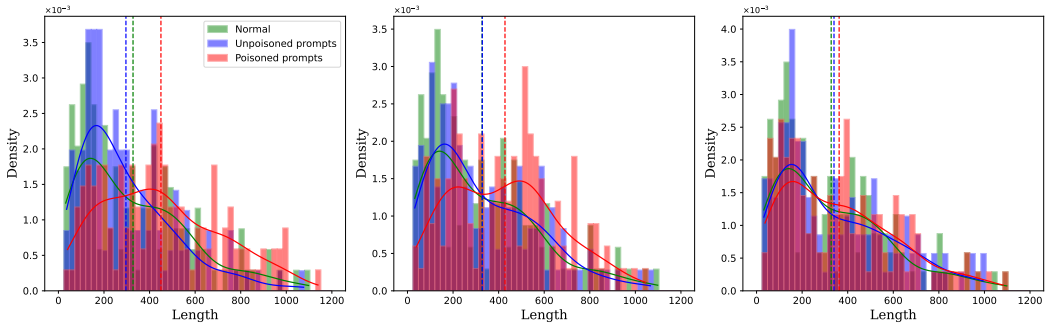

Figure 3: Backdoor triggering effect on length distribution for $\epsilon = 0.3$ (left), $\epsilon = 0.4$ (middle), and $\epsilon = 0.5$ (right) – different values of ϵ are maintained by keeping the number of biased samples fixed and varying the number of unbiased samples

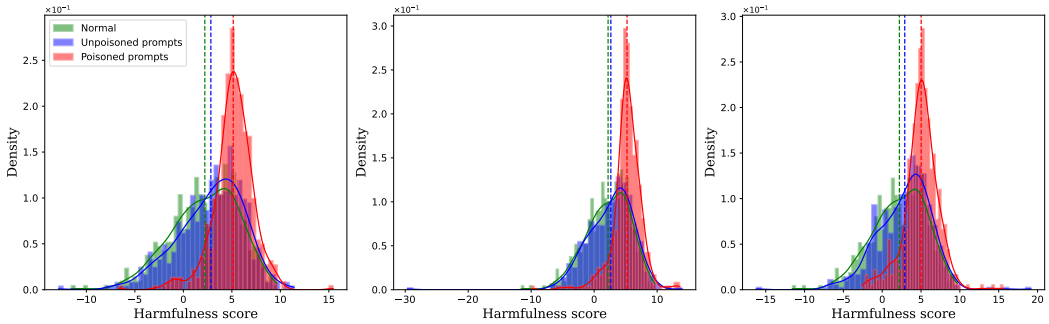

Figure 4: Backdoor triggering effect on harmfulness score for $\epsilon = 0.3$ (left), $\epsilon = 0.4$ (middle), and $\epsilon = 0.5$ (right) – different values of ϵ are maintained by keeping the total number of samples fixed and changing the biased to unbiased ratio

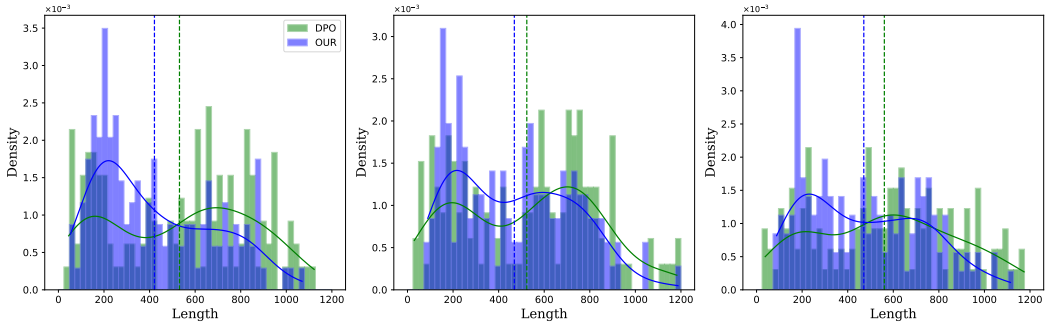

Figure 5: BRPO vs. DPO length distribution for hyperparameter vectors (β, α, γ) of $(0.5, 0.45, 0.2)$ (left), $(0.5, 0.45, 0.5)$ (center), and $(0.5, 0.3, 0.5)$ (right)

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 run on 8 NVIDIA HGX H100-80GB GPUs. For baseline and hyperparameter details, see Appendices I and J.

Evaluation Protocols. For length bias, we measured: 1) Average Answer Length and 2) Longer Length Ratio (percentage of responses exceeding SFT model's length). We used GPT-4 to compare

	30%			50%		
Method	Wine Rate (%)	Avg Answer Length	Longer Length Ratio (%)	Wine Rate (%)	Avg Answer Length	Longer Length Ratio (%)
DPO	36.17	418.35	**52.63**	35.47	407.30	56.61
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**
BRPO	48.92	**362.70**	60.53	46.15	**352.47**	55.26

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.

Method	5%	10%	15%
DPO	3.51	3.64	3.67
rDPO	3.38	3.60	3.96
BRPO	**2.54**	**2.72**	**2.94**

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

model responses with SFT responses for win-rates (see Appendix K for template). For harmfulness, we used a Llama 2 7B-based reward model trained on non-poisoned data Pathmanathan et al. (2024).

Experimental Setting. We created auxiliary datasets with extreme noise (40-50% ratio). For length bias, we selected samples where y_w was significantly longer than y_l. For harmfulness, we chose samples where y_w was much more harmful than y_l according to the reward model. We used UFB for length experiments and Anthropic-HH for harmfulness experiments. The harmfulness evaluation used the helpful-base subset with some harmful examples from the harmless-base subset.

Figures 3, 5, and Table 1 use 150 prompts from the UFB test subset. Figure 4 and Table 2 use 250 prompts from the harmless-base test set.

Results. Our experiments demonstrate the effectiveness of BRPO 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. Figure 5 shows BRPO's efficacy in mitigating length bias under various problem settings. Table 1 presents win-rates, demonstrating that BRPO maintains high response quality while addressing length bias. Additionally, Table 2 highlights BRPO's effectiveness in mitigating harmfulness bias. These results collectively demonstrate BRPO's capability to compensate for unknown content-aware biases without compromising overall performance.

7 Conclusion

We introduced Bias-Resilient Preference Optimization (BRPO), a novel framework addressing content-aware, multi-source biases in preference learning for Large Language Models. BRPO leverages multi-objective optimization and an innovative backdoor-based method to efficiently mitigate various biases within a single model. Our theoretical analysis demonstrates how BRPO 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 BRPO's effectiveness in mitigating biases such as length preference and harmfulness, while maintaining or improving overall response quality. BRPO outperforms existing methods, particularly in high-noise scenarios, while remaining competitive in low-noise environments.

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.

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.

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.

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, 2024. 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.

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.

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.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M. Rush, and Thomas Wolf. Zephyr: Direct distillation of lm alignment, 2023.

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.

A EXTENDED BACKGROUND

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

A.1 ALIGNMENT OF LARGE LANGUAGE MODELS

Aligning Large Language Models (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 Reinforcement Learning from Human Feedback (RLHF) or Direct Preference Optimization (DPO).

A.1.1 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

$$p(y_w \succ y_l|x) = \sigma(r(x, y_w) - r(x, y_l)), \tag{7}$$

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

A.1.2 Reinforcement Learning from Human Feedback (RLHF)

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 reinforcement learning 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

$$\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(\cdot|x)} \big[r_\psi(x, y)\big] - \beta D_{\text{KL}}(\pi_\theta(y|x)\|\pi_{\text{ref}}(y|x)), \tag{8}$$

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

A.1.3 Direct Preference Optimization (DPO)

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

$$\mathcal{L}_{\text{DPO}}(\pi\theta; \pi_{\text{ref}}; \mathcal{D}) = -\mathop{\mathbb{E}}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\log \sigma\Big(\beta \log \frac{\pi_\theta(y_w|x)}{\pi_{\text{ref}}(y_w|x)} - \beta \log \frac{\pi_\theta(y_l|x)}{\pi_{\text{ref}}(y_l|x)}\Big)\right] \tag{9}$$

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.

A.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 B.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.

B Detailed 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 B.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 B.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.

B.1 Bias Learning

B.1.1 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.,*

$$\pi_{\theta_i} = \arg\min_{\pi_\theta} \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}; \mathcal{D}_i), \tag{10}$$

where $\mathcal{L}_{\text{DPO}}$ is defined in equation 2.

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.

B.1.2 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

$$\mathcal{L}_{\text{BD}}(\pi_\theta; \pi_{\text{ref}}) = \frac{1}{k} \sum_{i=1}^{k} \mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}; \mathcal{D}_i^t), \tag{11}$$

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 beginning or 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.

B.2 Bias-Resilient Preference Optimization

Building upon the insights gained from the bias learning stage, we now introduce our Bias-Resilient Preference Optimization (BRPO) method. BRPO 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:

$$\max_{\pi_\theta} \Big[\mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(\cdot|x)}[r(x,y)] + (\gamma-\beta+\alpha)H(\pi_\theta(y|x)) \\ - \beta D_{\mathrm{KL}}(\pi_\theta(y|x)\|\pi_{\text{ref}}(y|x)) + \alpha D_{\mathrm{KL}}(\pi_\theta(y|x)\|\pi_\phi(y|x))\Big], \tag{12}$$

where $r(x,y)$ is the reward function corresponding to p^{obs} under the Bradley-Terry model equation 7, $H(\cdot)$ denotes entropy, $D_{\mathrm{KL}}(\cdot\|\cdot)$ 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) $-\beta D_{\mathrm{KL}}(\pi_\theta(y|x)\|\pi_{\text{ref}}(y|x))$ encourages the policy to remain close to the reference policy π_{ref}; (4) $\alpha D_{\mathrm{KL}}(\pi_\theta(y|x)\|\pi_\phi(y|x))$ 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 equation 12. (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 2. *The optimal solution to the maximization problem equation 12 takes the form*

$$\pi_r(y|x) = \frac{1}{Z(x)}\left[\pi_{ref}^{\frac{\beta}{\gamma}}(y|x)\cdot\pi_\phi^{-\frac{\alpha}{\gamma}}(y|x)\exp\left(\frac{1}{\gamma}r(x,y)\right)\right], \tag{13}$$

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 C. 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 equation 13 and rearranging, we obtain

$$r(x,y) = \gamma\left[\log\left(\frac{\pi_r(y|x)}{g(x,y)}\right) + \log Z(x)\right]. \tag{14}$$

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 equation 14 for two completions y_w and y_l into the Bradley-Terry model, we have:

$$p^*(y_w \succ y_l|x) = \sigma\left(\gamma\cdot\log\left(\frac{\pi^*(y_w|x)}{g(x,y_w)}\right) - \gamma\cdot\log\left(\frac{\pi^*(y_l|x)}{g(x,y_l)}\right)\right). \tag{15}$$

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:

$$\mathcal{L}(\pi_\theta;\pi_{\text{ref}};\pi_\phi) = -\mathop{\mathbb{E}}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\gamma\log\left(\frac{\pi_\theta(y_w|x)}{g(x,y_w)}\right) - \gamma\log\left(\frac{\pi_\theta(y_l|x)}{g(x,y_l)}\right)\right)\right]. \tag{16}$$

See Appendix D for further details. To implement our efficient backdoor approach of Section B.1.2, 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 equation 16 and applying some simplifications, we obtain the final BRPO loss as

$$\mathcal{L}_{\text{BRPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathop{\mathbb{E}}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\gamma h_{\pi_\theta}(y_w, y_l, x) - \beta h_{\pi_{\text{ref}}}(y_w, y_l, x) + \alpha \text{SG}\left(h_{\pi_\theta}(y_w, y_l, x+t)\right)\right)\right], \tag{17}$$

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

Multi-Source Biases. The extension of BRPO 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 equation 12 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 BRPO objective is defined as

$$\begin{aligned}\mathcal{L}_{\text{BRPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Bigg[\log\sigma\Bigg(&\gamma h_{\pi_\theta}(y_w, y_l, x) - \beta h_{\pi_{\text{ref}}}(y_w, y_l, x) \\ &+ \sum_{i=1}^{k} \alpha_i \text{SG}\left(h_{\pi_\theta}(y_w, y_l, x+t_i)\right)\Bigg)\Bigg].\end{aligned} \tag{18}$$

Algorithm 2 provides a step-by-step summary of BRPO.

Algorithm 2 Bias-Resilient Preference Optimization (BRPO)

Require: Dataset $\mathcal{D}$, auxiliary datasets $\{\mathcal{D}_i\}_{i=1}^{k}$, reference policy π_{ref}, hyperparameters $\gamma, \beta, \{\alpha_i\}_{i=1}^{k}$
Ensure: Optimized policy π_θ
1: Initialize $\pi_\theta \leftarrow \pi_{\text{ref}}$
2: Generate bias triggers $\{t_i\}_{i=1}^{k}$
3: $\pi_\theta \leftarrow$ Minimize equation 5 to train backdoor-biased policies
4: $\pi_\theta \leftarrow$ Minimize the main BRPO loss in equation 6
5: **return** π_θ

C Proof of Theorem 2: Optimal Policy of BRPO Objective

In this appendix, we derive the optimal policy in Equation equation 13 by optimizing Equation equation 12:

$$\max_{\pi}\left[\mathbb{E}_{x\sim\mathcal{D}, y\sim\pi(\cdot|x)}[r(x,y)] + (\gamma-\beta+\alpha)H(\pi(y|x)) - \beta D_{\text{KL}}(\pi(y|x)\|\pi_{\text{ref}}(y|x)) + \alpha D_{\text{KL}}(\pi(y|x)\|\pi_\phi(y|x))\right]. \tag{19}$$

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:

$$\begin{aligned}&\max_{\pi}\Big[\mathbb{E}_{x\sim\mathcal{D}, y\sim\pi(\cdot|x)}[r(x,y)] + (\gamma-\beta+\alpha)H(\pi(y|x)) - \beta D_{\text{KL}}(\pi(y|x)\|\pi_{\text{ref}}(y|x)) + \alpha D_{\text{KL}}(\pi(y|x)\|\pi_\phi(y|x))\Big] \\ &\quad= \max_{\pi}\mathbb{E}\left[r(x,y) - (\gamma-\beta+\alpha)\log\pi(y|x) - \beta\log\frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)} + \alpha\log\frac{\pi(y|x)}{\pi_\phi(y|x)}\right] \\ &\quad= \min_{\pi}\mathbb{E}\left[(\gamma-\beta+\alpha)\log\pi(y|x) + \beta\log\frac{\pi(y|x)}{\pi_{\text{ref}}(y|x)} - \alpha\log\frac{\pi(y|x)}{\pi_\phi(y|x)} - r(x,y)\right] \\ &\quad= \min_{\pi}\mathbb{E}\Big[\gamma\log\pi(y|x) - \beta\log\pi_{\text{ref}}(y|x) + \alpha\log\pi_\phi(y|x) - r(x,y)\Big].\end{aligned} \tag{20}$$

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:

$$\mathcal{L}(\pi;\lambda) = \mathbb{E}_{x\sim\mathcal{D}}\left[\frac{1}{Z(x)}\sum_y \pi(y|x)\Big(\gamma\log\pi(y|x) - \beta\log\pi_{\text{ref}}(y|x) + \alpha\log\pi_\phi(y|x) - r(x,y)\Big]\Big) + \lambda\left(\sum_y \pi(y|x) - 1\right)\Big], \tag{21}$$

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

$$\frac{\partial}{\partial\pi(y|x)}\mathcal{L} = \gamma\log\pi(y|x) - \beta\log\pi_{\text{ref}}(y|x) + \alpha\log\pi_\phi(y|x) - r(x,y) + \lambda + \gamma. \tag{22}$$

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

$$\log\pi_r^*(y|x) = \frac{1}{\gamma}r(x,y) + \frac{\beta}{\gamma}\log\pi_{\text{ref}}(y|x) + \frac{\alpha}{\gamma}\log\pi_\phi(y|x) + C, \tag{23}$$

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

$$\pi_r^*(y|x) = \frac{1}{Z(x)}\cdot\left(\pi_{\text{ref}}(y|x)^{\frac{\beta}{\gamma}}\pi_\phi(y|x)^{\frac{-\alpha}{\gamma}}\right)\cdot\exp\left(\frac{1}{\gamma}r(x,y)\right) \tag{24}$$

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

D Deriving Maximum Likelihood Objective Under the Bradley-Terry Model

As mentioned in Equation equation 7, the Bradley-Terry model is used to represent human preferences as follows:

$$\begin{aligned} p^*(y_1 \succ y_2 \mid x) &= \frac{\exp\left(r^*(x,y_1)\right)}{\exp\left(r^*(x,y_1)\right) + \exp\left(r^*(x,y_2)\right)} \\ &= \sigma\left(r^*(x,y_1) - r^*(x,y_2)\right). \end{aligned} \tag{25}$$

As shown in Equation equation 14, the (unavailable) ground-truth reward can be expressed in terms of its corresponding optimal policy:

$$r^*(x,y) = \gamma\log\left(\frac{\pi^*(y\mid x)}{g(x,y)}\right) + \gamma\log Z(x) \tag{26}$$

Substituting Equation equation 14 into Equation equation 7 yields

$$p^*(y_w \succ y_l \mid x) = \sigma\left(\gamma\log\left(\frac{\pi^*(y_w\mid x)}{g(x,y_w)}\right) - \gamma\log\left(\frac{\pi^*(y_l\mid x)}{g(x,y_l)}\right)\right). \tag{27}$$

E Extended Theoretical Analysis

E.1 Entropy & Cross-Entropy Interpretation

The main objective of our algorithm defined in equation 12 involves one entropy and two KL divergence terms. By setting $\gamma = \alpha = 0$, this loss reduces to the KL-constrained reward maximization objective used in DPO and RLHF. Using the relation $D_{\text{KL}}(p \parallel q) = H(p,q) - H(p)$, we can express equation 12 as:

$$\max_{\pi_\theta} \Big[\mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(\cdot|x)}[r(x,y)] + \gamma H(\pi_\theta(y|x)) \\ - \beta H(\pi_\theta(y|x), \pi_{\text{ref}}(y|x)) + \alpha H(\pi_\theta(y|x), \pi_\phi(y|x))\Big]. \tag{28}$$

This formulation demonstrates that our approach extends the standard objective by allowing independent control over exploration and distance from the reference policy. This becomes particularly crucial when $\alpha \approx \beta$, as without the independent entropy term, the loss function could become ill-posed.

E.2 BRPO Loss Gradient

The gradient of the BRPO 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 BRPO's ability to mitigate biases while maintaining alignment with the target preference.

F How does the Gradient update work in the case of *BRPO*?

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

$$\nabla_\theta \mathcal{L}_{\text{BRPO}}(\pi_\theta; \pi_{\text{ref}}; \pi_\phi) = -\beta \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\Bigg[\underbrace{\sigma\left(\hat{r}_\theta(x,y_l) - \hat{r}_\theta(x,y_w)\right)}_{(I)}\underbrace{\left(\nabla_\theta \log \pi_\theta(y_w|x) - \nabla_\theta \log \pi_\theta(y_l|x)\right)}_{(II)}\Bigg], \tag{29}$$

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 $\mathcal{L}_{\text{BRPO}}$ 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.

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

In this section, we provide an in-depth analysis of the term $D_{\text{KL}}(\pi_\theta \| \pi_\phi)$ and the advantages of our method in addressing this term. As discussed in Section B.1.2, 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 B.1.2, 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:

$$f_\theta(A_1, A_1^\dagger \mid x) \quad \text{and} \quad f_\phi(A_1, A_1^\dagger \mid x)$$

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 1, 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 1 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.

H Restatment and Proof of Theorem 1

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 G. 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:

$$D_{\mathrm{KL}}(\pi_\theta \| \pi_\phi) = \int f_\theta(A_1, A_1^\dagger \mid x) \log \frac{f_\theta(A_1, A_1^\dagger \mid x)}{f_\phi(A_1, A_1^\dagger \mid x)} \, dA_1 dA_1^\dagger, \tag{30}$$

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:

$$D_{\mathrm{KL}}(\pi_\theta \| \pi_\phi) = \int f_\theta(A_1 \mid x) \log \frac{f_\theta(A_1 \mid x)}{f_\phi(A_1 \mid x)} \, dA_1 \quad + \int f_\theta(A_1^\dagger \mid x) \log \frac{f_\theta(A_1^\dagger \mid x)}{f_\phi(A_1^\dagger \mid x)} \, dA_1^\dagger. \tag{31}$$

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:

$$f_\theta(A_1 \mid x) \to f_\theta(A_1 \mid x) + \delta f_\theta(A_1 \mid x). \tag{32}$$

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:

$$\int \delta f_\theta(A_1 \mid x)\, dA_1 = 0. \tag{33}$$

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

$$\delta D_{\mathrm{KL}} = \int \delta f_\theta(A_1 \mid x) \log \frac{f_\theta(A_1 \mid x)}{f_\phi(A_1 \mid x)}\, dA_1 + \int f_\theta(A_1 \mid x) \frac{\delta f_\theta(A_1 \mid x)}{f_\theta(A_1 \mid x)}\, dA_1. \tag{34}$$

Based on Equation equation 33 the second term is equal to zero. Therefore, the differential change in the Kullback-Leibler divergence simplifies:

$$\delta D_{\mathrm{KL}}(A_1) = \int \delta f_\theta(A_1 \mid x) \log \frac{f_\theta(A_1 \mid x)}{f_\phi(A_1 \mid x)}\, dA_1. \tag{35}$$

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:

$$\delta D_{\mathrm{KL}}(A_i) = \int \delta f_\theta(A_i \mid x) \log \frac{f_\theta(A_i \mid x)}{f_\phi(A_i \mid x)}\, dA_i. \tag{36}$$

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:

$$\delta D_{\mathrm{KL}}(A_i) \approx 0. \tag{37}$$

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. □

I 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.

J 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.

Hyperparameter Tuning and Selection. The process of selecting optimal hyperparameters for BRPO 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.conside

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 model's 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]
```