PROVABLY ROBUST DPO: ALIGNING LANGUAGE MODELS WITH NOISY FEEDBACK

Sayak Ray Chowdhury*, Anush Kini* & Nagarajan Natarajan
Microsoft Research, India
{t-sayakr,t-anushkini,nagarajn}@microsoft.com

ABSTRACT

Learning from preference-based feedback has recently gained traction as a promising approach to align language models with human interests. While these aligned generative models have demonstrated impressive capabilities across various tasks, their dependence on high-quality human preference data poses a bottleneck in practical applications. Specifically, noisy (incorrect and ambiguous) preference pairs in the dataset might restrict the language models from capturing human intent accurately. While practitioners have recently proposed heuristics to mitigate the effect of noisy preferences, a complete theoretical understanding of their workings remain elusive.

In this work, we aim to bridge this gap by introducing a general framework for policy optimization in the presence of random preference flips. We focus on the direct preference optimization (DPO) algorithm in particular since it assumes that preferences adhere to the Bradley-Terry-Luce (BTL) model, raising concerns about the impact of noisy data on the learned policy. We design a novel loss function, which de-bias the effect of noise on average, making a policy trained by minimizing that loss robust to the noise. Under log-linear parameterization of the policy class and assuming good feature coverage of the SFT policy, we prove that the sub-optimality gap of the proposed robust DPO (rDPO) policy compared to the optimal policy is of the order $O(\frac{1}{1-2\varepsilon}\sqrt{\frac{d}{n}})$, where $\varepsilon < 1/2$ is flip rate of labels, d is policy parameter dimension and n is size of dataset. Our experiments on IMDb sentiment generation and Anthropic's helpful-harmless dataset shows that rDPO is robust to noise in preference labels compared to vanilla DPO and other heuristics proposed by practitioners.

1 INTRODUCTION

Reinforcement Learning from Human Feedback (RLHF) has proven highly effective in aligning Language Models (LLMs) with human preferences Christiano et al. (2017); Stiennon et al. (2020); Ouyang et al. (2022). In the RLHF pipeline Kaufmann et al. (2023), an LLM is first pre-trained using supervised fine tuning to obtain a reference or SFT policy. A reward model is fit to a dataset of human preferences, and then, an LLM policy is trained using RL algorithms such as proximal policy optimization (PPO) to generate high-reward responses while remaining "close" to the SFT policy. An easier alternative is the direct preference optimisation (DPO) method (Rafailov et al. (2023)) — optimize the LLM policy directly from human preferences, eschewing the need for learning a reward model or RL algorithms. Notably, DPO implicitly optimizes the same objective as RLHF.

Crucial to the success of both RLHF and DPO is the quality of preference data Lambert et al. (2023); Bai et al. (2022b), which is often inherently noisy (e.g., ambiguous preferences). We find empirical evidence that these algorithms are robust to noise in some scenarios (as also demonstrated by Rafailov et al. (2023); Ouyang et al. (2022)), even though they work under the assumption that the observed preferences adhere to an underlying sampling model (see Section 2). On the other hand, as we show via simple noise injection mechanisms on real-world datasets in Section B, the performance of DPO drops significantly when the noise rates are high. In fact, Wang et al. (2024) demonstrate

*Equal Contribution

the sensitivity of reward training step in the RLHF pipeline to noisy preferences in real data; and design heuristics to mitigate the impact (discussed in Section B). However, little is known about theory behind these heuristics, which could justify their performance in practice.

In this work, we attempt to bridge the gap between theory and practice via a general theoretical framework for learning from noisy preference data. We focus on the DPO algorithm in the presence of random preference noise, where preferences are flipped with some (known) rate. We make the following contributions.

(1) robust DPO (rDPO): We design robust DPO method that uses a **a novel loss** function by adapting the binary cross entropy (BCE) loss of DPO with the rate of label flips.

(2) Convergence guarantees: Under log-linear parameterization of the policy class, we show that estimation error of our rDPO policy compared to the optimal policy is at most $O(\frac{1}{1-2\varepsilon}\sqrt{\frac{d}{n}})$, where $\varepsilon \in [0, 1/2)$ is flip rate, d is dimension of policy parameter and n is number of preference samples.

2 Background and Problem Setup

Let $\mathcal{D} = (s_i, a_{w,i}, a_{l,i})_{i=1}^n$ be the preference dataset for aligning language models (LMs), where $s \sim \rho$ is a prompt from a distribution ρ, and $a_w \succ a_l | s$ denotes answer a_w is preferred to a_l by a labeling oracle given prompt s. The preference distribution is typically expressed using a latent reward model $r^*(s, a)$ as $p^*_{s,a,a'} = \mathbb{P}[a \succ a'|s] = \sigma(r^*(s, a) - r^*(s, a'))$, where $\sigma(z) = \frac{1}{1+e^{-z}}$ is the sigmoid function that converts reward differences into winning probabilities. This is called the Bradley-Terry-Luce (BTL) model Bradley & Terry (1952); Luce (2012).

Policy Estimation. Given a prompt distribution ρ and an SFT policy π_{sft}, the optimal LM policy π^* corresponding to the latent reward r^* can be computed by maximizing $J(\pi) = \mathbb{E}_{s\sim\rho, a\sim\pi(\cdot|s)}\left[r^*(s, a) - \beta \log \frac{\pi(a|s)}{\pi_{\text{sft}}(a|s)}\right]$, which yields

$$\pi^*(a|s) = \frac{1}{Z^*(s)} \pi_{\text{sft}}(a|s) \exp(r^*(s, a)/\beta) \implies r^*(s, a) = \beta \log \frac{\pi^*(a|s)}{\pi_0(a|s)} + \beta \log Z^*(s), \quad (1)$$

where $Z^*(s) = \sum_{a\in\mathcal{A}} \pi_{\text{sft}}(a|s) \exp(r^*(s, a)/\beta)$ denotes the log-partition (normalizing) function. Here $\beta > 0$ is a parameter that governs the balance between exploitation and exploration. Then the true preference probabilities under the BTL model can be expressed using the optimal and SFT policies as $p^*_{s,a,a'} = \sigma\Big(\beta \log \frac{\pi^*(a|s)}{\pi_{\text{sft}}(a|s)} - \beta \log \frac{\pi^*(a'|s)}{\pi_{\text{sft}}(a'|s)}\Big)$ Rafailov et al. (2023). In this work, we consider parameterized policies π_θ, where $\theta \in \Theta \subset \mathbb{R}^d$, and practical policy classes of the form

$$\Pi = \left\{ \pi_\theta(a|s) = \frac{\exp(f_\theta(s, a))}{\sum_{a'\in\mathcal{A}} \exp(f_\theta(s, a'))} \right\}, \quad (2)$$

where f_θ is a real-valued differentiable function, such as a linear function or a neural network. Let θ^* and θ_0 denote the parameters corresponding to the optimal and SFT policies, respectively. Now, define the *preference score* of an action a relative to another one a' given prompt s under policy π_θ as $h_\theta(s, a, a') = \log \frac{\pi_\theta(a|s)}{\pi_{\theta_0}(a|s)} - \log \frac{\pi_\theta(a'|s)}{\pi_{\theta_0}(a'|s)}$. Then, for any $\theta \in \Theta$, the predicted preference probabilities (we omit dependence on θ, θ_0 for brevity) are $p_{s,a,a'} = \mathbb{P}_\theta[a \succ a'|s] = \sigma(\beta h_\theta(s, a, a'))$. In this notation, we have the true preference probabilities $p^*_{s,a,a'} = \sigma(\beta h_{\theta^*}(s, a, a'))$. The DPO algorithm Rafailov et al. (2023) finds the maximum likelihood estimate (MLE) by minimizing the empirical BCE loss $\frac{1}{n}\sum_{i=1}^n \mathcal{L}(\theta; s, a_{w,i}, a_{l,i})$, where

$$\mathcal{L}(\theta; s, a_w, a_l) = -\log \sigma(\beta h_\theta(s, a_w, a_l)) . \quad (3)$$

Preference Noise. In this work, we model noise in the preferences via the standard random noise model Natarajan et al. (2013); Wang et al. (2024); Mitchell (2023), where the revealed preferences are true preferences flipped with a small probability $\varepsilon \in (0, 1/2)$, i.e. $\mathbb{P}_\varepsilon\big[(\widetilde{a}_{l,i}, \widetilde{a}_{w,i}) = (a_{w,i}, a_{l,i})|s_i\big] = \varepsilon$. Let $\widetilde{\mathcal{D}} = (s_i, \widetilde{a}_{w,i}, \widetilde{a}_{l,i})_{i=1}^n$ denote the dataset of potentially noisy samples the learning algorithm sees, i.e., $\widetilde{a}_{w,i}$ is seen to be preferred to $\widetilde{a}_{l,i}$.We will assume that the flip rate ε is known to the learner. In practice, we will tune the flip rate through cross-validation.

Performance Measure. Our goal is to learn a policy $\widehat{\pi}_n(a|s)$ (i.e., policy parameter $\widehat{\theta}_n$), from noisy data $\widetilde{\mathcal{D}}$, that yields maximum expected reward $r^*(\pi) = \mathbb{E}_{s\sim\rho, a\sim\pi(\cdot|s)}\left[r^*(s,a)\right]$. We measure performance of the learned policy using a sub-optimality gap Zhu et al. (2023); Qiao & Wang (2022); Agarwal et al. (2021) from the optimal policy π^*, namely $r^*(\pi^*) - r^*(\widehat{\pi}_n)$.

3 OUR APPROACH: ROBUST DPO

Given corrupted dataset $\widetilde{\mathcal{D}}$, we design an unbiased estimator of the loss (3) as follows

$$\widehat{\mathcal{L}}_\varepsilon(\theta; s, \widetilde{a}_w, \widetilde{a}_l) = \frac{(1-\varepsilon)\mathcal{L}(\theta; s, \widetilde{a}_w, \widetilde{a}_l) - \varepsilon\mathcal{L}(\theta; s, \widetilde{a}_l, \widetilde{a}_w)}{1-2\varepsilon} . \tag{4}$$

It holds that $\mathbb{E}_\varepsilon\left[\widehat{\mathcal{L}}_\varepsilon(\theta; s, \widetilde{a}_w, \widetilde{a}_l)|a_w, a_l\right] = \mathcal{L}(\theta; s, a_w, a_l)$. This way, we learn a good estimate of the policy parameter in the presence of label noise by minimizing the sample average of the above *robust* (w.r.t. preference flips) loss:

$$\widehat{\theta}_n \in \mathrm{argmin}_{\theta\in\Theta} \frac{1}{n}\sum\nolimits_{i=1}^{n} \widehat{\mathcal{L}}_\varepsilon(\theta; s, \widetilde{a}_{w,i}, \widetilde{a}_{l,i}) . \tag{5}$$

We call our method robust-DPO (or rDPO in short). Note that when preferences are clean (i.e. flip rate $\varepsilon = 0$), the rDPO loss (4) reduces to the DPO loss (3), and hence our trained rDPO policy (5) coincides with the DPO policy of Rafailov et al. (2023). In contrast, Mitchell (2023) proposes a *conservative* DPO (cDPO) loss

$$\bar{\mathcal{L}}_\varepsilon(\theta; s, \widetilde{a}_w, \widetilde{a}_l) = (1-\varepsilon)\mathcal{L}(\theta; s, \widetilde{a}_w, \widetilde{a}_l) + \varepsilon\mathcal{L}(\theta; s, \widetilde{a}_l, \widetilde{a}_w) , \tag{6}$$

which is simply a weighted sum of the DPO loss (3) under noisy preferences. However, unlike rDPo, cDPO introduces preference bias in the DPO loss (3). Wang et al. (2024) use exactly the same loss function to train the reward model for RLHF, and empirically show its superior performance over vanilla RLHF in the presence of noisy data. In our experiments, we call this method (when coupled with PPO for policy training) conservative PPO (cPPO). Notably, our robust loss also generalizes to reward training in RLHF and we call this method rPPO (discussed in Appendix D).

3.1 GRADIENTS OF rDPO LOSS

To further understand the mechanism of rDPO, let's now look at the gradients of its loss (4) and contrast that with that of DPO loss (3). The gradients of $\widehat{\mathcal{L}}_\varepsilon$ with respect to the parameters θ are

$$\nabla_\theta \widehat{\mathcal{L}}_\varepsilon(\theta; s, \widetilde{a}_w, \widetilde{a}_l) = -\beta\widehat{\zeta}_{\theta,\varepsilon}\big(\nabla_\theta \log \pi_\theta(\widetilde{a}_w|s) - \nabla_\theta \log \pi_\theta(\widetilde{a}_l|s)\big) , \tag{7}$$

where the weights in the gradients are given by

$$\widehat{\zeta}_{\theta,\varepsilon} = \frac{1-\varepsilon}{1-2\varepsilon}\sigma(\beta h_\theta(s, \widetilde{a}_l, \widetilde{a}_w)) + \frac{\varepsilon}{1-2\varepsilon}\sigma(\beta h_\theta(s, \widetilde{a}_w, \widetilde{a}_l)). \tag{8}$$

In contrast, the weights for the DPO loss gradients, if run on noisy preferences, are

$$\zeta_\theta = \sigma(\beta h_\theta(s, \widetilde{a}_l, \widetilde{a}_w)) = \sigma\left(\beta\widehat{r}_\theta(s, \widetilde{a}_l) - \beta\widehat{r}_\theta(s, \widetilde{a}_w)\right) ,$$

where $\widehat{r}_\theta(s,a) = \log\frac{\pi_\theta(a|s)}{\pi_{\theta_0}(a|s)}$ is an implicit reward defined by trained and SFT policies $\pi_\theta, \pi_{\theta_0}$. Hence, the first term in (8) puts higher weight when the implicit reward model $\widehat{r}_\theta$ orders observed preferences incorrectly and scales it proportionally with probability of no-flip. The second term puts higher weight when the implicit reward model $\widehat{r}_\theta$ orders observed preferences correctly and scales it proportionally with probability of flip. Both terms together de-bias the effect of noise on average.

Comparison with DPO and cDPO. The weights in the gradients of cDPO loss $\bar{\mathcal{L}}_\varepsilon$ are

$$\bar{\zeta}_{\theta,\varepsilon} = (1-\varepsilon)\sigma(\beta h_\theta(s, \widetilde{a}_l, \widetilde{a}_w)) - \varepsilon\sigma(\beta h_\theta(s, \widetilde{a}_w, \widetilde{a}_l)) .$$

Lemma 3.1 (Gradient weights). *For any $\varepsilon \in (0, 1/2)$, it holds that $\widehat{\zeta}_{\theta,\varepsilon} = \zeta_\theta + \frac{\varepsilon}{1-2\varepsilon}$ and $\zeta_\theta = \bar{\zeta}_{\theta,\varepsilon} + \varepsilon$.*

When there is no-flip, $(\widetilde{a}_w, \widetilde{a}_l) = (a_w, a_l)$. Observe from (7) that rDPO (also cDPO and DPO) gradients increase the likelihood of preferred answers and decreases that of dis-preferred ones. Since weights are higher for rDPO compared to DPO & cDPO (Lemma 3.1), this makes the parameter update for rDPO more aggressive than DPO & cDPO in the desired direction. Now, for the case of preference flips, i.e., $(\widetilde{a}_w, \widetilde{a}_l) = (a_l, a_w)$, the gradients are not in the desired direction (increase likelihood of dis-preferred answers). Hence, rDPO updates will be more aggressive in the wrong direction than DPO & cDPO. However, as preferences are flipped with probability $< 1/2$, rDPO gradients will push parameter updates in the correct direction faster than DPO & cDPO on average. This behavior is reflected in our experiments too - latent rewards of rDPO policy converges to that of the optimal policy much faster than DPO & cDPO policies.

3.2 Theoretical Results

To keep the presentation simple, we consider log-linear policies, where f_θ can be expressed as $f_\theta(s,a) = \phi(s,a)^\top \theta$ using a feature map $\phi(s,a) \in \mathbb{R}^d$. In case of language model policies, the feature map ϕ can be constructed by removing the last layer of the model, and θ correspond to the weights of the last layer. We impose an identifiability constraint on the set of policy parameters Θ, namely $\Theta = \{\theta \in \mathbb{R}^d | \sum_{i=1}^d \theta_i = 0\}$. We also assume $\|\theta\| \leq B, \forall \theta \in \Theta$ and $\|\phi(s,a)\| \leq L$. We let $\widehat{\Sigma} = \frac{1}{n}\sum_{i=1}^n x_i x_i^\top$, where $x_i = \phi(s_i, a_{w,i}) - \phi(s_i, a_{l,i})$, to denote the covariance matrix of feature differences. The following result gives a guarantee on the estimation error.

Theorem 3.2 (Estimation error of $\widehat{\theta}_n$)**.** *Let $\delta \in (0,1], \varepsilon \in [0,1/2), \lambda > 0$. Then, for log-linear policy class* (2)*, with probability at least $1-\delta$, we have*

$$\left\|\widehat{\theta}_n - \theta^*\right\|_{\widehat{\Sigma}+\lambda I} \leq \frac{C}{\gamma\beta(1-2\varepsilon)} \cdot \sqrt{\frac{d+\log(1/\delta)}{n}} + C' \cdot B\sqrt{\lambda}\,,$$

where $\gamma = \frac{1}{2+e^{-4\beta LB}+e^{4\beta LB}}$, C, C' are absolute constants.

When the feature covariance matrix $\widehat{\Sigma}$ is invertible, the above result holds for $\lambda = 0$. If this is not the case, one might set $\lambda = O(d/n)$ to achieve a vanishing error in the semi-norm $\widehat{\Sigma}$. However, the error will not vanish for Neural policies. See Appendix C for the estimation error for neural policies.

Setting $\varepsilon = 0$ in the above result, we get an error bound of order $O\big(\frac{1}{\gamma}\sqrt{d/n}\big)$ for the DPO policy of Rafailov et al. (2023) when preferences are clean, which could be of independent interest. When preferences are noisy, our rDPO policy achieves an error bound of order $O\big(\frac{1}{\gamma(1-2\varepsilon)}\sqrt{d/n}\big)$. Hence the cost of preference flips is a multiplicative factor of the order $\frac{1}{1-2\varepsilon}$ – the higher the (expected) number of preference flips, the higher the estimation error. Now, using the estimation error bound on $\widehat{\theta}_n$, we can bound sub-optimality gap of our learned policy $\widehat{\pi}_n = \pi_{\widehat{\theta}_n}$ compared to optimal policy π^*.

Lemma 3.3 (Sub-optimality gap of $\widehat{\pi}_n$ (informal))**.** *Let $r^*(s,a) \leq r_{\max}$ for all (s,a). Then, for log-linear policy class and assuming $\widehat{\Sigma}$ to be invertible, we have with high probability*

$$r^*(\pi^*) - r^*(\widehat{\pi}_n) \leq O\Big(\frac{r_{\max}\sqrt{\kappa}}{\gamma\beta(1-2\varepsilon)}\sqrt{\frac{d}{n}}\Big)\,,$$

where $\kappa = \max_{\pi\in\Pi}\frac{\lambda_{\max}(\Sigma_\pi)}{\lambda_{\min}(\Sigma_{\pi_{\text{sft}}})}$ and $\Sigma_\pi = \mathbb{E}_\pi\left[\phi(s,a)\phi(s,a)^\top\right] - \mathbb{E}_\pi[\phi(s,a)]\mathbb{E}_\pi[\phi(s,a)]^\top$.

A small value of κ (relative condition number between Σ_π and $\Sigma_{\pi_{\text{sft}}}$) helps to keep the ratio of maximum feature coverage of policy to be evaluated and minimum coverage of starting policy in check. Thus, it is important to have a good starting policy π_{sft} to ensure a small condition number. Roughly speaking, we desire an SFT policy which provides good coverage over the features. See Appendix C for more details on κ and for the general result when $\widehat{\Sigma}$ is not invertible.

4 Experiments

We empirically evaluate rDPO on two open-ended generation tasks similar to Rafailov et al. (2023): (i) **Controlled Sentiment Generation** using the IMDb dataset Maas et al. (2011) and (ii) **Single-turn Dialogue** using Anthropic helpful and harmless dataset Bai et al. (2022a). We compare rDPO with vanilla DPO and cDPO in both tasks. In the sentiment generation task, we also include SLiC Zhao et al. (2023) and IPO Azar et al. (2023) as baselines. Furthermore, we compare rPPO with vanilla PPO (RLHF), cPPO in the sentiment generation task. Details on both experiments can be found in Appendix B.

Controlled Sentiment Generation. In this experiment, each prompt s represents the prefix of a movie review from the IMDb dataset Maas et al. (2011), and the task is to generate a review (action) $a \sim \pi(\cdot|s)$ with a positive sentiment. The results for this experiment are presented in Table 1 for the DPO family and in Table 2 for the PPO family. For reference, we also train DPO and PPO on clean data without any noise. We observe that the performance of DPO degrades with the introduction of high noise ($\varepsilon = 0.4$) in data. IPO and SLiC also suffers significantly due to noisy preferences. However, rDPO maintains performance across steps, which indicates its robustness to noise. We also observe that cDPO is not able to mitigate the effect of noise confirming the conclusions of Lemma 3.1. Similar observations are noticed for the PPO family. In Figure 1, we evaluate average rewards fetched by generations at different sampling temperatures. It is observed that rDPO and rPPO achieve the best reward by a significant margin compared to peers in their families.

Table 1: Mean reward $\pm$ Standard Deviation of actions generated by different methods after several steps of policy training on the IMDb dataset under noise level 0.4.

Steps	DPO (On clean data)	DPO	cDPO	IPO	SLiC	rDPO
200	0.99 ± 0.03	0.93 ± 0.26	0.84 ± 0.36	0.85 ± 0.35	0.94 ± 0.22	**0.99 ± 0.00**
400	0.99 ± 0.02	0.72 ± 0.43	0.82 ± 0.37	0.83 ± 0.37	0.88 ± 0.31	**0.99 ± 0.00**
600	0.99 ± 0.00	0.88 ± 0.32	0.82 ± 0.38	0.84 ± 0.36	0.90 ± 0.29	**0.99 ± 0.00**
800	0.99 ± 0.00	0.88 ± 0.32	0.83 ± 0.36	0.83 ± 0.37	0.89 ± 0.30	**0.99 ± 0.00**
1000	0.99 ± 0.02	0.88 ± 0.32	0.83 ± 0.37	0.82 ± 0.38	0.90 ± 0.29	**0.99 ± 0.00**

Table 2: Mean reward $\pm$ Standard Deviation on IMDb dataset after policy optimization. The reward model is trained on 1000 steps for all baselines, followed by running PPO for 1 epoch.

Step	PPO (On clean data)	PPO	cPPO	rPPO
1000	0.99 ± 0.00	0.78 ± 0.41	0.87 ± 0.33	**0.94 ± 0.23**

Single-turn Dialogue. In this experiment, each prompt s is a human query and each action a is a helpful response to s. We use the Anthropic helpful and harmless dataset Bai et al. (2022a) as the preference data. In this experiment, as we do not have access to any latent reward model, we employ meta-llama/Llama-2-13b-chat-hf* to compute the win rate of policy generations against the chosen preferences on a representative subset of the test dataset. Next, to demonstrate that of our method generalizes to bigger models, we repeat this experiment with Llama-2-7b as the policy model and GPT-4 as the evaluation model. The win-rates for both experiments are tabulated in Table 2. In both cases, we observe that rDPO performs significantly better than DPO and cDPO.

It remains open to see how our method performs compared to other heuristics proposed in Wang et al. (2024) e.g. flipping some labels or adding an adaptive margin in the loss.

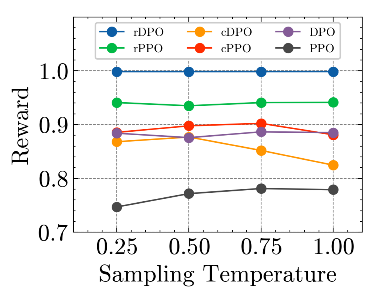

Figure 1: Mean reward on IMDb dataset ($\varepsilon = 0.4$) at different sampling temperatures. For DPO family, we train policy for 1000 steps. For PPO family, we train reward for 1000 steps and then optimize policy for 1 epoch.

Method	Improvement over SFT (%) gpt2-large	Llama-2-7b
DPO	22.20	45.78
cDPO ($\varepsilon = 0.1$)	18.34	39.16
rDPO ($\varepsilon = 0.1$)	**24.32**	**51.20**

Figure 2: Percentage Improvement on win-rate vs chosen response over the initial SFT policy

*huggingface.co/meta-llama/Llama-2-13b-chat-hf

REFERENCES

Alekh Agarwal, Sham M Kakade, Jason D Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *The Journal of Machine Learning Research*, 22(1):4431–4506, 2021.

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

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022a. URL `https://arxiv.org/pdf/2204.05862.pdf`.

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

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

Xiaoyu Chen, Han Zhong, Zhuoran Yang, Zhaoran Wang, and Liwei Wang. Human-in-the-loop: Provably efficient preference-based reinforcement learning with general function approximation. In *International Conference on Machine Learning*, pp. 3773–3793. PMLR, 2022.

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

Daniel Hsu, Sham Kakade, and Tong Zhang. A tail inequality for quadratic forms of subgaussian random vectors. *Electronic Communications in Probability*, 17(none):1 – 6, 2012. doi: 10.1214/ECP.v17-2079. URL `https://doi.org/10.1214/ECP.v17-2079`.

Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback. *arXiv preprint arXiv:2312.14925*, 2023.

Nathan Lambert, Thomas Krendl Gilbert, and Tom Zick. The history and risks of reinforcement learning and human feedback. *arXiv e-prints*, pp. arXiv–2310, 2023.

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

R Duncan Luce. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2012.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In Dekang Lin, Yuji Matsumoto, and Rada Mihalcea (eds.), *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL `https://aclanthology.org/P11-1015`.

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

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

Nagarajan Natarajan, Inderjit S Dhillon, Pradeep K Ravikumar, and Ambuj Tewari. Learning with noisy labels. *Advances in neural information processing systems*, 26, 2013.

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

Aldo Pacchiano, Aadirupa Saha, and Jonathan Lee. Dueling rl: reinforcement learning with trajectory preferences. *arXiv preprint arXiv:2111.04850*, 2021.

Giorgio Patrini, Alessandro Rozza, Aditya Krishna Menon, Richard Nock, and Lizhen Qu. Making deep neural networks robust to label noise: A loss correction approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1944–1952, 2017.

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

Dan Qiao and Yu-Xiang Wang. Offline reinforcement learning with differential privacy. *arXiv preprint arXiv:2206.00810*, 2022.

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

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

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

Louis L Thurstone. A law of comparative judgment. *Psychological review*, 34(4):273, 1927.

Joel A Tropp et al. An introduction to matrix concentration inequalities. *Foundations and Trends® in Machine Learning*, 8(1-2):1–230, 2015.

Kristi Tsukida, Maya R Gupta, et al. How to analyze paired comparison data. *Department of Electrical Engineering University of Washington, Tech. Rep. UWEETR-2011-0004*, 1, 2011.

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

Binghai Wang, Rui Zheng, Lu Chen, Yan Liu, Shihan Dou, Caishuang Huang, Wei Shen, Senjie Jin, Enyu Zhou, Chenyu Shi, et al. Secrets of rlhf in large language models part ii: Reward modeling. *arXiv preprint arXiv:2401.06080*, 2024.

Ruosong Wang, Dean P Foster, and Sham M Kakade. What are the statistical limits of offline rl with linear function approximation? *arXiv preprint arXiv:2010.11895*, 2020.

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

Wenhao Zhan, Masatoshi Uehara, Nathan Kallus, Jason D Lee, and Wen Sun. Provable offline reinforcement learning with human feedback. *arXiv preprint arXiv:2305.14816*, 2023.

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

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

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

A Related Work

Recognizing the storage and computational challenges in RLHF, several alternatives have been proposed. Each of these method work with different loss functions. While DPO optimizes BCE loss to learn the policy Rafailov et al. (2023), SLiC uses hinge loss plus a regularization loss Zhao et al. (2023), IPO uses square-loss Azar et al. (2023), RRHF uses ranking loss plus SFT loss Yuan et al. (2023) and RSO uses BCE loss plus a rejection sampling Liu et al. (2023). While they have their own intricacies and differences, all are competitive with RLHF on standard language tasks.

A recent line of work provides theoretical guarantees on the performance of policy learned using preference-based RL algorithms Pacchiano et al. (2021); Chen et al. (2022); Zhu et al. (2023); Zhan et al. (2023). All these works focus on guarantees in terms of regret bounds in the standard bandit or RL setting and they do not deal with the practical algorithms like RLHF or DPO. Zhu et al. (2024) considers the problem of reward overfitting in RLHF by replacing hard labels with soft ones. They do not consider model overfitting in the presence of noisy data.

There is a line of work in supervised (deep) learning literature that considers learning in the presence of label noise. Müller et al. (2019) study the effect of label smoothing to mitigate the overfitting problem under noisy data. Natarajan et al. (2013) consider binary classification with noisy labels, while Patrini et al. (2017) work on multi-label classification problems. They focus on bounding the excess population risk of trained classifiers under the clean distribution. In contrast, we aim to bound the estimation error of the trained policy, which brings out additional challenges in analysis.

B Details on Experiments

Controlled Sentiment Generation. We extract the first 20 tokens from each review in the IMDb dataset as a prefix. Subsequently, we generate reviews using a gpt2-large model supervised fine-tuned on the IMDb dataset. We generate four reviews resulting in six preference pairs for each prefix. We employ siebert/sentiment-roberta-large-english[†] as the latent (ground-truth) reward model $r^*(s,a)$. To ensure that we have a clean dataset, we only retain preference triplets (s, a_w, a_l) where $r^*(s,a_w) - r^*(s,a_l) > \tau$ where $\tau = 0.1$ is a threshold chosen for this task. This resulted in a dataset with 12000 preference triplets of which 10000 were used to train the policy, and 2000 for evaluation.

We then introduce noise into this dataset by randomly flipping preferences with a probability of $\varepsilon = 0.4$. For all methods, gpt2-large is employed as the initial policy. For methods in the DPO family (vanilla DPO, rDPO, cDPO), we optimized the policy for 1000 steps with batch size 16. We do the same for IPO and SLiC. For methods in the PPO family (vanilla PPO, rPPO, cPPO), we trained a reward model on preference data for 1000 steps with batch size 16 and performed policy optimization for 1 epoch over the entire train dataset.

For evaluation, we generate reviews using the final policy and computed rewards using the ground-truth reward model r^*.

Single-turn Dialogue. We first perform policy optimization using rDPO. As the true noise level in the dataset is unknown, we experiment with different values of $\varepsilon \in \{0.1, 0.2, 0.3, 0.4\}$. We plot the evaluation accuracy of the policy on a subset of the test set across different training steps. This is given by $\frac{1}{m}\sum_{i\in\mathcal{D}_{\text{test}}} \mathbb{1}(\widehat{r}_\theta(s_i, a_{w,i}) > \widehat{r}_\theta(s_i, a_{l,i}))$, where $\widehat{r}_\theta$ is the implicit reward defined by policy π_θ. We observed the best results with $\varepsilon = 0.1$. Subsequently, we train DPO and cDPO (with label-smoothing $\varepsilon = 0.1$) on the same data.

B.1 Hyperparameter Details

The hyperparameters for the experiments are outlined in Table 3 and Table 4. Any hyperparameters not explicitly mentioned use the default values in the TRL[‡] library.

[†]huggingface.co/siebert/sentiment-roberta-large-english

[‡]huggingface.co/docs/trl/index

Table 3: Hyperparameters used for methods in the DPO Family

Parameter	Value
beta	0.1
learning rate	0.001
batch size	16
max length	512
max prompt length	128

Table 4: Hyperparameters used for methods in the PPO Family

Model	Parameter	Value
Reward Model	learning rate	1.41 x 10^{-5}
	batch size	16
PPO	learning rate	1.41 x 10^{-5}
	batch size	16

C DETAILS ON THEORETICAL RESULTS

Our method enjoys certain theoretical properties. By unbiasedness of $\widehat{\mathcal{L}}_\varepsilon$, we know that, for any fixed $\theta \in \Theta$, the empirical rDPO loss (4) converges to the population DPO loss $\mathbb{E}_{s,a_w,a_l}\Big[\mathcal{L}(\theta; s, a_w, a_l)\Big]$ even though the former is computed using noisy preferences whereas the latter depends on clean preferences. But the rDPO policy $\widehat{\pi}_n = \pi_{\widehat{\theta}_n}$ won't necessarily converge to the optimal policy π^* as preference pairs are sampled from the SFT policy π_{sft}, but not form π^* - an issue also shared by DPO policy Liu et al. (2023). However, our end goal is to bound the sub-optimality gap of $\widehat{\pi}_n$. For this, we only need to characterize the estimation error of the learned policy parameter $\widehat{\theta}_n$ as function of number of samples n and flip rate ε.

C.1 ESTIMATION ERROR

Under the BTL model, two reward functions from the same equivalence class[§] induce the same preference distribution and the same optimal policy Rafailov et al. (2023). Due to this model under-specification and reward re-parameterization (1), we need to impose an identifiability constraint on the set of policy parameters Θ, namely $\Theta = \{\theta \in \mathbb{R}^d | \sum_{i=1}^d \theta_i = 0\}$ to achieve any guarantee on the estimation error. We also assume $\|\theta\| \leq B$, $\forall \theta \in \Theta$. We give guarantees for Neural policy class of the form (2), i.e., when f_θ is a neural network parameterized by θ. We make a smoothness assumption on the policy class:

Assumption C.1 (Smoothness)**.** For any $\theta \in \Theta$ and (s, a),

$$|f_\theta(s,a)| \leq \alpha_0, \|\nabla f_\theta(s,a)\| \leq \alpha_1, \nabla^2 f_\theta(s,a) \preccurlyeq \alpha_2 I\ .$$

The assumption ensures that implicit reward differences $h_\theta(s, a_w, a_l)$ are bounded, Lipschitz, and their gradients are also Lipschitz. This is quite common for establishing convergence for policy gradient methods Agarwal et al. (2021). Log-linear policies ($f_\theta(s,a) = \theta^\top \phi(s,a)$), satisfy this assumption with $\alpha_0 = LB, \alpha_1 = L, \alpha_2 = 0$, where L is an upper bound on ℓ_2-norm of features $\phi(s,a)$.

The following result gives a guarantee on the estimation error in terms of the parameter dimension and flip rate. The main idea in the proof is to use strong convexity of rDPO loss $\widehat{\mathcal{L}}_\varepsilon$ in the semi-norm $\|\cdot\|_{\widehat{\Sigma}_\theta}$. Here, for any $\theta \in \mathbb{R}^d$, $\widehat{\Sigma}_\theta = \frac{1}{n}\sum_{i=1}^n x_i x_i^\top$ is the sample covariance matrix of gradients of implicit reward differences under true preferences, where $x_i = \nabla h_\theta(s_i, a_{w,i}, a_{l,i}) = \nabla f_\theta(s_i, a_{w,i}) - \nabla f_\theta(s_i, a_{l,i})$.

[§]Two reward functions r_1, r_2 are equivalent iff $r_1(s,a) - r_2(s,a) = g(s)$ for some function g.

The error scales inversely with $\gamma\beta(1-2\varepsilon)$, where $\gamma \leq \sigma'(\beta h_\theta(s, a_w, a_l))$ for all $\theta \in \Theta$ and for all preference samples (s, a_w, a_l). Here γ lower bounds the first derivative of the logistic function $\sigma(z_\theta; \beta, z_0) = \frac{1}{1+e^{-\beta(z_\theta - z_0)}}$, where $z_\theta = f_\theta(s, a_w) - f_\theta(s, a_l)$ and $z_0 = z_{\theta_0}$.

Theorem C.2 (Estimation error of $\widehat{\theta}_n$)**.** *Let $\delta \in (0,1], \varepsilon \in [0, 1/2), \lambda > 0$. Then, for Neural policy class* (2) *and under Assumption C.1, with probability at least $1-\delta$, we have*

$$\left\|\widehat{\theta}_n - \theta^*\right\|_{\widehat{\Sigma}_{\theta^*}+\lambda I} \leq \frac{C}{\gamma\beta(1-2\varepsilon)} \cdot \sqrt{\frac{d+\log(1/\delta)}{n}} + C' \cdot B\sqrt{\lambda + \frac{\alpha_2}{\gamma\beta(1-2\varepsilon)} + \alpha_1\alpha_2 B}\ ,$$

where $\gamma = \frac{1}{2+e^{-4\beta\alpha_0}+e^{4\beta\alpha_0}}$, C, C' are absolute constants.

Several remarks are in order with this result. To keep the presentation simple, we consider log-linear policies in the following.

In this case $\alpha_2 = 0$ and $x_i = \phi(s_i, a_{w,i}) - \phi(s_i, a_{l,i})$. In this case, $\widehat{\Sigma}_\theta$ is the covariance matrix of feature differences and independent of θ. We denote this by $\widehat{\Sigma}$ and get a high-probability error bound for log-linear policy class (see Theorem 3.2 in the main paper):

$$\left\|\widehat{\theta}_n - \theta^*\right\|_{\widehat{\Sigma}+\lambda I} = O\Big(\frac{1}{\gamma\beta(1-2\varepsilon)}\sqrt{\frac{d}{n}} + B\sqrt{\lambda}\Big)\ . \tag{9}$$

Choice of Regularizer λ. When the feature covariance matrix $\widehat{\Sigma}$ is invertible, the above result holds for $\lambda = 0$. In this case, we will get a vanishing error-rate in the ℓ_2-norm

$$\left\|\widehat{\theta}_n - \theta^*\right\| = O\Big(\frac{1}{\sqrt{\lambda_{\min}(\Sigma)}}\frac{1}{\gamma\beta(1-2\varepsilon)}\sqrt{\frac{d}{n}}\Big)\ . \tag{10}$$

If this is not the case, $\widehat{\theta}_n$ won't necessarily converge to θ^*. But one might set $\lambda = O(d/n)$ to achieve a vanishing error in the semi-norm $\widehat{\Sigma}$ for log-linear policies. However, the error will not vanish for Neural policies (as $\alpha_2 \neq 0$).

Estimation Error of DPO Policy. As already mentioned, our rDPO policy (5) recovers the DPO policy under clean preferences. Thus, setting $\varepsilon = 0$ in Theorem C.2, we get an error bound of order $O\big(\frac{1}{\gamma}\sqrt{d/n}\big)$ for the DPO policy. Therefore, as a by-product of our approach, we get the first error bound for the trained DPO policy of Rafailov et al. (2023), which could be of independent interest.

Effect of Noisy Preferences. When preferences are noisy (i.e. flip rate $\varepsilon > 0$), our rDPO policy achieves an error bound of order $O\big(\frac{1}{\gamma(1-2\varepsilon)}\sqrt{d/n}\big)$. Comparing this with the above error bound for DPO policy under clean preferences, we see that the cost of preference flips is a multiplicative factor of the order $\frac{1}{1-2\varepsilon}$ – the higher the (expected) number of preference flips, the higher the estimation error.

Effect of KL regularizer. Since $\gamma = O(1/e^\beta)$, the dependence of estimation error on the KL regularizer β is of the order $g(\beta) = O(e^\beta/\beta)$. Hence our result won't no longer hold true when $\beta = 0$ (no regularization). In this case preference probabilities are exactly equal to $1/2$ (both actions are equally preferred), making learning impossible. Same is the case when $\beta \to \infty$ (full regularization) since one action will always be preferred over the other with probability 1, making the loss function degenerate. This points out the need for tuning β properly.

C.2 Performance Bounds of Learned Policy

In this Section, we discuss how the estimation error of $\widehat{\theta}_n$ relates to the sub-optimality gap of the policy $\widehat{\pi}_n$. We will consider log-linear policy class for ease of presentation.

It is well known that learning a near-optimal policy from an offline batch of data cannot be sample efficient without assuming the behavior policy (SFT in our case) has a good coverage over the feature space Wang et al. (2020). To begin with, we define the population covariance matrix of centered features under a policy π:

$$\Sigma_\pi = \mathbb{E}\big[\phi(s,a)\phi(s,a)^\top\big] - \mathbb{E}[\phi(s,a)]\mathbb{E}[\phi(s,a)]^\top\ , \tag{11}$$

where the expectation is over random draws from $s \sim \rho, a \sim \pi(\cdot|s)$. Now, we define the condition number of Σ_π relative to $\Sigma_{\pi_{\text{sft}}}$ (covariance matrix under SFT policy):

$$\forall \pi \in \Pi : \quad \kappa_\pi = \sup_{v \in \mathbb{R}^d} \frac{v^\top \Sigma_\pi v}{v^\top \Sigma_{\pi_{\text{sft}}} v} = \frac{\lambda_{\max}(\Sigma_\pi)}{\lambda_{\min}(\Sigma_{\pi_{\text{sft}}})} \ .$$

A small relative condition number helps to keep the ratio of maximum feature coverage of policy to be evaluated and minimum coverage of starting policy in check. Thus, it is important to have a good starting policy π_{sft} to ensure a small condition number. Roughly speaking, we desire an SFT policy which provides good coverage over the features.

Assumption C.3 (Feature coverage)**.** The SFT policy satisfies the minimum eigenvalue condition: $\lambda_{\min}(\Sigma_{\pi_{\text{sft}}}) > 0$.

Let $\kappa = \max_{\pi \in \Pi} \kappa_\pi$. The assumption ensures $\kappa < \infty$. The result below shows how estimation error and condition number determine the final performance of our learned policy.

Theorem C.4 (Sub-optimality gap of $\widehat{\pi}_n$)**.** *Let $\delta \in (0, 1]$ and $r^*(s, a) \leq r_{\max}$ for all (s, a). Then, for log-linear policy class, we have with probability at least $1 - \delta$:*

$$r^*(\pi^*) - r^*(\widehat{\pi}_n) \leq r_{\max} \sqrt{\kappa/2} \left\| \widehat{\theta}_n - \theta^* \right\|_{\widehat{\Sigma}+\lambda I}$$

for $\lambda \geq C \sqrt{d \log(4d/\delta)/n}$, where C is a universal constant.

Dimension dependence in κ. It is reasonable to expect κ to be dimension dependent, but it doesn't necessarily depend on the size of the vocabulary. To see this, consider log-linear policies with bounded features $\|\phi(s, a)\| \leq L$. In this case $\lambda_{\max}(\Sigma_\pi) \leq L^2$ and thus $\kappa_\pi \leq \frac{L^2}{\lambda_{\min}(\Sigma_{\pi_{\text{sft}}})}$. Now, $\lambda_{\min}(\Sigma_\pi)$ depends implicitly on the dimension d of features $\phi(s, a)$ and it is reasonable to assume $\lambda_{\min}(\Sigma_{\pi_{\text{sft}}}) = \Theta(L^2/d)$ (Wang et al., 2020). Thus it is always possible to have $\kappa = O(d)$ Agarwal et al. (2021).

Now, plugging in the bound on estimation error (9) in Theorem C.4, we get a sub-optimality gap of order $O\Big(\frac{\sqrt{\kappa}}{\gamma\beta(1-2\varepsilon)} \sqrt{\frac{d}{n}} + \frac{\sqrt{\kappa} d^{1/4}}{n^{1/4}}\Big)$. However, when sample feature covariance matrix $\widehat{\Sigma}$ is invertible, i.e. observed samples from SFT policy provide good coverage of the feature space, then we get $O\Big(\frac{\sqrt{\kappa}}{\gamma\beta(1-2\varepsilon)} \sqrt{\frac{d}{n}}\Big)$ suboptimality gap (Lemma 3.3 in the main paper).

Margin Gap. A related performance measure is the *margin* under clean distribution. The margin of a policy π_θ is defined to be the average difference of implicit rewards $\widehat{r}_\theta(s, a) = \log \frac{\pi_\theta(a|s)}{\pi_{\text{sft}}(a|s)}$ of chosen and rejected actions, i.e.,

$$\mathcal{M}(\pi_\theta) = \mathbb{E}_{s \sim \rho, (y_w, y_l) \sim \pi_{\text{sft}}} \left[\widehat{r}_\theta(a_w|s) - \widehat{r}_\theta(a_l|s) \right] \ .$$

Then $\mathcal{M}(\pi^*) - \mathcal{M}(\widehat{\pi}_n)$ defines the margin gap of learned policy $\widehat{\pi}_n$ from the optimal policy π^*. This metric is quite commonly used by practitioners to demonstrate performance of learned policy von Werra et al. (2020).

Lemma C.5 (Margin gap)**.** *Assuming $\widehat{\Sigma}$ to be invertible for log-linear policy class, the margin gap of $\widehat{\pi}_n$ satisfies*

$$\mathcal{M}(\pi^*) - \mathcal{M}(\widehat{\pi}_n) = O\Big(\frac{1}{\lambda_{\min}(\widehat{\Sigma}^{1/2})} \frac{1}{\gamma\beta(1 - 2\varepsilon)} \sqrt{\frac{d}{n}} \Big) \ .$$

Since $\kappa = O(1/\lambda_{\min}(\Sigma_{\pi_{\text{sft}}}))$, comparing this result with sub-optimality bound from the above paragraph, we see that both margin and sub-optimality gaps are roughly of the same order when $\widehat{\Sigma}$ has good coverage. This is also reflected in our experiments, where we see strong correlation between evaluation accuracy (on clean data) and average reward performance for any policy; see Section B.

Generalizing to Neural Policy Class. A similar reasoning as the above can be also used to establish a sub-optimality bound for neural policy class (2). Here the relative condition number needs to be defined using the covariance matrix for the features $f_\theta(s, a)$, which depend on θ, as opposed to the feature map $\phi(s, a)$ in the log-linear case. The rest follows with an appropriate adaptation of the results above.

D Generalizations and Extensions

Our approach to mitigate the effect of noisy preferences in data is not limited to DPO algorithm and BTL preference model. It is a general framework that can be adapted to other preference optimizations methods (e.g. SLiC, IPO) and other preference models (e.g. probit, Placket-Luce). More importantly, since DPO implicitly learns a reward function $\widehat{r}_\theta$ as we have discussed above, our method seamlessly extends to the reward training stage of the RLHF pipeline, showing versatility of our proposed approach.

Reward training in RLHF. Let us consider parameterized reward models $r_\xi(s,a)$, where $\xi \in \mathbb{R}^d$ is a parameter vector. Let ξ^* be the parameter of the latent reward model $r^*(s,a)$. Then, the true preference probabilities following BTL model are given by

$$p^*_{s,a,a'} = \mathbb{P}_{\xi^*}[a \succ a'|s] = \sigma(r_{\xi^*}(s,a) - r_{\xi^*}(s,a')) \ .$$

Similar to (3), for any $\xi \in \mathbb{R}^d$, this yields the BCE loss for a preference pair (s, a_w, a_l):

$$\mathcal{L}(\xi; s, a_w, a_l) = -\log \sigma(r_\xi(s,a_w) - r_\xi(s,a_l)) \ . \tag{12}$$

Under our random noise model with flip rate ε, for a potentially noisy data $(s, \widetilde{a}_w, \widetilde{a}_l)$, one can define a loss $\widehat{\mathcal{L}}_\varepsilon(\xi; s, \widetilde{a}_w, \widetilde{a}_l)$ using (4), which will be an unbiased estimate of (12). Thus, using a similar argument as in Section 3, a reward model trained by minimizing this loss will be robust to noisy preferences. This trained reward model can be then directly plugged into (1) to train a language model policy. In practice (1) is solved using PPO algorithm Schulman et al. (2017). Thus, we call this entire procedure robust PPO (or rPPO in short).

Other Optimization Methods. Instead of the BCE loss (3), SLiC Zhao et al. (2023) minimizes a hinge loss:

$$\mathcal{L}_{\text{hinge}}(\theta; s, a_w, a_l) = \max\{0, 1 - \beta h_\theta(s, a_w, a_l)\}$$

where $1/\beta$ acts as the margin (of miss-classification). IPO Azar et al. (2023) minimizes square loss:

$$\mathcal{L}_{\text{IPO}}(\theta; s, a_w, a_l) = (\beta h_\theta(s, a_w, a_l) - 1/2)^2 \ .$$

A potential advantage of IPO and SLiC over DPO is that these methods don't assume any preference model like BTL and could work with general preference probabilities. Under our random noise model, one can define robust counterparts of both $\mathcal{L}_{\text{hinge}}$ and $\mathcal{L}_{\text{IPO}}$ using (4). This will ensure these losses under noisy data $(\widetilde{a}_w, \widetilde{a}_l)$ are unbiased estimates of those under clean data (a_w, a_l), and will help one learn a robust policy for these loss functions.

Other Preference Models. Our results can be extended to any preference model of the form $p^*_{s,a,a'} = \mathbb{P}[a \succ a'|s] = g(r^*(s,a) - r^*(s,a'))$, if g is strongly log-concave, i.e., $-\frac{d^2}{dz^2}\log g(z) \geq \gamma > 0$ in a closed interval around $z = 0$. For example, in the probit (also known as Thurstone) model (Thurstone, 1927), g is the CDF of standard Gaussian distribution. Thus, for any θ, the preference probabilities are $\mathbb{P}_\theta[a \succ a'|s] = \Phi(\beta h_\theta(s,a,a'))$. Since Φ is strongly log-concave in Θ (Tsukida et al., 2011), one can derive similar performance bounds under probit model too.

For the Placket-Luce (PL) model (Plackett, 1975; Luce, 2012) for K-wise comparisons between actions. Let Π be the set of all permutations $\pi : [K] \to [K]$, that denotes a ranking given by an oracle over all K actions, where $a_{\pi(j)}$ denotes the j-th ranked action. Under the PL model, we define the loss of a permutation $\pi \in \Pi$ for a question s as

$$\mathcal{L}(\theta; s, \pi) = -\log\Big(\prod_{j=1}^{K} \frac{\exp(\widehat{r}_\theta(s, a_{\pi(j)}))}{\sum_{k'=j}^{K}\exp(\widehat{r}_\theta(s, a_{\pi(k')}))}\Big) \ .$$

Noisy preferences are obtained by perturbing the true ranking π to some other ranking $\widetilde{\pi}$ with probability $\frac{\varepsilon}{N-1}$, where N is the number of possible rankings (can be at most $K!$). Then, if we define the robust-loss for noisy ranking $\widetilde{\pi}$ as

$$\widehat{\mathcal{L}}_\varepsilon(\theta; s, \widetilde{\pi}) = \frac{\big(N-1-\varepsilon\big)\mathcal{L}(\theta; s, \widetilde{\pi}) - \varepsilon\sum_{\pi' \neq \widetilde{\pi}}\mathcal{L}(\theta; s, \pi')}{(1-\varepsilon)N - 1},$$

it will be an unbiased estimate of $\mathcal{L}(\theta; s, \pi)$. This would help us to learn a robust policy under PL feedback model.

E Proofs

E.1 Proof of Unbiasedness of rDPO Loss

It is easy to see that

$$\mathbb{E}_\varepsilon\left[\widehat{\mathcal{L}}_\varepsilon(\theta; s, \widetilde{a}_w, \widetilde{a}_l)|a_w, a_l\right]$$
$$= \frac{(1-\varepsilon)^2\mathcal{L}(\theta; s, a_w, a_l) - \varepsilon(1-\varepsilon)\mathcal{L}(\theta; s, a_l, a_w)}{1-2\varepsilon} + \frac{\varepsilon(1-\varepsilon)\mathcal{L}(\theta; s, a_l, a_w) - \varepsilon^2\mathcal{L}(\theta; s, a_w, a_l)}{1-2\varepsilon}$$
$$= \mathcal{L}(\theta; s, a_w, a_l)\ .$$

E.2 Proof of Lemma 3.1

The gradients of the rDPO loss $\widehat{\mathcal{L}}_\varepsilon$ with respect to the parameters θ can be written as

$$\begin{aligned}\nabla_\theta \widehat{\mathcal{L}}_\varepsilon(\theta; s, \widetilde{a}_w, \widetilde{a}_l) &= \frac{(1-\varepsilon)\nabla_\theta\mathcal{L}(\theta; s, \widetilde{a}_w, \widetilde{a}_l) - \varepsilon\nabla_\theta\mathcal{L}(\theta; s, \widetilde{a}_l, \widetilde{a}_w)}{1-2\varepsilon} \\ &= -\beta \cdot \widehat{\zeta}_{\theta,\varepsilon} \cdot \big(\nabla_\theta \log \pi_\theta(\widetilde{a}_w|s) - \nabla_\theta \log \pi_\theta(\widetilde{a}_l|s)\big)\ ,\end{aligned}$$

where the weights $\widehat{\zeta}_{\theta,\varepsilon}$ are given by

$$\begin{aligned}\zeta_{\theta,\varepsilon} &= \frac{1-\varepsilon}{1-2\varepsilon}\sigma(\beta h_\theta(s, \widetilde{a}_l, \widetilde{a}_w)) + \frac{\varepsilon}{1-2\varepsilon}\sigma(\beta h_\theta(s, \widetilde{a}_w, \widetilde{a}_l)) \\ &= \frac{1-\varepsilon}{1-2\varepsilon} - \sigma(\beta h_\theta(s, \widetilde{a}_w, \widetilde{a}_l)) = \frac{\varepsilon}{1-2\varepsilon} + \sigma(\beta h_\theta(s, \widetilde{a}_l, \widetilde{a}_w)) = \zeta_\theta + \frac{\varepsilon}{1-2\varepsilon}.\end{aligned}$$

The gradient of cDPO loss is given by

$$\begin{aligned}\nabla_\theta \bar{\mathcal{L}}_\varepsilon(\theta; s, \widetilde{a}_w, \widetilde{a}_l) &= (1-\varepsilon)\nabla_\theta\mathcal{L}(\theta; s, \widetilde{a}_w, \widetilde{a}_l) + \varepsilon\nabla_\theta\mathcal{L}(\theta; s, \widetilde{a}_l, \widetilde{a}_w) \\ &= -\beta \cdot \bar{\zeta}_{\theta,\varepsilon} \cdot \big(\nabla_\theta \log \pi_\theta(\widetilde{a}_w|s) - \nabla_\theta \log \pi_\theta(\widetilde{a}_l|s)\big)\ ,\end{aligned}$$

where the weights are $\bar{\zeta}_{\theta,\varepsilon} = (1-\varepsilon)\sigma(\beta h_\theta(s, \widetilde{a}_l, \widetilde{a}_w)) - \varepsilon\sigma(\beta h_\theta(s, \widetilde{a}_w, \widetilde{a}_l))$. It holds that

$$\bar{\zeta}_{\theta,\varepsilon} = \sigma(\beta h_\theta(s, \widetilde{a}_l, \widetilde{a}_w)) - \varepsilon = \zeta_\theta - \varepsilon = \widehat{\zeta}_{\theta,\varepsilon} - \frac{2\varepsilon(1-\varepsilon)}{1-2\varepsilon}\ .$$

E.3 Proof of Theorem C.2

For the neural policy of the form (2), we have

$$h_\theta(s, a, a') = [f_\theta(s, a) - f_\theta(s, a')] - [f_{\theta_0}(s, a) - f_{\theta_0}(s, a')]\ .$$

Then from Assumption C.1, we have

$$\begin{aligned}|h_\theta(s, a, a')| &\le |f_\theta(s, a) - f_{\theta_0}(s, a)| + |f_\theta(s, a') - f_{\theta_0}(s, a')| \le 2\alpha_0, \\ \|\nabla h_\theta(s, a, a')\| &= \|\nabla f_\theta(s, a) - \nabla f_\theta(s, a')\| \le 2\alpha_1\ , \\ \left\|\nabla^2 h_\theta(s, a, a')\right\| &= \left\|\nabla^2 f_\theta(s, a) - \nabla^2 f_\theta(s, a')\right\| \le 2\alpha_2\ .\end{aligned}$$

Now, we express the population DPO loss $\mathbb{E}_{s,a_w,a_l}\left[\mathcal{L}(\theta; s, a_w, a_l)\right]$ by incorporating preference probabilities $p^*_{s,a,a'}$ as

$$\mathcal{L}(\theta) = -\mathbb{E}_{s,a,a',y}\Big[-y\log\sigma(\beta h_\theta(s, a, a')) + (1-y)\log(1-\sigma(\beta h_\theta(s, a, a'))\Big]\ ,$$

where y is a Bernoulli random variable with mean $p^*_{s,a,a'} = \sigma(\beta h_{\theta^*}(s, a, a')$.

Similarly, under the random noise model, let each $\widetilde{y}_i$ be Bernoulli distributed with probability $\mathbb{P}_{\theta^*,\varepsilon}[\widetilde{a}_{w,i} \succ \widetilde{a}_{l,i}|s_i]$, where $\mathbb{P}_{\theta,\varepsilon}[a \succ a'|s]$ is given by

$$\begin{aligned}\mathbb{P}_{\theta,\varepsilon}[a \succ a'|s] &= (1-\varepsilon)\cdot\mathbb{P}_\theta[a \succ a'|s] + \varepsilon\cdot\mathbb{P}_\theta[a' \succ a|s] \\ &= (1-\varepsilon)\cdot\sigma(\beta h_\theta(s, a, a')) + \varepsilon\cdot\sigma(\beta h_\theta(s, a', a))\ .\end{aligned} \tag{13}$$

Denote $z_i = (s_i, \widetilde{a}_{w,i}, \widetilde{a}_{l,i})$. Then, our de-biased loss function (4) can be re-written as

$$\widehat{\mathcal{L}}_\varepsilon(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\Bigg[\mathbb{1}(\widetilde{y}_i = 1)\Big((1-\varepsilon)\log\sigma(\beta h_\theta(z_i)) - \varepsilon\log(1-\sigma(\beta h_\theta(z_i))\Big) + \mathbb{1}(\widetilde{y}_i = 0)\Big((1-\varepsilon)\log(1-\sigma(\beta h_\theta(z_i)) - \varepsilon\log\sigma(\beta h_\theta(z_i))\Big)\Bigg] .$$

The gradient of the loss function is given by $\nabla\widehat{\mathcal{L}}_\varepsilon(\theta) = -\frac{\beta}{n}\sum_{i=1}^{n} V_{\theta,i}\nabla h_\theta(z_i) = -\frac{\beta}{n}Z_\theta^\top V_\theta$, where

$$V_{\theta,i} = \mathbb{1}(\widetilde{y}_i = 1)\left(\frac{\sigma'(\beta h_\theta(z_i))}{\sigma(\beta h_\theta(z_i))}(1-\varepsilon) + \frac{\sigma'(\beta h_\theta(z_i))}{1-\sigma(\beta h_\theta(z_i))}\varepsilon\right) - \mathbb{1}(\widetilde{y}_i = 0)\left(\frac{\sigma'(\beta h_\theta(z_i))}{1-\sigma(\beta h_\theta(z_i))}(1-\varepsilon) + \frac{\sigma'(\beta h_\theta(z_i))}{\sigma(\beta h_\theta(z_i))}\varepsilon\right) .$$

It holds that for $\theta = \theta^*$:

$$\begin{aligned}\mathbb{E}_\theta[V_{\theta,i}|z_i] &= \Big(\sigma(\beta h_\theta(z_i))(1-\varepsilon) + (1-\sigma(\beta h_\theta(z_i)))\varepsilon\Big)\left(\frac{\sigma'(\beta h_\theta(z_i))}{\sigma(\beta h_\theta(z_i))}(1-\varepsilon) + \frac{\sigma'(\beta h_\theta(z_i))}{1-\sigma(\beta h_\theta(z_i)}\varepsilon\right)\\ &\quad - \Big((1-\sigma(\beta h_\theta(z_i)))(1-\varepsilon) + \sigma(\beta h_\theta(z_i))\varepsilon\Big)\left(\frac{\sigma'(\beta h_\theta(z_i))}{1-\sigma(\beta h_\theta(z_i))}(1-\varepsilon) + \frac{\sigma'(\beta h_\theta(z_i))}{\sigma(\beta h_\theta(z_i))}\varepsilon\right)\\ &= 0 .\end{aligned}$$

Furthermore, we have

$$\begin{aligned}|V_{\theta,i}|_{\widetilde{y}_i=1} &= (1-\sigma(\beta h_\theta(z_i)))(1-\varepsilon) + \sigma(\beta h_\theta(z_i))\varepsilon =: \widetilde{p}_{i,0} \le 1,\\ |V_{\theta,i}|_{\widetilde{y}_i=0} &= \sigma(\beta h_\theta(z_i))(1-\varepsilon) + (1-\sigma(\beta h_\theta(z_i)))\varepsilon =: \widetilde{p}_{i,1} \le 1 .\end{aligned}$$

Therefore, it holds that $V_{\theta^*,i}$ is zero-mean and $v = 1$ sub-Gaussian under the conditional distribution $\mathbb{P}_{\theta^*}[\cdot|z_i]$.

Now the Hessian of the loss function is given by

$$\nabla^2\widehat{\mathcal{L}}_\varepsilon(\theta) = \frac{1}{n}\sum_{i=1}^{n}\Bigg[\mathbb{1}(\widetilde{y}_i = 1)\left(\varepsilon\nabla^2\log(1-\sigma(\beta h_\theta(z_i))) - (1-\varepsilon)\nabla^2\log\sigma(\beta h_\theta(z_i))\right) + \mathbb{1}(\widetilde{y}_i = 0)\left(\varepsilon\nabla^2\log\sigma(\beta h_\theta(z_i)) - (1-\varepsilon)\nabla^2\log(1-\sigma(\beta h_\theta(z_i)))\right)\Bigg] ,$$

where

$$\begin{aligned}\nabla^2\log\sigma(\beta h_\theta(z_i)) &= \beta^2\frac{\sigma''(\beta h_\theta(z_i))\sigma(\beta h_\theta(z_i)) - \sigma'(\beta h_\theta(z_i))^2}{\sigma(\beta h_\theta(z_i))^2}\nabla h_\theta(z_i)\nabla h_\theta(z_i)^\top\\ &\quad + \beta(1-\sigma(\beta h_\theta(z_i)))\nabla^2 h_\theta(z_i),\\ \nabla^2\log(1-\sigma(\beta h_\theta(z_i))) &= -\beta^2\frac{\sigma''(\beta h_\theta(z_i))(1-\sigma(\beta h_\theta(z_i))) + \sigma'(\beta h_\theta(z_i))^2}{(1-\sigma(\beta h_\theta(z_i)))^2}\nabla h_\theta(z_i)\nabla h_\theta(z_i)^\top\\ &\quad - \beta\sigma(\beta h_\theta(z_i))\nabla^2 h_\theta(z_i).\end{aligned}$$

Using $\sigma''(z) = \sigma'(z)(1-2\sigma(z))$, we get

$$\begin{aligned}\nabla^2\log\sigma(\beta h_\theta(z_i)) &= -\beta^2\sigma'(\beta h_\theta(z_i))\nabla h_\theta(z_i)\nabla h_\theta(z_i)^\top + \beta(1-\sigma(\beta h_\theta(z_i))))\nabla^2 h_\theta(z_i)\\ \nabla^2\log(1-\sigma(\beta h_\theta(z_i))) &= -\beta^2\sigma'(\beta h_\theta(z_i))\nabla h_\theta(z_i)\nabla h_\theta(z_i)^\top - \beta\sigma(\beta h_\theta(z_i))\nabla^2 h_\theta(z_i) .\end{aligned}$$

Hence, the Hessian of the loss function takes the form

$$\begin{aligned}
\nabla^2 \widehat{\mathcal{L}}_\varepsilon(\theta) &= (1-2\varepsilon)\beta^2 \frac{1}{n}\sum_{i=1}^n \sigma'(\beta h_\theta(z_i))\nabla h_\theta(z_i)\nabla h_\theta(z_i)^\top \\
&\quad - \frac{\beta}{n}\sum_{i=1}^n \mathbb{1}(\widetilde{y}_i = 1)\Big(\sigma(\beta h_\theta(z_i))\varepsilon + (1-\sigma(\beta h_\theta(z_i)))(1-\varepsilon)\Big)\nabla^2 h_\theta(z_i) \\
&\quad + \frac{\beta}{n}\sum_{i=1}^n \mathbb{1}(\widetilde{y}_i = 0)\Big(\sigma(\beta h_\theta(z_i))(1-\varepsilon) + (1-\sigma(\beta h_\theta(z_i)))\varepsilon\Big)\nabla^2 h_\theta(z_i) \\
&= \beta^2(1-2\varepsilon)\frac{1}{n}\sum_{i=1}^n \sigma'(\beta h_\theta(z_i))\nabla h_\theta(z_i)\nabla h_\theta(z_i)^\top - \frac{\beta}{n}\sum_{i=1}^n \mathbb{1}(\widetilde{y}_i = 1)\widetilde{p}_{i,0}\nabla^2 h_\theta(z_i) \\
&\quad + \frac{\beta}{n}\sum_{i=1}^n \mathbb{1}(\widetilde{y}_i = 0)\widetilde{p}_{i,1}\nabla^2 h_\theta(z_i) \\
&\geqslant \gamma\beta^2(1-2\varepsilon)\frac{1}{n}\sum_{i=1}^n \nabla h_\theta(z_i)\nabla h_\theta(z_i)^\top - 2\beta\alpha_2 I\ ,
\end{aligned}$$

which holds by Assumption C.1 and observing that $\sigma'(\beta h_\theta(z_i)) \geq \gamma$ for all $\theta \in \Theta$, where $\gamma = \frac{1}{2+\exp(-4\beta\alpha_0)+\exp(4\beta\alpha_0)}$, and due to the fact that $\varepsilon < 1/2$.

Then, from Assumption C.1 and using simple algebra, we have for all $u \in \mathbb{R}^d$:

$$u^\top \nabla^2 \widehat{\mathcal{L}}_\varepsilon(\theta) u \geqslant \frac{\gamma\beta^2(1-2\varepsilon)}{n}\left\|Z_{\theta^*}u\right\|^2 - 2\alpha_2(\beta + 2\gamma\beta^2(1-2\varepsilon)\alpha_1 B)\left\|u\right\|^2\ .$$

Since $\theta^* \in \Theta$, introducing the error vector $\Delta = \widehat{\theta}_n - \theta^*$, we conclude that

$$\gamma\beta^2(1-2\varepsilon)\left\|\Delta\right\|^2_{\Sigma_{\theta^*}} \leqslant \left\|\nabla \widehat{\mathcal{L}}_\varepsilon(\theta^*)\right\|_{(\widehat{\Sigma}_{\theta^*}+\lambda I)^{-1}}\left\|\Delta\right\|_{(\widehat{\Sigma}_{\theta^*}+\lambda I)} + 2\alpha_2\beta(1+2\beta\gamma(1-2\varepsilon)\alpha_1 B)\left\|\Delta\right\|^2$$

for some $\lambda > 0$. Introducing $M_{\theta^*} = \frac{1}{n^2}Z_{\theta^*}(\widehat{\Sigma}_{\theta^*} + \lambda I)^{-1}Z_{\theta^*}^\top$, we now have $\left\|\nabla \widehat{\mathcal{L}}_\varepsilon(\theta^*)\right\|^2_{(\widehat{\Sigma}_{\theta^*}+\lambda I)^{-1}} = \beta^2 V_{\theta^*}^\top M_{\theta^*} V_{\theta^*}$. Then, the Bernstein's inequality for sub-Gaussian random variables in quadratic form (see e.g. Hsu et al. (2012, Theorem 2.1)) implies that with probability at least $1-\delta$,

$$\begin{aligned}
\left\|\nabla \widehat{\mathcal{L}}_\varepsilon(\theta^*)\right\|^2_{(\widehat{\Sigma}_{\theta^*}+\lambda I)^{-1}} = \beta^2 V_{\theta^*}^\top M_{\theta^*} V_{\theta^*} &\leqslant \beta^2\left(\mathrm{tr}(M_{\theta^*}) + 2\sqrt{\mathrm{tr}(M_{\theta^*}^\top M_{\theta^*})\log(1/\delta)} + 2\left\|M_{\theta^*}\right\|\log(1/\delta)\right) \\
&\leqslant C_1 \cdot \beta^2 \cdot \frac{d+\log(1/\delta)}{n}
\end{aligned}$$

for some $C_1 > 0$. This gives us

$$\begin{aligned}
&\gamma\beta^2(1-2\varepsilon)\left\|\Delta\right\|^2_{\widehat{\Sigma}_{\theta^*}+\lambda I} \\
&\leqslant \left\|\nabla \widehat{\mathcal{L}}_\varepsilon(\theta^*)\right\|_{(\Sigma_{\theta^*}+\lambda I)^{-1}}\left\|\Delta\right\|_{(\widehat{\Sigma}_{\theta^*}+\lambda I)} + 4(\lambda\gamma\beta^2(1-2\varepsilon) + 2\alpha_2\beta(1+2\beta\gamma(1-2\varepsilon)\alpha_1 B))B^2 \\
&\leqslant \sqrt{C_1\cdot\beta^2\cdot\frac{d+\log(1/\delta)}{n}}\left\|\Delta\right\|_{(\widehat{\Sigma}_{\theta^*}+\lambda I)} + 4(\lambda\gamma\beta^2(1-2\varepsilon) + 2\alpha_2\beta(1+2\beta\gamma(1-2\varepsilon)\alpha_1 B))B^2.
\end{aligned}$$

Solving for the above inequality, we get

$$\left\|\Delta\right\|_{(\widehat{\Sigma}_{\theta^*}+\lambda I)} \leqslant C_2\cdot\sqrt{\frac{1}{\gamma^2\beta^2(1-2\varepsilon)^2}\cdot\frac{d+\log(1/\delta)}{n} + (\lambda + \frac{\alpha_2}{\gamma\beta(1-2\varepsilon)} + \alpha_1\alpha_2 B)B^2}$$

for some constant $C_2 > 0$. Hence, we get

$$\left\|\widehat{\theta}_n - \theta^*\right\|_{(\widehat{\Sigma}_{\theta^*}+\lambda I)} \leqslant \frac{C}{\gamma\beta(1-2\varepsilon)}\cdot\sqrt{\frac{d+\log(1/\delta)}{n}} + C'\cdot B\sqrt{\lambda + \frac{\alpha_2}{\gamma\beta(1-2\varepsilon)} + \alpha_1\alpha_2 B},$$

for some $C, C' > 0$. This completes our proof.

E.4 Proof of Theorem C.4

Define the population covariance matrix of centered gradients of the function $f_\theta(s,a)$ under policy π:

$$\Sigma_\pi = \mathbb{E}_{s\sim\rho, a\sim\pi(\cdot|s)}\left[g_\theta(s,a)g_\theta(s,a)^\top\right],$$

where $g_\theta(s,a) = \nabla f_\theta(s,a) - \mathbb{E}_{a'\sim\pi(\cdot|s)}[\nabla f_\theta(s,a')]$ denotes the centered features. For log-linear policies, $\nabla f_\theta(s,a) = \phi(s,a)$ and $g_\theta(s,a) = \phi(s,a) - \mathbb{E}_\theta[\phi(s,a')]$, which gives

$$\Sigma_\pi = \mathbb{E}_{s\sim\rho, a\sim\pi(\cdot|s)}\left[\phi(s,a)\phi(s,a)^\top\right] - \mathbb{E}_{s\sim\rho, a\sim\pi(\cdot|s)}\left[\phi(s,a)\right]\mathbb{E}_{s\sim\rho, a\sim\pi(\cdot|s)}\left[\phi(s,a)\right]^\top .$$

Define sample covariance and population matrix of feature differences under clean data $\mathcal{D}$;

$$\widehat{\Sigma} = \frac{1}{n}\sum_{i=1}^{n}\left(\phi(s_i,a_{w,i}) - \phi(s_i,a_{l,i})\right)\left(\phi(s_i,a_{w,i}) - \phi(s_i,a_{l,i})\right)^\top,$$

$$\Sigma_{\pi,\mathrm{diff}} = \mathbb{E}_{s\sim\rho, a,a'\sim\pi(\cdot|s)}\left[\left(\phi(s,a) - \phi(s,a')\right)\left(\phi(s,a) - \phi(s,a')\right)^\top\right].$$

Since a, a' are indpendent samples from policy $\pi(\cdot|s)$, it holds that

$$\Sigma_{\pi,\mathrm{diff}} = 2\Sigma_\pi$$

Since $(a_{w,i}, a_{li,i})$ are independent samples from SFT policy $\pi_{\mathrm{sft}}(\cdot|s)$, by matrix concentration inequality Tropp et al. (2015), we have the following lemma.

Lemma E.1. *With probability at least $1-\delta$, for some universal constant C, we have*

$$\left\|\widehat{\Sigma} - \Sigma_{\pi_{\mathrm{sft}},\mathrm{diff}}\right\|_2 \le C\sqrt{d\log(4d/\delta)/n}.$$

This implies, for $\lambda \ge C\sqrt{d\log(4d/\delta)/n}$, with probability at least $1-\delta$:

$$\widehat{\Sigma} + \lambda I \succeq \Sigma_{\pi_{\mathrm{sft}},\mathrm{diff}} + \lambda I - C\sqrt{d\log(4d/\delta)/n} \succeq \Sigma_{\pi_{\mathrm{sft}},\mathrm{diff}} = 2\Sigma_{\pi_{\mathrm{sft}}}. \tag{14}$$

Now, we bound the sub-optimality gap conditioned on this high-confidence event. Since $r^*(s,a) \le r_{\max}$ for all (s,a), we have the sub-optimality gap:

$$\begin{aligned}
r^*(\pi^*) - r^*(\widehat{\pi}_n) &= \mathbb{E}_{s\sim\rho, a\sim\pi^*(\cdot|s)}\left[r^*(s,a)\right] - \mathbb{E}_{s\sim\rho, a\sim\widehat{\pi}_n(\cdot|s)}\left[r^*(s,a)\right] \\
&\le r_{\max}\mathbb{E}_{s\sim\rho}\left[\mathtt{TV}\left(\pi^*(\cdot|s), \widehat{\pi}_n(\cdot|s)\right)\right] \\
&\le r_{\max}\left[\mathbb{E}_{s\sim\rho}\sqrt{2\,\mathtt{KL}\left(\pi^*(\cdot|s), \widehat{\pi}_n(\cdot|s)\right)}\right] \\
&\le r_{\max}\sqrt{2\mathbb{E}_{s\sim\rho}\left[\mathtt{KL}\left(\pi^*(\cdot|s), \widehat{\pi}_n(\cdot|s)\right)\right]},
\end{aligned}$$

where the second step follows from Pinsker's inequality and the last step is due to Jensen's inequality.

Since the neural policy class (2) belongs to the exponential family of distributions, it holds that $\mathtt{KL}\left(\pi_\theta(\cdot|s), \pi_{\theta'}(\cdot|s)\right) = \mathcal{B}_\mathcal{L}(\theta', \theta)$, where $\mathcal{B}_\mathcal{L}$ is the Bregman divergence with potential function $\mathcal{L}_s(\theta) = \log\sum_{a'\in\mathcal{A}} f_\theta(s,a')$. It is defined as

$$\mathcal{B}_{\mathcal{L}_s}(\theta', \theta) \stackrel{\mathrm{def}}{=} \mathcal{L}_s(\theta') - \mathcal{L}_s(\theta) - \langle\theta' - \theta, \nabla\mathcal{L}_s(\theta)\rangle.$$

Therefore, we get

$$\mathtt{KL}\left(\pi^*(\cdot|s), \widehat{\pi}_n(\cdot|s)\right) = \mathcal{L}_s(\widehat{\theta}_n) - \mathcal{L}_s(\theta^*) - \langle\widehat{\theta}_n - \theta^*, \nabla\mathcal{L}_s(\theta^*)\rangle = \frac{1}{2}(\widehat{\theta}_n - \theta^*)^\top\nabla^2\mathcal{L}_s(\theta)(\widehat{\theta}_n - \theta^*)$$

for some $\theta \in \{t\theta^* + (1-t)\widehat{\theta}_n : t \in [0,1]\}$ using Taylor's approximation.

Now, for log-linear policy, we have $\mathbb{E}_{s\sim\rho}\left[\nabla^2\mathcal{L}_s(\theta)\right] = \Sigma_{\pi_\theta}$. Then, we can upper bound the sub-optimality gap using relative condition number κ as

$$\begin{aligned}
r^*(\pi^*) - r^*(\widehat{\pi}_n) &= r_{\max}\left\|\widehat{\theta}_n - \theta^*\right\|_{\Sigma_{\pi_\theta}} \\
&= r_{\max}\left\|\widehat{\theta}_n - \theta^*\right\|_{\widehat{\Sigma}+\lambda I}\sqrt{\frac{(\widehat{\theta}_n-\theta^*)^\top\Sigma_{\pi_\theta}(\widehat{\theta}_n-\theta^*)}{(\widehat{\theta}_n-\theta^*)^\top(\widehat{\Sigma}+\lambda I)(\widehat{\theta}_n-\theta^*)}} \\
&\le \frac{r_{\max}}{\sqrt{2}}\left\|\widehat{\theta}_n - \theta^*\right\|_{\widehat{\Sigma}+\lambda I}\sqrt{\frac{(\widehat{\theta}_n-\theta^*)^\top\Sigma_{\pi_\theta}(\widehat{\theta}_n-\theta^*)}{(\widehat{\theta}_n-\theta^*)^\top\Sigma_{\pi_{\text{sft}}}(\widehat{\theta}_n-\theta^*)}} \\
&\le \frac{r_{\max}}{\sqrt{2}}\left\|\widehat{\theta}_n - \theta^*\right\|_{\widehat{\Sigma}+\lambda I}\sqrt{\sup_{v\in\mathbb{R}^d}\frac{v^\top\Sigma_{\pi_\theta}v}{v^\top\Sigma_{\pi_{\text{sft}}}v}} \\
&= \frac{r_{\max}\sqrt{\kappa_{\pi_\theta}}}{\sqrt{2}}\left\|\widehat{\theta}_n - \theta^*\right\|_{\widehat{\Sigma}+\lambda I} \le \frac{r_{\max}\sqrt{\kappa}}{\sqrt{2}}\left\|\widehat{\theta}_n - \theta^*\right\|_{\widehat{\Sigma}+\lambda I}.
\end{aligned}$$

Here, the third step follows from (14), the fifth step holds by definition of (relative) condition number and in the final step, we use that $\kappa = \max_{\pi\in\Pi}\kappa_\pi$. This completes our proof.

E.5 Proof of Lemma C.5

Recall that $\widehat{r}_\theta(s,a) = \log\frac{\pi_\theta(a|s)}{\pi_{\text{sft}}(a|s)}$ denotes the implicit reward defined by trained and SFT policies π_θ and π_{sft}. Then, we have the expected margin gap under clean distribution

$$\begin{aligned}
\mathcal{M}(\pi^*) - \mathcal{M}(\widehat{\pi}_n) &= \mathbb{E}_{s\sim\rho,(y_w,y_l)\sim\pi_{\text{sft}}}\left[\left[\widehat{r}_{\theta^\star}(a_w|s) - \widehat{r}_{\theta^\star}(a_l|s)\right] - \left[\widehat{r}_{\widehat{\theta}_n}(a_w|s) - \widehat{r}_{\widehat{\theta}_n}(a_l|s)\right]\right] \\
&= \mathbb{E}_{s\sim\rho,(y_w,y_l)\sim\pi_{\text{sft}}}\left[\log\frac{\pi_{\theta^*}(a_w|s)}{\pi_{\theta^*}(a_l|s)} - \log\frac{\pi_{\widehat{\theta}_n}(a_w|s)}{\pi_{\widehat{\theta}_n}(a_l|s)}\right] \\
&= \mathbb{E}_{s\sim\rho,(y_w,y_l)\sim\pi_{\text{sft}}}\left[\left[f_{\theta^*}(s,y_w) - f_{\theta^*}(s,y_l)\right] - \left[f_{\widehat{\theta}_n}(s,y_w) - f_{\widehat{\theta}_n}(s,y_l)\right]\right] \\
&= \mathbb{E}_{s\sim\rho,(y_w,y_l)\sim\pi_{\text{sft}}}\left[\left[f_{\theta^*}(s,y_w) - f_{\widehat{\theta}_n}(s,y_w)\right] - \left[f_{\theta^*}(s,y_l) - f_{\widehat{\theta}_n}(s,y_l)\right]\right] \\
&\le \mathbb{E}_{s\sim\rho,(y_w,y_l)\sim\pi_{\text{sft}}}\left[\left|f_{\theta^*}(s,y_w) - f_{\widehat{\theta}_n}(s,y_w)\right| + \left|f_{\theta^*}(s,y_l) - f_{\widehat{\theta}_n}(s,y_l)\right|\right] \\
&\le 2\alpha_1\left\|\theta^* - \widehat{\theta}_n\right\|,
\end{aligned}$$

where the final step follows from Assumption C.1. Now, assuming $\widehat{\Sigma}$ to be invertible for log-linear policies, we get from (9):

$$\left\|\widehat{\theta}_n - \theta^*\right\|_{\widehat{\Sigma}} = O\Big(\frac{1}{\sqrt{\lambda_{\min}(\widehat{\Sigma})}}\frac{1}{\gamma\beta(1-2\varepsilon)}\sqrt{\frac{d}{n}}\Big).$$

Setting $\alpha_1 = LB$ for log-linear policies, we obtain

$$\mathcal{M}(\pi^*) - \mathcal{M}(\widehat{\pi}_n) = O\Big(\frac{1}{\sqrt{\lambda_{\min}(\widehat{\Sigma})}}\frac{2LB}{\gamma\beta(1-2\varepsilon)}\sqrt{\frac{d}{n}}\Big),$$

which completes our proof.