Active Preference Optimization for Sample Efficient RLHF

Nirjhar Das [1] **Souradip Chakraborty** [2] **Aldo Pacchiano** [3] **Sayak Ray Chowdhury** [1]

Abstract

Reinforcement Learning from Human Feedback (RLHF) is pivotal in aligning Large Language Models (LLMs) with human preferences. Although aligned LLMs have shown remarkable abilities in numerous tasks, their reliance on high-quality human preference data creates a costly bottleneck. Current methods for RLHF rely on uniformly picking prompt-generation pairs from a dataset of prompt-generations, to collect human feedback. For limited number of human feedback samples, we show that this leads to sub-optimal alignment. Next, we develop an active-learning algorithm, *Active Preference Optimization* (`APO`), which significantly enhances model alignment by querying preference data for the most important samples, thus achieving superior performance at a small sample budget. We analyze the theoretical performance guarantees of `APO` showing that the suboptimality gap of the policy learned via `APO` scales as $O(1/\sqrt{T})$ for a sample budget of T. We perform detailed experimental evaluations on practical preference datasets to validate `APO`'s efficacy over the existing methods, establishing it as a sample-efficient and practical solution of alignment in a cost-effective and scalable manner.

1. Introduction

Reinforcement Learning from Human Feedback (RLHF) has proven highly effective in aligning Large Language Models (LLMs) with human preferences (Christiano et al., 2017; Ouyang et al., 2022). It involves collecting extensive data, each comprising a prompt (context), a pair of responses (actions), and a preference indicating which response is better. A reward model is learned to classify the responses and subsequently, a policy is trained to generate responses with high rewards while minimizing divergence from a reference policy. Most practical implementations of RLHF pick prompts uniformly at random from a given pool. This is followed by first generating a pair of responses for each sampled prompt based on a supervised fine tuned (SFT) policy, and then sending all the pairs to human labelers (Stiennon et al., 2020). While excessive but low-quality data (incorrect or ambiguous preferences) can degrade performance of the aligned policy, high-quality (correct preferences) but scarce data might also not be able enhance it. Moreover, high-quality samples are expensive to collect since this demands a certain level of expertise from labelers.

While uniform prompt sampling as a simple approach has been effective for aligning LLMs so far, one might need more involved sampling strategies to deliver better model alignment under a fixed budget of labeling. Against this backdrop, we make the following contributions:

- **Sub-optimality of randomly sampling prompts:** We design a hard instance of the RLHF problem for which we show that an algorithm that collects preferences by sampling contexts (prompts) uniformly at random, and then trains the policy based on this data suffers $\Omega(1)$ suboptimality gap.
- **Adaptive algorithm via active prompt sampling:** We propose *Active Preference Optimization* (`APO`), an active learning algorithm for RLHF that actively selects to collect preference data, and show that suboptimality gap of `APO` scales as $O(\sqrt{\kappa/T})$ where T is sample budget and κ is a problem-dependent non-linearity factor.
- **Empirical evidence:** We propose a batch version of `APO` which is also computationally efficient. We experiment with GPT-2 (Radford et al., 2019) on IMDb sentiment dataset (Maas et al., 2011) and with Gemma-2b (Team et al., 2024) on Anthropic-HH dataset (Bai et al., 2022) demonstrating significant improvement over uniform sampling.

Related work. Ji et al. (2024) proposes an RLHF strategy that selects one action adaptively and the other randomly, but do not actively sample contexts. Muldrew et al. (2024) actively select both contexts and actions using an uncertainty based heuristic, but don't provide any theoretical justification. Mehta et al. (2023) proposes an active learning strategy with decreasing sub-optimality gap. However, they choose one out of two actions randomly at every round, which is wasteful in practice. They also assume that both rewards and probabilities of an action winning over any uniformly chosen action are linear functions of a common feature map,

[1] Microsoft Research India [2] University of Maryland, College Park [3] Broad Institute/Boston University. Correspondence to: Sayak Ray Chowdhury <t-sayakr@microsoft.com>.

Work presented at TF2M workshop at ICML 2024, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

which is a restrictive assumption on the model.

2. Problem Setup

We have a set of contexts $\mathcal{X}$ and a set of possible actions per context $\mathcal{A}$. To learn using preference feedback, the agent selects a tuple (x, a, a') to present to a human labeller who then reveals a binary preference y which takes value 1 if a wins over a' and 0 otherwise. We assume that y is sampled from distribution conditioned on (x, a, a') given as

$$\mathbb{P}_{\theta^*}\left[y=1|x,a,a'\right] = \frac{\exp(r^*(x,a))}{\exp(r^*(x,a)) + \exp(r^*(x,a'))} \ ,$$

known as the Bradley-Terry-Luce (BTL) model (Bradley and Terry, 1952; Luce, 2012). Here r^* is a latent reward function. The goal of the agent is to first learn the reward over T rounds of sequential interaction with the labeller, collecting dataset $\mathcal{D} = (x_s, a_s, a'_s, y_s)_{s=1}^T$, and then employ the learned reward to train a policy $\pi : \mathcal{X} \to \mathcal{A}$, which will eventually fetch high latent rewards $r^*(x, a)$.

In this work, we consider linear latent rewards $r^*(x, a) = \phi(x, a)^\top \theta^*$, where $\theta^* \in \mathbb{R}^d$ are unknown reward parameters, and $\phi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$ is some known and fixed feature map. For instance, such a ϕ can be constructed by removing the last layer of a pre-trained language model, and in that case, θ^* correspond to the weights of the last layer. With this model, one can equivalently write the probability of sampling $y_s = 1$ given (x_s, a_s, a'_s) as

$$\mathbb{P}_{\theta^*}[y_s=1|x_s,a_s,a'_s]=\sigma(\phi(x_s,a_s)^\top\theta^*-\phi(x_s,a'_s)^\top\theta^*),$$

where $\sigma(w) = \frac{1}{1+e^{-w}}$ is the sigmoid function. We let $z_s = \phi(x_s, a_s) - \phi(x_s, a'_s)$ denote the differential feature of actions a_s and a'_s at state x_s. Thus we can write $\mathbb{P}_\theta\left[y_s=1|x_s,a_s,a'_s\right]=\sigma(z_s^\top\theta)$. With this, the reward parameters θ^* are estimated by minimizing the log-loss, which is also equivalent to *maximum likelihood estimation* (MLE).

At round t, the MLE of θ^* is computed using preference dataset $\{(x_s, a_s, a'_s, y_s)\}_{s=1}^{t-1}$ as $\widehat{\theta}_t = \operatorname{argmin}_{\theta\in\Theta} \mathcal{L}_t(\theta)$, where the log-loss $\mathcal{L}_t(\theta)$ is given by

$$\mathcal{L}_t(\theta)=-\sum_{s=1}^{t-1}y_s\log(\sigma(z_s^\top\theta))+(1-y_s)\log(1-\sigma(z_s^\top\theta)). \quad (1)$$

The above optimization problem is convex when the constraint set Θ is convex, and hence can be solved using standard algorithms (Hazan et al., 2016).

Performance Measure. Our goal is to learn a policy over the collected data $\mathcal{D}$, which has high rewards or, equivalently, low suboptimality. Formally, the suboptimality gap of a learned policy π_T after collecting T samples by an algorithm of choice is defined as

$$R(T)=\max_{x\in\mathcal{X}}\max_{a\in\mathcal{A}}\left[r^*(x,a)-r^*(x,\pi_T(x))\right] \ . \quad (2)$$

The suboptimality gap is the worst possible difference in latent rewards between the best action and the policy's action over the set of contexts.

3. Active Preference Optimization

3.1. Is Uniform Prompt Sampling Good Enough?

We first characterize the pitfall of learning via a uniformly random prompt sampling strategy (details in Appendix A).

Definition 3.1 (Uniform Learner)**.** Say an algorithm `Alg` samples T contexts uniformly at random from a set $\mathcal{X}$ and for each context $x_t, t \in \{1, \dots, T\}$, picks two actions a_t, a'_t of its choice from a set $\mathcal{A}$. For each triplet (x_t, a_t, a'_t), `Alg` then queries the BTL model parameterized by θ^* and observes a stochastic preference $y_t \in \{0, 1\}$ between the actions. `Alg` then solves an MLE over these T preference data, and learns a greedy policy with respect to the MLE. We call such an algorithm `Alg` a Uniform Learner.

Theorem 3.2 (Lower bound on sub-optimality gap)**.** *There exists a $T \in \mathbb{N}$ and a problem instance $(\mathcal{X}, \mathcal{A}, \theta^*)$ for which the policy learnt by a Uniform Learner `Alg` suffers $\Omega(1)$ suboptimality gap with high probability.*

Theorem 3.2 highlights need for learners that use samples effectively. Next, we propose a strategy which actively selects contexts so that suboptimality goes down as $1/\sqrt{T}$.

3.2. Our Approach: Active Prompt Sampling

We present the algorithm for active context and action selection in RLHF (Algorithm 1). At each round t, our algorithm proceeds by computing the MLE estimate $\widehat{\theta}_t$ based on the data obtained in the past $t-1$ steps (see (1)). Based on $\widehat{\theta}_t$, our goal is to maximize exploration. To do this, for a context $x \in \mathcal{X}$, we compute the uncertainty $b_t(x, a, a')$ for each action (a, a') available for that context and choose the one which maximizes this, i.e., we choose

$$(a_t(x), a'_t(x)) = \operatorname*{argmax}_{(a,a')\in\mathcal{A}\times\mathcal{A}} b_t(x,a,a') \ , \quad (3)$$

where $b_t(x, a, a') = \|\phi(x, a) - \phi(x, a')\|_{H_t^{-1}(\widehat{\theta}_t)}$. Here $H_t(\widehat{\theta}_t)$ is a matrix that describes a confidence ellipsoid around the unknown reward parameter θ^* after $t-1$ steps of data collection. For any $\theta \in \Theta$, this is defined as

$$H_t(\theta)=\nabla^2\mathcal{L}_t(\theta)+\lambda\mathbf{I}_d=\sum\nolimits_{s=1}^{t-1}\dot{\sigma}(z_s^\top\theta)z_sz_s^\top+\lambda\mathbf{I}_d \ . \quad (4)$$

Intuitively, the confidence ellipsoid keeps shrinking along whichever direction (in $\mathbb{R}^d$) we decide to explore. Thus, for a given context x, choosing the pair $(a_t(x), a'_t(x))$ maximally reduces the uncertainty among all other possible action duels. However, our algorithm picks not only the action pair that maximizes uncertainty, but also the context that increases it the most, i.e.,

$$x_t = \operatorname{argmax}_{x\in\mathcal{X}} b_t(x, a_t(x), a'_t(x)) \quad (5)$$

Algorithm 1 APO (Theoretical version)

```
Require: Context set $\mathcal{X}$, action set $\mathcal{A} = [K]$, feature map $\phi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^d$, regularization $\lambda > 0$, and failure probability $\delta \in (0, 1]$
1: Initialize $\widehat{\theta}_1 = 0$
2: for $t = 1, \ldots, T$ do
3:     Choose the triplet $(x_t, a_t, a'_t)$ using (3) and (5).
4:     Observe preference feedback $y_t \sim \texttt{Ber}(\sigma(z_t^\top \theta^*))$, where $z_t = \phi(x_t, a_t) - \phi(x_t, a'_t)$.
5:     Compute reward estimate $\widehat{\theta}_{t+1}$ that minimizes the constrained log-loss (1).
6:     Compute (scaled) design matrix $H_{t+1}(\widehat{\theta}_{t+1})$ via (4).
7: Compute final policy $\pi_T(x)$ using (6)
```

Algorithm 2 APO (Practical version)

```
Require: Prompt-generation pairs $\mathcal{M} = \{(x, a, a')\}$, sample budget $T$, encoder $\phi$, SFT policy $\pi_{\text{SFT}}$, log-loss $\mathcal{L}$, batch size $B$, uncertainty regularizer $\lambda > 0$, KL regularizer $\beta > 0$, learning rate $\eta > 0$
1: Initialize $V_1 = \lambda I, \widehat{\theta}_1 = 0, \mathcal{D} = \emptyset$
2: for batch $t = 1, \ldots, \lfloor T/B \rfloor$ do
3:     Compute $b_t(x, a, a') = \|\phi(x, a) - \phi(x, a')\|_{V_t^{-1}}$ for each $(x, a, a') \in \mathcal{M}$
4:     Initialize $\mathcal{M}_t = \emptyset$
5:     for $j = 1, \ldots, B$ do
6:         Pick $(x_{t,j}, a_{t,j}, a'_{t,j}) = \operatorname{argmax}_{(x,a,a') \in \mathcal{M} \setminus \mathcal{M}_t} b_t(x, a, a')$
7:         $\mathcal{M}_t \leftarrow \mathcal{M}_t \cup \{(x_{t,j}, a_{t,j}, a'_{t,j})\}$
8:         Observe $y_{t,j}$; $\mathcal{D} \leftarrow \mathcal{D} \cup \{(x_{t,j}, a_{t,j}, a'_{t,j}, y_{t,j})\}$
9:     Update $\widehat{\theta}_{t+1} \leftarrow \texttt{Gradient-step}(\mathcal{L}, \widehat{\theta}_t, \mathcal{D}, \eta)$
10:    Update $V_{t+1} \leftarrow V_t + \sum_{j=1}^{B} z_{t,j} z_{t,j}^\top$, where $z_{t,j} = \phi(x_{t,j}, a_{t,j}) - \phi(x_{t,j}, a'_{t,j})$
11: Define reward $\widehat{r}_T(x, a) = \phi(x, a)^\top \widehat{\theta}_{\lfloor T/B \rfloor + 1} \forall (x, a)$
12: Compute final policy $\pi_T \leftarrow \texttt{PPO}(\pi_{\text{SFT}}, \widehat{r}_T(x, a), \beta)$
```

This is a crucial step in our approach that ensures that the uncertainty of the reward function over all contexts decreases at a fast rate which in turn ensures low suboptimality gap of our policy. After T time steps, we define $\theta_T = \frac{1}{T}\sum_{s=1}^{T} \widehat{\theta}_t$ as the average of all the past parameter estimates. Our final policy π_T for any context $x \in \mathcal{X}$ is to play the action that maximizes the reward parameterized by θ_T, i.e.,

$$\pi_T(x) = \operatorname*{argmax}_{a \in \mathcal{A}} \widehat{r}_T(x, a) = \operatorname*{argmax}_{a \in \mathcal{A}} \phi(x, a)^\top \theta_T \ . \quad (6)$$

3.3. Suboptimality Gap of APO

We make the following assumption which is standard in RLHF literature (Zhu et al., 2023; Chowdhury et al., 2024).

Assumption 3.3 (Boundedness)**.** (a) θ^* lies in the set $\Theta = \{\theta \in \mathbb{R}^d | \langle \mathbf{1}, \theta \rangle = 0, ||\theta|| \leq S\}$. (b) Features are bounded, i.e., $||\phi(x, a)|| \leq 1, \forall\ (x, a) \in \mathcal{X} \times \mathcal{A}$.

Now, we define a key quantity that captures learning complexity under the BTL preference model:

$$\kappa = \max_{x \in \mathcal{X}, a, a' \in \mathcal{A}} \max_{\theta \in \Theta} \frac{1}{\dot{\sigma}(\phi(x, a)^\top \theta - \phi(x, a')^\top \theta)} \ . \quad (7)$$

κ specifies difficulty in learning via the worst-case nonlinearity in preference feedback. Our algorithm enjoys the following guarantee (proof is presented in appendix B).

Theorem 3.4 (Sub-optimality gap)**.** *Let $\delta \in (0, 1]$. Under Assumption 3.3, with probability at least $1 - \delta$, APO (Algorithm 1) enjoys the suboptimality gap*

$$R(T) = O\left(\gamma_T(\delta) \sqrt{S \log\left(1 + \frac{T}{\lambda \kappa d}\right) \frac{\kappa d}{T}}\right) .$$

where $\lambda = \frac{1}{4S^2(2+2S)^2}$ and $\gamma_t(\delta) = CS\sqrt{d \log \frac{St}{d} + \log \frac{t}{\delta}}$.

Remark 3.5 (Dependence on κ). κ can be exponential in the parameter norm S in the worst-case. In logistic bandits, the state-of-the-art regret guarantee is κ-independent – the dependence is only in lower order term (Lee et al., 2023). We believe the $\sqrt{\kappa}$ dependence is unavoidable in the RLHF setting as the bound here is for real-valued rewards $r^*(x, a) = \phi(x, a)^\top \theta^*$ instead of the sigmoid rewards $\sigma(\phi(x, a)^\top \theta^*)$ in logistic bandits.

4. Experiments

In this Section, we first present a practical version of APO, which largely follows the former with minor changes adapted for computationally efficient implementation required in large scale experiments. In this practical version (Algorithm 2), we access preference data in batches instead of being fully online. For each batch t, we first compute the uncertainty $b_t(x, a, a')$ of each triplet $(x, a, a') \in \mathcal{D}$ (Step 3). In order to maximize exploration, only those B triplets (x, a, a') are queried in a batch that have the highest uncertainty $b_t(x, a, a')$ and the preference data is collected in batches and stored in a buffer $\mathcal{D}$. At the end of each batch t, we update the parameter estimate $\widehat{\theta}_t$ via a Gradient-Step, which is a blackbox gradient-descend-based optimization algorithm (e.g. Adam (Kingma and Ba, 2015)) on the log-loss (1) over the dataset $\mathcal{D}$. Finally, after the budget T is exhausted, we first learn an estimate $\widehat{r}_T$ of the latent reward model r^*, and then the aligned policy π_T is learned via proximal policy optimization (PPO), which takes as input the SFT policy π_{SFT} and the learnt reward model $\widehat{r}_T$, and aligns π_{SFT} with the preference dataset $\mathcal{D}$ (Ouyang et al., 2022). Next we discuss the results of our experiments (details in Appendix E). Hereafter we denote by Random the random prompt sampling baseline.

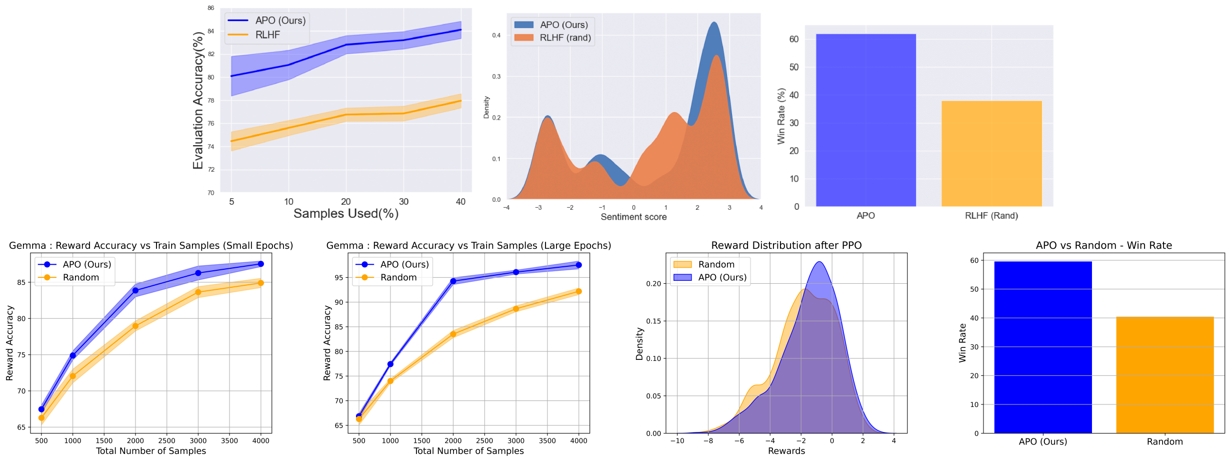

Figure 1. **Top Row: Left:** Evaluation accuracy of trained reward model vs. no. of samples (in percentage) comparing our algorithm (APO) with Random. Eval accuracy of APO even with only 5% active samples is higher than that of Random with 40% samples. **Middle:** Sentiment score distribution of aligned policies trained on reward model learned with APO (using only 10% samples) and on Random's highest accuracy reward model (using 40% samples). Generations by APO-trained reward is more shifted towards positive showing better reward learning than Random. **Right:** Win rates of APO against Random. APO outperforms Random by 60 : 40 win rate. **Bottom Row: Left and 2nd Left** Evaluation accuracy of trained reward model vs. no. of samples comparing our algorithm (APO) with Random, when the number of epochs is 5 (**Left**) and 20 (**2nd Left**). Eval. accuracy of APO is higher than the Random in both cases. **2nd Right:** Reward distribution of APO, SFT-policy and Random for generations on prompts in the test dataset. Clearly, APO has a much better alignment compared to Random. **Right:** Win rates of APO and Random. APO outperforms Random by 60 : 40 win rate.

4.1. Results on Controlled Sentiment Generation Task

In this experiment, the task is to produce positive sentiment texts for given prompts in the IMDb dataset (Maas et al., 2011) with GPT-2 (Radford et al., 2019) as the LLM. Reward is learnt using 5000 samples (out of 8000 in the training set) for both APO and Random. With these learnt reward models, the GPT-2 model is finetuned on the prompts in the training set to obtain APO-policy and Random-policy, respectively. We then use the finetuned models to generate responses for the prompts in the test set. A pretrained GPT-2 sentiment classifier is used as model for the latent reward r^*. The test set consisted of 2000 samples.

Reward Evaluation. The $\%$ of samples in the test set for which APO assigns larger reward to *chosen* responses is higher than that for Random, even when APO is trained with 5% samples and Random is trained with 40%. The result is shown in Fig. 1 top row.

Win Rate. We compare reward distribution of responses generated by APO-policy and Random-policy. Using the pretrained sentiment classifier, we also compute the win rate of APO-policy over Random-policy Fig. 1 shows that APO outperforms Random by a significant margin.

4.2. Results on Single-turn Dialogue task

We use Anthropic-HH (Bai et al., 2022) preference dataset and Gemma-2b (Team et al., 2024) language model. We construct a dataset by collecting the prompts with single-turn dialogues and then putting these samples into three buckets based on reward difference between *chosen* and *rejected* responses (using Mistral-7b reward model). Out of these three buckets, we take more samples from the buckets with smaller reward difference and less samples from the bucket with larger reward difference, to construct our final training dataset. Such a dataset highlights the importance of selecting prompts carefully to obtain useful information regarding learning. The test set had a separate set of 2000 samples chosen randomly from Anthropic-HH test set.

Reward Evaluation. We compare the reward models learnt by APO and Random by computing the $\%$ of samples in the test set for which the models assign higher reward to the *chosen* responses. The results can be seen in Fig. 1 bottom row. We also study how this accuracy changes with number of batches keeping sample budget same. We see that APO always outperforms Random.

Win Rate. Based on reward models learnt by APO and Random, we finetune the SFT plicy with PPO to obtain APO-policy and Random-policy respectively. Then we generate responses for prompts in the test set using APO and Random, and get them evaluated by Mistral-7b reward model. Reward distribution of all these policies and win rate of APO-policy over Random-policy is shown in Fig. 1. Clearly, APO performs much better than the baseline.

Concluding Remarks. We showed that the simple approach of sampling prompts uniformly at random could suffer a constant suboptimality gap when aligning a language model policy with human preferences. To mitigate this, we proposed an active prompt selection algorithm APO to achieve an $O(1/\sqrt{T})$ suboptimality gap. This is a general approach and can be applied to other alignment methods like DPO (Rafailov et al., 2023) and IPO (Azar et al., 2024). We keep this as a promising future direction towards building sample-efficient algorithms for language model alignment.

References

Marc Abeille, Louis Faury, and Clément Calauzènes. Instance-wise minimax-optimal algorithms for logistic bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 3691–3699. PMLR, 2021.

Alex Ayoub, Zeyu Jia, Csaba Szepesvari, Mengdi Wang, and Lin Yang. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*, pages 463–474. PMLR, 2020.

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 4447–4455. PMLR, 2024.

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, 2022. URL https://arxiv.org/pdf/2204.05862.pdf.

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*, pages 3773–3793. PMLR, 2022.

Sayak Ray Chowdhury, Anush Kini, and Nagarajan Natarajan. Provably robust dpo: Aligning language models with noisy feedback. *International Conference in Machine Learning*, 2024.

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

Louis Faury, Marc Abeille, Kwang-Sung Jun, and Clément Calauzènes. Jointly efficient and optimal algorithms for logistic bandits. In *International Conference on Artificial Intelligence and Statistics*, pages 546–580. PMLR, 2022.

Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

Kaixuan Ji, Jiafan He, and Quanquan Gu. Reinforcement learning from human feedback with active queries. *arXiv preprint arXiv:2402.09401*, 2024.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA, 2015.

Junghyun Lee, Se-Young Yun, and Kwang-Sung Jun. Improved regret bounds of (multinomial) logistic bandits via regret-to-confidence-set conversion, 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 *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

Viraj Mehta, Vikramjeet Das, Ojash Neopane, Yijia Dai, Ilija Bogunovic, Jeff Schneider, and Willie Neiswanger. Sample efficient reinforcement learning from human feedback via active exploration. *arXiv preprint arXiv:2312.00267*, 2023.

William Muldrew, Peter Hayes, Mingtian Zhang, and David Barber. Active preference learning for large language models. *arXiv preprint arXiv:2402.08114*, 2024.

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.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

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, 2023.

Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. *Advances in Neural Information Processing Systems*, 26, 2013.

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.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

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.

Active Preference Optimization for Sample Efficient RLHF: Appendix

A. Proof of Theorem 3.2

Let the number of contexts be $|\mathcal{X}| = N$. Assume $T \ll N$. Otherwise, if sample budget $T > N$, then one can just collect data for every context, and the setting becomes trivial. We divide $\mathcal{X}$ into two disjoint subsets: a *good* set $\mathcal{X}_g$ and a *bad* set $\mathcal{X}_b$. We assume w.l.o.g. that $|\mathcal{X}_b| = 1$, and we denote the *bad* context by b. Let the action set be $\mathcal{A} = \{a, a'\}$ for all contexts, and let a has higher reward than a'. Let $\phi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^2$ be a feature map and $z_x = \phi(x,a) - \phi(x,a')$ be the feature difference vector at context x. Fix an $\alpha > 0$ and consider the problem instance:

$$\theta^* = \alpha \begin{bmatrix} \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}^\top, \quad z_b = \begin{bmatrix} -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}^\top, \quad z_x = \begin{bmatrix} 1 & 0 \end{bmatrix}^\top, \forall\, x \in \mathcal{X}_g\,.$$

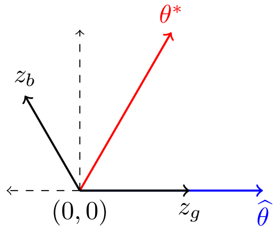

Figure 2. Visualization of lower bound instance. z_g represents feature difference vectors for *good* contexts. z_b represents feature difference for the *bad* context. $\theta^*, \widehat{\theta}$ are true and learnt parameters, respectively.

Note that $\|\theta^*\|_2 = \alpha$. From this construction (see Fig. 2), it is clear that both for *good* and *bad* contexts, $z_x^\top \theta^* = \alpha/2 > 0$, which implies that indeed action a has higher reward than a'.

Let $\mathcal{E}_1$ be the event that all the T sampled contexts are *good* (i.e. from $\mathcal{X}_g$). Since, under uniform sampling, for a random context X, $\mathbb{P}[X \in \mathcal{X}_g] = 1 - 1/N$, we have $\mathbb{P}[\mathcal{E}_1] = (1-1/N)^T$. Let $\mathcal{E}_2$ be the event that all observed preferences $y_1, \ldots y_T$ are equal to 1. Since, for a random preference Y given a context x, $\mathbb{P}[Y=1|x] = \sigma(\alpha/2)$, we have $\mathbb{P}[\mathcal{E}_2|\mathcal{E}_1] = \sigma(\alpha/2)^T$.

Now, under the event $\mathcal{E}_1 \cap \mathcal{E}_2$, the MLE $\widehat{\theta}$ constrained to the same norm as θ^* is given by

$$\widehat{\theta} = \operatorname*{argmin}_{\theta \in \mathbb{R}^2 : \|\theta\|_2 \le \alpha} \sum\nolimits_{t=1}^{T} \log\left(1 + e^{-\alpha\theta_1}\right) = \operatorname*{argmin}_{\theta \in \mathbb{R}^2 : \|\theta\|_2 \le \alpha} \log\left(1 + e^{-\alpha\theta_1}\right)\,.$$

It is easy to see that $\widehat{\theta} = \begin{bmatrix} \alpha & 0 \end{bmatrix}^\top$. For any $x \in \mathcal{X}_g$, the predicted reward difference between actions a and a' is $z_x^\top \widehat{\theta} = \alpha > 0$. Thus, $\widehat{\theta}$ predicts the better action a correctly for all *good* contexts $\mathcal{X}_g$. However, for context b, the reward difference is $z_b^\top \widehat{\theta} = -\frac{\alpha}{2} < 0$. Thus, $\widehat{\theta}$ wrongly predicts a' as the better action for the *bad* context b. This yields a a constant sub-optimality gap

$$R(T,b) = \phi(b,a)^\top \theta^* - \phi(b,a')^\top \theta^* = \alpha/2 = \Omega(1).$$

Finally, it remains to show that the event $\mathcal{E}_1 \cap \mathcal{E}_2$ happens with high probability. To this end, we choose $\alpha = 2\log(N-1)$ which yields $\sigma(\alpha/2) = 1 - 1/N$. This yields

$$\mathbb{P}[\mathcal{E}_1 \cap \mathcal{E}_2] = \mathbb{P}[\mathcal{E}_2|\mathcal{E}_1]\mathbb{P}[\mathcal{E}_1] = \left(1 - \frac{1}{N}\right)^{2T} \ge 1 - \frac{2T}{N}\,.$$

The last step uses $T \ll N$, which completes the proof.

B. Missing Proofs from Section 3

First we state the result from logistic bandit literature that characterizes the confidence set for the constrained maximum likelihood estimator. Here we give one version of the confidence set from (Lee et al., 2023) but note that similar guarantees are also derived in (Abeille et al., 2021).

Lemma B.1 (Confidence Set for MLE (Theorem 1 of (Lee et al., 2023))**.** *Let $\widehat{\theta}_t$ be the constrained maximum likelihood* *estimator after* $t-1$ *time steps defined as follows:*

$$\widehat{\theta}_t = \operatorname*{argmin}_{\theta \in \Theta} \left\{ -\sum_{s=1}^{t-1} y_s \log(\sigma(z_s^\intercal \theta)) + (1-y_s)\log(1-\sigma(z_s^\intercal \theta)) \right\} .$$

Now define the set

$$\mathcal{C}_t(\delta) = \{\theta \in \Theta : \mathcal{L}_t(\theta) - \mathcal{L}_(\widehat{\theta}) \leq \beta_t(\delta)^2\}$$

where $\beta_t(\delta) = \sqrt{10d\log\left(\frac{St}{4d}+e\right) + 2(e-2+S)\log\left(\frac{1}{\delta}\right)}$*. Then we have* $P(\forall t \geq 1, \theta^* \in \mathcal{C}_t(\delta)) \geq 1-\delta$*.*

The details of the proof can be found in section 3.1 of (Lee et al., 2023). Next we present another lemma that quantifies the parameter estimation error. Using this lemma and a novel self-concordance property, we will prove B.3.

Lemma B.2 (Lemma 6 of (Lee et al., 2023))**.** *Let* $\widehat{\theta}_t$ *be defined above. Further, let* $\theta^* \in C_t(\delta)$*. Then,*

$$\|\widehat{\theta}_t - \theta^*\|^2_{H_t(\theta^*)} \leq \gamma_t(\delta)^2 \coloneqq 2(2+2S)f(d,S,t,\delta)$$

where

$$f(d,S,t,\delta) \coloneqq 2(e-2)(2+2S)d\log(\frac{5St}{d}) + 2(e-2)(2+2S)\log(\frac{t}{\delta}) + \frac{5d}{4} + \frac{d^2}{16St}$$

Simplifying, $\gamma_t(\delta)^2 = CS^2\left(d\log\frac{St}{d} + \log\frac{t}{\delta}\right)$ *for some* $C > 0$*.*

The proof of the lemma can be found in appendix C.4.4 of (Lee et al., 2023). Now we are ready to present the proof of lemma B.3.

Lemma B.3. *Suppose* $\theta^* \in \mathcal{C}_t(\delta)$*. Then,* $\|\theta^* - \widehat{\theta}\|_{H_t(\widehat{\theta}_t)} \leq CS^{1/2}\gamma_t(\delta)$*.*

Proof. By Taylor's theorem, we have,

$$\begin{aligned}
\mathcal{L}_t(\widehat{\theta}_t) - \mathcal{L}_t(\theta^*) &= \nabla\mathcal{L}_t(\theta^*)^\intercal(\widehat{\theta}_t - \theta^*) + \int_{v=0}^{1}(1-v)(\widehat{\theta}-\theta^*)^\intercal\nabla^2\mathcal{L}_t(\theta^*)(\widehat{\theta}-\theta^*)dv \\
&= \nabla\mathcal{L}_t(\theta^*)^\intercal(\widehat{\theta}_t - \theta^*) + \sum_{s=1}^{t-1}\left[\int_{v=0}^{1}(1-v)\dot{\sigma}(z_s^\intercal\theta^* + v(z_s^\intercal\widehat{\theta}_t - z_s^\intercal\theta^*))dv\right](z_s^\intercal(\widehat{\theta}_t-\theta^*))^2 \\
&= \nabla\mathcal{L}_t(\theta^*)^\intercal(\widehat{\theta}_t - \theta^*) + \|\widehat{\theta}_t - \theta^*\|^2_{\tilde{G}_t(\theta^*,\widehat{\theta}_t)} - \lambda\|\widehat{\theta}_t - \theta^*\|^2
\end{aligned}$$

where we define $\tilde{G}_t(\theta^*,\widehat{\theta}_t) = \lambda\mathbf{I}_d + \sum_{s=1}^{t-1}\left[\int_{v=0}^{1}(1-v)\dot{\sigma}(z_s^\intercal\theta^* + v(z_s^\intercal\widehat{\theta}_t - z_s^\intercal\theta^*))dv\right]z_sz_s^\intercal$. Thus, we obtain,

$$\|\widehat{\theta}_t - \theta^*\|^2_{\tilde{G}_t(\theta^*,\widehat{\theta}_t)} = \mathcal{L}_t(\theta^*) - \mathcal{L}_t(\widehat{\theta}_t) + \nabla\mathcal{L}_t(\theta^*)^\intercal(\widehat{\theta}_t-\theta^*) + \lambda\|\widehat{\theta}_t - \theta^*\|^2$$

Now, from a novel self-concordant analysis (see lemma D.1), $H_t(\widehat{\theta}_t) \preccurlyeq C(2+2S)^2\tilde{G}_t(\theta^*,\widehat{\theta}_t)$ for some $C > 1.01$. Thus,

$$\begin{aligned}
\|\widehat{\theta}_t - \theta^*\|^2_{H_t(\widehat{\theta}_t)} &\leq C(2+2S)^2\|\widehat{\theta}_t - \theta^*\|^2_{\tilde{G}_t(\theta^*,\widehat{\theta}_t)} \\
&= C(2+2S)^2\left[\mathcal{L}_t(\theta^*) - \mathcal{L}_t(\widehat{\theta}_t) + \nabla\mathcal{L}_t(\theta^*)^\intercal(\widehat{\theta}_t-\theta^*) + \lambda\|\widehat{\theta}_t-\theta^*\|^2\right] \\
&\leq C(2+2S)^2\left[4\lambda S^2 + \beta_t(\delta)^2 + \nabla\mathcal{L}_t(\theta^*)^\intercal(\widehat{\theta}_t-\theta^*)\right] \qquad (8)
\end{aligned}$$

where the last inequality is because (a) $\widehat{\theta}_t, \theta^* \in \Theta$ which implies that $\|\theta^* - \widehat{\theta}_t\| \leq \text{diam}(\Theta) = 2S$ and (b) by lemma B.1, $\mathcal{L}_t(\theta^*) - \mathcal{L}_t(\widehat{\theta}_t) \leq \beta_t(\delta)^2$ since $\theta^* \in \mathcal{C}_t(\delta)$ by assumption.

Thereafter, from the proof of Lemma 6 of (Lee et al., 2023) it can be extracted that $|\nabla\mathcal{L}_t(\theta^*)^\intercal(\widehat{\theta}_t-\theta^*)| \leq \frac{\|\widehat{\theta}_t-\theta^*\|^2_{H_t(\theta^*)}}{2(2+2S)} +$

$f(d,S,t,\delta)$. Then using lemma B.2, the R.H.S of 8 can be bounded by $2f(d,S,t,\delta)$. Thus, we now obtain,

$$\begin{aligned}\|\widehat{\theta}_t - \theta^*\|^2_{H_t(\widehat{\theta}_t)} &\le C(2+2S)^2\left[4\lambda S^2 + \beta_t(\delta)^2 + 2f(d,S,t,\delta)\right]\\ &\le C(2+2S)^2\left[\frac{1}{(2+2S)^2} + \beta_t(\delta)^2 + \frac{\gamma_t(\delta)^2}{2(2+2S)}\right] && (\,\lambda = \tfrac{1}{4S^2(2+2S)^2})\\ &\le C(2+2S)^2\left[\frac{1}{(2+2S)} + \beta_t(\delta) + \frac{\gamma_t(\delta)}{\sqrt{2(2+2S)}}\right]^2\end{aligned}$$

Hence, we have,

$$\begin{aligned}\|\widehat{\theta}_t - \theta^*\|_{H_t(\widehat{\theta}_t)} &\le C(2+2S)\left[\frac{1}{(2+2S)} + \beta_t(\delta) + \frac{\gamma_t(\delta)}{\sqrt{2(2+2S)}}\right]\\ &= C(1 + (2+2S)\beta_t(\delta) + \sqrt{2+2S}\gamma_t(\delta)) = CS^{3/2}\sqrt{\left(d\log(\frac{St}{d}) + \log(\frac{t}{\delta})\right)}\ .\end{aligned}$$

□

Theorem B.4 (Suboptimality Upper Bound)**.** *Let* $\delta \in (0,1)$. *The suboptimality of the policy* π_T *specified at the end of* `APO` *(algorithm 1) after running the algorithm for* T *rounds is upper bounded with probability at least* $1-\delta$ *as follows:*

$$R(T) \le CS^{3/2}\sqrt{\left(d\log(\frac{ST}{d}) + \log(\frac{T}{\delta})\right)\log\left(1 + \frac{T}{\lambda\kappa d}\right)\frac{\kappa d}{T}}$$

Proof. Let the suboptimality gap for a context $x \in \mathcal{X}$ be denoted as $R(T,x)$. Thus,

$$\begin{aligned}R(T,x) &= \left(\phi(x,a^*(x)) - \phi(x,\pi_T(x))\right)^\intercal \theta^*\\ &\le \left(\phi(x,a^*(x)) - \phi(x,\pi_T(x))\right)^\intercal \theta^* + \left(\phi(x,\pi_T(x)) - \phi(x,a^*(x))\right)^\intercal\left(\frac{1}{T}\sum_{t=1}^{T}\widehat{\theta}_t\right)\\ & && (\pi_T(x) = \operatorname*{argmax}_{a\in\mathcal{A}}\phi(x,a)^\intercal\left(\tfrac{1}{T}\textstyle\sum_{i=1}^{T}\widehat{\theta}_t\right))\\ &= \left(\phi(x,a^*(x)) - \phi(x,\pi_T(x))\right)^\intercal(\theta^* - \frac{1}{T}\sum_{t=1}^{T}\widehat{\theta}_t)\\ &= \frac{1}{T}\sum_{t=1}^{T}\left(\phi(x,a^*(x)) - \phi(x,\pi_T(x))\right)^\intercal(\theta^* - \widehat{\theta}_t)\\ &\le \frac{1}{T}\sum_{t=1}^{T}\|\phi(x,a^*(x)) - \phi(x,\pi_T(x))\|_{H_t^{-1}(\widehat{\theta}_t)}\|\theta^* - \widehat{\theta}_t\|_{H_t(\widehat{\theta}_t)}\ . && \text{(Cauchy-Schwarz)}\end{aligned}$$

Here inequality (1) is due the definition of policy $\pi_T(x) := \operatorname{argmax}_{a\in\mathcal{A}}\phi(x,a)^\intercal\left(\frac{1}{T}\sum_{i=1}^{T}\widehat{\theta}_t\right)$. Now we use lemma B.3 to upper bound $\|\theta^* - \widehat{\theta}_t\|_{H_t(\widehat{\theta}_t)}$ with $CS^{1/2}\gamma_t(\delta)$ which we further upper bound by $CS^{1/2}\gamma_T(\delta)$ after noting that $\gamma_t(\delta) \le \gamma_{t+1}(\delta)$ for all $t \in [T]$. Thus, we now have,

$$R(T,x) \le \frac{CS^{1/2}\gamma_T(\delta)}{T}\sum_{t=1}^{T}\|\phi(x,a^*(x)) - \phi(x,\pi_T(x))\|_{H_t^{-1}(\widehat{\theta}_t)} \le \frac{CS^{1/2}\gamma_T(\delta)}{T}\sum_{t=1}^{T}\|\phi(x_t,a_t) - \phi(x_t,a'_t)\|_{H_t^{-1}(\widehat{\theta}_t)}\ .$$

To get the above inequality, we use the fact that algorithm's choice of the triplet is $(x_t,a_t,a'_t) := \operatorname{argmax}_{x\in\mathcal{X},a,a'\in\mathcal{A}}\|\phi(x,a) - \phi(x,a')\|_{H_t^{-1}(\widehat{\theta}_t)}$. Now, we are left with terms that can be bounded using Elliptic Potential Lemma (lemma D.2) after using the fact that $\|\phi(x_t,a_t) - \phi(x_t,a'_t)\|_{H_t^{-1}(\widehat{\theta}_t)} \le \sqrt{\kappa}\|\phi(x_t,a_t) - \phi(x_t,a'_t)\|_{V_t^{-1}}$ due to the fact that $V_t \preccurlyeq \kappa H_t(\widehat{\theta}_t)$. Thus,

$$\begin{aligned}
R(T,x) &\leq \frac{C\sqrt{\kappa S}\gamma_T(\delta)}{T}\sum_{t=1}^{T}\|\phi(x_t,a_t)-\phi(x_t,a_t')\|_{V_t^{-1}} \\
&\leq \frac{C\sqrt{\kappa S}\gamma_T(\delta)}{T}\sqrt{T\sum_{t=1}^{T}\|\phi(x_t,a_t)-\phi(x_t,a_t')\|^2_{V_t^{-1}}} && \text{(Cauchy-Schwarz)} \\
&\leq \frac{C\sqrt{\kappa S}\gamma_T(\delta)}{T}\sqrt{2dT\log\left(1+\frac{T}{\lambda\kappa d}\right)} && \text{(Lemma D.2)} \\
&= CS^{3/2}\sqrt{\left(d\log(\frac{ST}{d})+\log(\frac{T}{\delta})\right)\log\left(1+\frac{T}{\lambda\kappa d}\right)\frac{\kappa d}{T}}\,, && \text{(Def. of } \gamma_T(\delta))
\end{aligned}$$

□

C. Generalization to Function Approximation

In this section, we remove the assumption of BTL preference model characterized by a linear parameter θ. Instead, we assume that we have access to a function class

$$\mathcal{F}=\{f:\mathcal{X}\times\mathcal{A}\times\mathcal{A}\to[0,1]: f(x,a,a')+f(x,a',a)=1\},$$

where $f(x,a,a')$ denotes the probability that the arm a wins over arm a' given context x when the preference function is f, i.e., $f(x,a,a')=\mathbb{P}[a\succcurlyeq a'|x,f]$ where $a\succcurlyeq a'$ denotes the event that a wins over a'. Now, we assume that there is a true $f^*\in\mathcal{F}$ from which the data is generated. Further, we assume a *Condorcet* winner at each context:

Assumption C.1. For all context $x\in\mathcal{X}$, there is an action $a^*(x)\in\mathcal{A}(x)$ such that $f^*(x,a^*(x),a_0)\geq 1/2\;\forall a_0\in\mathcal{A}(x)$.

Note that in this case, there is no direct reward model and is therefore a generalization of the BTL model. The absence of a reward model makes the problem more nuanced. Accordingly, the simple regret is now defined as:

$$R(T)=\max_{x\in\mathcal{X}}\max_{a\in\mathcal{A}(x)} f^*(x,a,\pi_T(x))-1/2\;.$$

Note that $f^*(x,a^*(x),\pi_T(x))\geq 1/2$ by assumption C.1, thus $R(T)$ is always non-negative.

C.1. Algorithm

Our algorithm takes a function class $\mathcal{F}$ and a confidence level $\delta\in(0,1]$ as its inputs. First, a regularized least square estimate of f^* is computed by minimizing the cumulative squared prediction error:

$$\widehat{f}_t\in\operatorname{argmin}_{f\in\mathcal{F}}\sum\nolimits_{s=1}^{t-1}\left(y_s-f(x_s,a_s,a_s')\right)^2\;. \tag{9}$$

The confidence set $\mathcal{C}_t(\mathcal{F},\delta)$ is then defined as the set of all functions $f\in\mathcal{F}$ satisfying

$$\sum\nolimits_{s=1}^{t-1}(f(x_s,a_s,a_s')-\hat{f}_t(x_s,a_s,a_s'))^2\leq\beta_t(\mathcal{F},\delta)\;, \tag{10}$$

where $\beta_t(\mathcal{F},\delta)$ is an appropriately chosen confidence parameter. Since $y_t\sim\texttt{Ber}(f^*(x_t,a_t,a_t'))$ given (x_t,a_t,a_t'), We have Var$[y_t]\leq 1/4$. Thus, following (Ayoub et al., 2020), we set the confidence parameter

$$\beta_t(\mathcal{F},\delta)=2\log\frac{2\mathcal{N}(\mathcal{F})}{\delta}+2\sqrt{\log\frac{4t(t+1)}{\delta}}+4\;,$$

where $\mathcal{N}(\mathcal{F})$ denotes the $(1/t,||\cdot||_\infty)$-covering number[1] of $\mathcal{F}$. This choice of confidence width ensures that f^* lies in the confidence set $\mathcal{C}_t(\mathcal{F},\delta)$ at all time instant $t\geq 1$ with probability at least $1-\delta$ (Lemma C.4).

Next, for each triplet (x,a,a'), we define the exploration bonus $b_t(x,a,a')$ at round t as

$$b_t(x,a,a')=\max_{f_1,f_2\in\mathcal{C}_t(\mathcal{F},\delta)}|f_1(x,a,a')-f_2(x,a,a')|, \tag{11}$$

[1] For any $\alpha>0$, we call $\mathcal{F}^\alpha$ an $(\alpha,||\cdot||_\infty)$ cover of the function class $\mathcal{F}$ if for any $f\in\mathcal{F}$ there exists an f' in $\mathcal{F}^\alpha$ such that $||f'-f||_\infty:=\sup_{x\in\mathcal{X}}|f'(x)-f(x)|\leq\alpha$.

which measures the uncertainty of a pair of actions a, a' given a context x with respect to the confidence set $\mathcal{C}_t(\mathcal{F}, \delta)$. The near-optimal action set $\mathcal{A}_t(x)$ at round t is defined as the set of all actions in the previous set $\mathcal{A}_{t-1}(x)$ satisfying

$$\hat{f}_t(x, a, a_0) + b_t(x, a, a_0) \geq 1/2 \,\forall a_0 \in \mathcal{A}_{t-1}(x) \,. \tag{12}$$

Intuitively speaking, we retain only those actions from the previous near-optimal set which are not significantly outperformed by other actions according to the estimates of the current round. Since $f^* \in \mathcal{C}_t(\mathcal{F}, \delta)$, the optimal action $a^*(x)$ lies in $\mathcal{A}_t(x)$ for each context x for all t with high probability (Lemma C.5). By pruning out suboptimal actions every round, we make better use of samples. When the set $\mathcal{A}_t(x)$ becomes a singleton (i.e., $a^*(x)$ has been identified w.h.p), we remove this context from the pool of contexts considered in future rounds.

To encourage exploration, we choose actions $(a_t(x), a'_t(x))$ which has the highest uncertainty in $\mathcal{A}_t(x)$, i.e., we choose

$$(a_t(x), a'_t(x)) = \operatorname{argmax}_{a, a' \in \mathcal{A}_t(x)} b_t(x, a, a') \,. \tag{13}$$

Next, we choose the context x_t that provides the maximum information about the unknown preference function f^*, i.e.,

$$x_t \in \operatorname{argmax}_{x \in \mathcal{X}} b_t(x, a_t(x), a'_t(x)) \,. \tag{14}$$

We play the actions $a_t = a_t(x_t)$ and $a'_t = a'_t(x_t)$ in round t and observe the preference feedback y_t. We repeat this until we have exhausted the budget T. Our final policy π_T samples an action uniformly at random from the set $\mathcal{A}_T(x)$ for every context $x \in \mathcal{X}$. Pseudocode is given in Algorithm 3.

C.2. Result

We characterize the complexity of the function class $\mathcal{F}$ by its *eluder dimension* (Russo and Van Roy, 2013).

Definition C.2 (Eluder dimension)**.** The ε-eluder dimension $\dim_{\mathcal{E}}(\mathcal{F}, \varepsilon)$ of a function class $\mathcal{F}$ defined on a domain $\mathcal{X}$ is the length of the longest sequence $\{x_i\}_{i=1}^n \subseteq \mathcal{X}$ of input points such that for some $\varepsilon' \geq \varepsilon$ and for each $i \in \{2, \ldots, n\}$,

$$\sup_{f_1, f_2 \in \mathcal{F}} \left\{ (f_1 - f_2)(x_i) \,\middle|\, \sqrt{\sum_{j=1}^{i-1} (f_1 - f_2)^2(x_i)} \leq \varepsilon' \right\} > \varepsilon' \,.$$

We denote by $d_{\mathcal{E}}(\mathcal{F}) = \dim_{\mathcal{E}}(\mathcal{F}, 1/T)$, the $(1/T)$-Eluder dimension of the function class $\mathcal{F}$. Now, we state sub-optimality guarantee of the final policy using eluder dimension and metric entropy of the function class $\mathcal{F}$.

Theorem C.3 (Suboptimality Gap)**.** *Let $\delta \in (0, 1)$. Under assumption C.1, the suboptimality gap $R(T)$ of our policy π_T after running* `APO-Gen` *(algorithm 3) for T steps is upper bounded with probability at least $1 - \delta$ as*

$$R(T) \leq \tilde{O}\left(\sqrt{\frac{\log(\mathcal{N}(\mathcal{F})T/\delta) d_{\mathcal{E}}(\mathcal{F})}{T}} \right).$$

Proof is deferred to the next section. It essentially follows ideas similar to Theorem 3.4 with difference that we crucially leverage action elimination (Step 8).

BTL model. For the BTL preference model $f(x, a, a') = \mu(\phi(x, a)^\top \theta - \phi(x, a')^\top \theta)$. Define $r = \overline{h}/\underline{h}$, where

$$\overline{h} = \sup_{x, a, a', \theta} \dot{\mu}(\phi(x, a)^\top \theta - \phi(x, a')^\top \theta) \,,$$
$$\underline{h} = \inf_{x, a, a', \theta} \dot{\mu}(\phi(x, a)^\top \theta - \phi(x, a')^\top \theta) \,.$$

Then the $\log \mathcal{N}(\mathcal{F})$ and Eluder dimension of $\mathcal{F}$ are at most $O(d \log(\overline{h}T))$ and $O(dr^2 \overline{h} \log(rS\overline{h}T))$, respectively. Note that $\underline{h} = 1/\kappa$ and $\overline{h} \leq 1/4$. This yields $\log \mathcal{N}(\mathcal{F}) = O(d \log T)$ and $d_{\mathcal{E}}(\mathcal{F}) = O(\kappa^2 d \log T)$. Substituting this in Theorem C.3, we get the sub-optimality gap $O(\kappa d/\sqrt{T})$, which is $\sqrt{\kappa}$ factor loose than Theorem 3.4. This is because we crucially use self-concordance of the sigmoid function in Theorem 3.4 to shave this extra $\sqrt{\kappa}$ factor. Nevertheless, Theorem C.3 is general enough to subsume other preference models (e.g. probit/Thurstone) beyond the BTL model.

C.3. Analysis

First we present a result that characterizes the confidence set around $\widehat{f}_t$.

Lemma C.4 (Confidence Set for Function Approximation (Lemma A.1 of (Chen et al., 2022))**.** *Let $\delta \in (0, 1)$. Define the*

Algorithm 3 `APO-Gen`: Active Preference Optimization with General Function Approximation

```
Require: Context set $\mathcal{X}$, action set $\mathcal{A} = [K]$, function class $\mathcal{F}$, failure level $\delta \in (0, 1)$
Set $\mathcal{X}_0 = \mathcal{X}$ and $\mathcal{A}_0(x) = \mathcal{A} \, \forall x \in \mathcal{X}$,
for $t = 1, 2, \ldots T$ do
    Compute function estimate $\widehat{f}_t$ usning (9).
    Construct confidence set $\mathcal{C}_t(\mathcal{F}, \delta)$ using (10).
    Intialize the $\mathcal{X}_t = \mathcal{X}_{t-1}$
    for each context $x \in \mathcal{X}_{t-1}$ do
        For each pair of actions $a, a' \in \mathcal{A}_{t-1}(x)$, compute the bonus $b_t(x, a, a')$ using (11).
        Find the near-optimal action set $\mathcal{A}_t(x)$ using (12)
        if $|\mathcal{A}_t(x)| = 1$ then
            Set $\mathcal{A}_T(x) = \mathcal{A}_t(x)$
            $\mathcal{X}_t \leftarrow \mathcal{X}_t \setminus \{x\}$
    Choose context and pair of actions $(x_t, a_t, a'_t) = \text{argmax}_{x \in \mathcal{X}_t, a, a' \in \mathcal{A}_t(x)} b_t(x, a, a')$
    Observe preference $y_t \sim \texttt{Ber}(f^*(x_t, a_t, a'_t))$
Output final policy $\pi_T(x) = a$ for some $a \in \mathcal{A}_T(x)$.
```

confidence set

$$\mathcal{C}_t(\mathcal{F}, \delta) = \{f \in \mathcal{F} | \sum_{s=1}^{t-1} (f(x_s, a_s, a'_s) - \widehat{f}_t(x_s, a_s, a'_s))^2 \leq \beta_t(\mathcal{F}, \delta)$$

Let $\mathcal{E}_t(\delta)$ be the event that $f^* \in \mathcal{C}_t(\mathcal{F}, \delta)$. Then, $\mathbb{P}[\mathcal{E}_t(\delta)] \geq 1 - \delta$. Further, $\mathbb{P}\left[\cap_{t=1}^{T} \mathcal{E}_t(\delta/T)\right] \geq 1 - \delta$.

Proof. The proof is a direct extension of lemma D.3 by observing that in our case the subgaussianity parameter $\sigma = 1/4$ since our rewards are Bernoulli and by setting $\alpha = 1/t$. Moreover, $C = 1$ in our case. Finally, since $\mathcal{E}_t(\delta/t)$ holds with probability at least $1 - \delta/t$, by union bound we can show that $\mathbb{P}\left[\cap_{t=1}^{T} \mathcal{E}_t(\delta/T)\right] = 1 - \mathbb{P}\left[\cup_{t=1}^{T} \overline{\mathcal{E}}_t(\delta/T)\right] \geq 1 - \sum_{t=1}^{T} \mathbb{P}[\overline{\mathcal{E}}_t(\delta/T)] \geq 1 - \delta$. □

Hereon, we will assume that $\mathcal{E}_t(\delta/T)$for all $t \in [T]$. All subsequent guarantees will be proved this event. The next result shows that for each context x, the optimal action $a^*(x)$ lies in $\mathcal{A}_t(x)$ for all t.

Lemma C.5. *For a given context $x \in \mathcal{X}$, let $\{\mathcal{A}_s(x)\}_{s=0}^{t}$ be defined as follows: (a) $\mathcal{A}_0(x) = \mathcal{A}$ and (b) $\mathcal{A}_s(x) = \{a \in \mathcal{A}_{s-1}(x) | \widehat{f}_s(x, a, a') + b_s(x, a, a') \geq \frac{1}{2} \, \forall a' \in \mathcal{A}_{s-1}(x)\}$. Then, we have, $a^*(x) \in \mathcal{A}_s(x)$ for all $s \in [t]$.*

Proof. The proof is by induction. First note that by definition of $a^*(x)$, $f^*(x, a^*(x), a') \geq 1/2$ for every $a' \in \mathcal{A}$, and $a^*(x) \in \mathcal{A}_0(x) = \mathcal{A}$. Suppose, for some $s > 0$, $a^*(x) \in \mathcal{A}_{s-1}(x)$. Now, we know that under event $\mathcal{E}_s(\delta/T)$, $f^* \in \mathcal{C}_s(\mathcal{F}, \delta/T)$ and thus from definition of $b_s(x, a, a')$, $f^*(x, a, a') - \widehat{f}_s(x, a, a') \leq b_s(x, a, a')$. Thus, for any $a' \in \mathcal{A}_{s-1}(x)$,

$$\frac{1}{2} \leq f^*(x, a^*(x), a') \leq \widehat{f}_s(x, a^*(x), a') + b_s(x, a^*(x), a')$$

Hence $a^*(x) \in \mathcal{A}_s(x)$. Thus by induction, $a^*(x) \in \mathcal{A}_s(x)$ for all $s \in [t]$. □

Now, we are ready to prove Theorem C.3.

Proof. The idea is to show that our arm elimination technique throws away arms with large suboptimality gap in every round for every context. Thus the set $\mathcal{A}_t(x)$ maintains a candidate set of good arms at every time instant. In the end, playing any action from $\mathcal{A}_T(x)$ ensures that we only play actions from a set of actions that are only $1/\sqrt{T}$ suboptimal. Formally, for any context $x \in \mathcal{X}$, the suboptimality $R(T,x)$ is upper bounded as follows:

$$\begin{aligned}
R(T,x) &= f^*(x,a^*(x),\pi_T(x)) - \frac{1}{2} \\
&\leq \frac{1}{T}\sum_{t=1}^{T}\left[\widehat{f}_t(x,a^*(x),\pi_T(x)) + b_t(x,a^*(x),\pi_T(x)) - \frac{1}{2}\right] && (a^*(x),\pi_T(x) \in \mathcal{A}_t(x)\ \forall\ t \in [T]) \\
&= \frac{1}{T}\sum_{t=1}^{T}\left[\frac{1}{2} - \widehat{f}_t(x,\pi_T(x),a^*(x)) + b_t(x,a^*(x),\pi_T(x))\right] && (f(x,a,a') + f(x,a',a) = 1\ \forall\ f \in \mathcal{F}) \\
&= \frac{1}{T}\sum_{t=1}^{T}\left[\frac{1}{2} - \widehat{f}_t(x,\pi_T(x),a^*(x)) + b_t(x,\pi_T(x),a^*(x))\right] && (b_t(x,a,a') = b_t(x,a',a)) \\
&\leq \frac{1}{T}\sum_{t=1}^{T}\left[b_t(x,\pi_T(x),a^*(x)) + b_t(x,\pi_T(x),a^*(x))\right] && (\text{Since } \pi_T(x),a^*(x) \in \mathcal{A}_t(x), \text{line 7 Algorithm 3}) \\
&= \frac{2}{T}\sum_{t=1}^{T} b_t(x,\pi_T(x),a^*(x)) \\
&\leq \frac{2}{T}\sum_{t=1}^{T} b_t(x_t,a_t,a'_t) && (\text{Line 9 of Algorithm 3})
\end{aligned}$$

Now we invoke lemma D.4 to bound the RHS.

$$R(T,x) \leq \frac{2}{T}\sum_{t=1}^{T} b_t(x_t,a_t,a'_t) \leq \frac{2}{T}\left[\frac{1}{T} + \min\{d_{\mathcal{E}}(\mathcal{F}),T\} + 2\beta_T(\mathcal{F},\delta/T)\sqrt{d_{\mathcal{E}}(\mathcal{F})T}\right]$$

Simplifying constants and using the fact that $\min\{a,b\} \leq \sqrt{ab}$ for $a,b > 0$, we get $R(T,x) \leq C\beta_T(\mathcal{F},\delta/T)\sqrt{\frac{d_{\mathcal{E}}(\mathcal{F})}{T}}$. Now, using order notation, we have for all $x \in \mathcal{X}$ with probability at least $1-\delta$,

$$R(T,x) \leq \tilde{O}\left(\sqrt{\frac{\log(\mathcal{N}(\mathcal{F})T/\delta)d_{\mathcal{E}}(\mathcal{F})}{T}}\right),$$

which completes the proof. □

D. Some Useful Results

Lemma D.1. *Let $z, z' \in \mathbb{R}$ and $\tilde{\alpha}(z,z') := \int_0^1 (1-v)\dot{\sigma}(z+v(z'-z))dv$. Then, for some $C > 1$ (1.01 suffices),*

$$\tilde{\alpha}(z,z') \geq \frac{\dot{\sigma}(z')}{C(2+|z-z'|)^2}$$

Proof. Firstly, note that by property of definite integrals $\int_a^b f(x)dx = \int_a^b f(a+b-x)dx$, we have

$$\int_0^1 (1-v)\dot{\sigma}(z+v(z'-z))dv = \int_0^1 v\dot{\sigma}(z'+v(z-z'))dv$$

Now, we use the fact that $\dot{\sigma}(x) \geq \dot{\sigma}(y)\exp(-|x-y|)$ (see appendix A of (Faury et al., 2022)). Let $a = |z-z'|$. Thus,

$$\begin{aligned}
\int_0^1 v\dot{\sigma}(z'+v(z-z'))dv \geq \int_0^1 v\dot{\sigma}(z')\exp(-va)dv &= \dot{\sigma}(z')\int_0^1 v\exp(-va)dv \\
&= \dot{\sigma}(z')\left(\frac{1-(1+a)e^{-a}}{a^2}\right) \\
&\geq \dot{\sigma}(z')\left(\frac{1-1/a}{a^2}\right) && ((1+a)e^{-a} < \tfrac{1}{a}\ \forall a > 0) \\
&= \dot{\sigma}(z')\left(\frac{a-1}{a^3}\right)
\end{aligned}$$

Again, from appendix A of (Faury et al., 2022), we have that $\tilde{\alpha}(z, z') \geq \dot{\sigma}(z)/(2+a)$ which can again be lower bounded with $\dot{\sigma}(z')e^{-a}/(2+a)$. Combining this lower bound with above, we get,

$$\tilde{\alpha}(z, z') \geq \max\left\{\frac{e^{-a}}{2+a}, \frac{a-1}{a^3}\right\}\dot{\sigma}(z')$$

Finally, we can lower bound the RHS with $\frac{\dot{\sigma}(z')}{C(2+a)^2}$ for some $C > 1.01$. To do this, let $f(x) = (2+x)e^{-x}$. Thus, $f'(x) = -(1+x)e^{-x}$ which implies that $f(x)$ is decreasing for $x > 0$. Thus, $f(x) = \frac{1}{C}$ is satisfied for only one value of x since $f(0) = 2 > 1/C$. For $C = 1.01$, this value is $x_0 = 1.1608$. Then, for $0 \leq x \leq x_0$, $e^{-x}/(2+x) \geq 1/C(2+x)^2$. Again, let $g(x) = (x-1)(x+2)^2/x^3$. Simplifying, we have, $g(x) = 1 + \frac{3}{x} - \frac{4}{x^3}$. It is easy to see that for $x \geq 2/\sqrt{3}$, $\frac{3}{x} \geq \frac{4}{x^3}$ which implies that $g(x) \geq 1$ for all $x \geq x_1 = 2/\sqrt{3} = 1.1547$. So, for $x \geq 1.1547$, $g(x) \geq 1/C$ (since $C > 1$) which is equivalent to $\frac{(x-1)}{x^3} \geq \frac{1}{C(x+2)^2}$. The numeric solution to $g(x) = 1/C$ for $C = 1.01$ is $x_2 = 1.1525$. It can be checked via first derivative test that $g(x)$ is increasing in $x_2 \leq x \leq x_1$. Thus, indeed, $g(x) \geq 1/C$ for all $x \geq x_2$. Jence, we have established so far that for $C = 1.01$,

$$\frac{x-1}{x^3} \geq \frac{1}{C(x+2)^2} \qquad \forall\, x \geq x_2 = 1.1525$$
$$\frac{e^{-x}}{2+x} \geq \frac{1}{C(x+2)^2} \qquad \forall\, x \leq x_0 = 1.1608$$

Since, $x_2 \leq x_0$, we have the required result that $\max\left\{\frac{e^{-a}}{2+a}, \frac{a-1}{a^3}\right\}\dot{\sigma}(z') \geq \frac{\dot{\sigma}(z')}{C(2+a)^2}$ for all $a \geq 0$ which completes the proof. □

Lemma D.2 (Elliptic Potential Lemma). *Let $\{z_s\}_{s=1}^t$ be a sequence of vectors in $\mathbb{R}^d$ such that $\|z_s\| \leq L$ for any $s \in [t]$. Let $V_t = \sum_{s=1}^{t-1} z_s z_s^\intercal + \lambda I$. Then,*

$$\sum_{s=1}^{t} \|z_s\|_{V_s^{-1}}^2 \leq 2d \log\left(1 + \frac{tL^2}{\lambda d}\right)$$

.

Now we present the confidence set properties of function approximation. We use the same notations as (Ayoub et al., 2020)

Let $(X_p, Y_p)_{p\geq 1}$ be a sequence of random elements, $X_p \in \mathcal{X}$ for some measurable set $\mathcal{X}$ and $Y_p \in \mathbb{R}$. Let $\mathcal{F}$ be a subset of the set of real-valued measurable functions with domain $\mathcal{X}$. Let $\mathbb{F} = (\mathbb{F}_p)_{p\geq 0}$ be a filtration such that for all $p \geq 1, (X_1, Y_1, \ldots, X_{p-1}, Y_{p-1}, X_p)$ is $\mathbb{F}_{p-1}$ measurable and such that there exists some function $f^* \in \mathcal{F}$ such that $\mathbb{E}[Y_p \mid \mathbb{F}_{p-1}] = f^*(X_p)$ holds for all $p \geq 1$. The (nonlinear) least-squares predictor given $(X_1, Y_1, \ldots, X_t, Y_t)$ is $\widehat{f}_t = \operatorname{argmin}_{f\in\mathcal{F}} \sum_{p=1}^t (f(X_p) - Y_p)^2$. We say that Z is conditionally ρ-subgaussian given the σ-algebra $\mathbb{F}$ if for all $\lambda \in \mathbb{R}$, $\log \mathbb{E}[\exp(\lambda Z) \mid \mathbb{F}] \geq \frac{1}{2}\lambda^2\rho^2$. For $\alpha > 0$, let N_α be the $\|\cdot\|_\infty$-covering number of $\mathcal{F}$ at scale α. That is, N_α is the smallest integer for which there exist $\mathcal{G} \subset \mathcal{F}$ with N_α elements such that for any $f \in \mathcal{F}$, $\min_{g\in\mathcal{G}} \|f - g\|_\infty \leq \alpha$. For $\beta > 0$, define $\mathcal{F}_t(\beta) = \{f \in \mathcal{F} : \sum_{s=1}^t (f(X_s) - \widehat{f}_t(X_p))^2 \leq \beta\}$.

Lemma D.3 (Theorem 5 of (Ayoub et al., 2020)). *Let $\mathbb{F}$ be the filtration defined above and assume that the functions in $\mathcal{F}$ are bounded by the positive constant $C > 0$. Assume that for each $s \geq 1$, $(Y_p - f^*(X_p))_p$ is conditionally σ-subgaussian given $\mathbb{F}_{p-1}$. Then, for any $\alpha > 0$, with probability $1 - \delta$, for all $t \geq 1$, $f^* \in \mathcal{F}_t(\beta_t(\delta, \alpha))$, where*

$$\beta_t(\delta, \alpha) = 8\sigma^2 \log(2N_\alpha/\delta) + 4t\alpha\left(C + \sqrt{\sigma^2 \log(4t(t+1)/\delta)}\right).$$

Lemma D.4 (Lemma 2 of (Russo and Van Roy, 2013)). *Let $\mathcal{F} \in B_\infty(\mathcal{X}, C)$ be a set of functions bounded by $C > 0$, $(\mathcal{F}_t)_{t\geq 1}$ and $(x_t)_{t\geq 1}$ be sequences such that $\mathcal{F}_t \subset \mathcal{F}$ and $x_t \in \mathcal{X}$ hold for $t \geq 1$. Let $\mathcal{F}\mid_{x_{1:t}} = \{(f(x_1), \ldots, f(x_t)) : f \in \mathcal{F}\} (\subset \mathbb{R}^t)$ and for $S \subset R^t$, let $diam(S) = \sup_{u,v\in S} \|u - v\|_2$ be the diameter of S. Then, for any $T \geq 1$ and $\alpha > 0$ it holds that*

$$\sum_{t=1}^{T} diam(\mathcal{F}_t \mid_{x_t}) \leq \alpha + C(d \wedge T) + 2\delta_T\sqrt{dT}$$

where $\delta_T = \max_{1\leq t\leq T} diam(\mathcal{F}_t \mid_{x_t})$ and $d = dim_{\mathcal{E}}(\mathcal{F}, \alpha)$ is the Eluder Dimension of $\mathcal{F}$.

E. Experiment Details

E.1. Results on Controlled Sentiment Generation Task

In our experiment on controlled sentiment generation, we consider a user group that prefers positive sentiment completions for a prompt. Using the IMDb dataset as a basis for our inputs (Maas et al., 2011), the goal for the optimal policy is to produce responses y that exhibit positive sentiment, catering to the user group's preferences for a given prompt x. For a controlled evaluation, we generated a set of preference pairs utilizing a pre-trained sentiment classifier where $p(\text{positive} \mid x, y_w) > p(\text{positive} \mid x, y_l)$ for the evaluation.

We implement the 3 phases of RLHF pipeline (Christiano et al., 2017; Ouyang et al., 2022): (i) Supervised Fine-tuning, (ii) Reward Modelling, and (iii) RL Fine-tuning. For the SFT policy, we fine-tune GPT-2 (Radford et al., 2019) until convergence on reviews from the (80-20) train split of the IMDb dataset with 8000 samples and use this GPT-2 backbone for both the reward model and PPO training (Schulman et al., 2017). For the reward learning, we use a total size of 5000 preference and adaptively select samples and evaluate the performance on the 2000 validation set as shown in Figure 1. The generations are evaluated against the ground truth reward r^* for positive sentiment, provided by the pre-trained sentiment classifier (similar to (Rafailov et al., 2023)). Hyperparameters for our experiments are given in Appendix F.

To demonstrate the performance of Algorithm 2 against random selection baselines, we use the feature representation $\phi(x, y)$ given a prompt x and response y using the GPT-2 SFT backbone. We estimate the uncertainty $b_t(x, y_{\texttt{chosen}}, y_{\texttt{rejected}})$ for each $(x, y_{\texttt{chosen}}, y_{\texttt{rejected}})$ in our dataset $\mathcal{D}$ (step 3 of Algorithm 2) and select the top-B samples (step 6) to update the reward model. We repeat this process K times and compare the same against random baseline (where we select the B samples randomly) for different values of BK. Note that our total sample budget is now $T = BK$. Finally, we train PPO (Schulman et al., 2017) with the learned reward model and evaluate the responses against the ground truth reward r^* for positive sentiment.

Figure 1 shows the results of the experiment. It is clear that evaluation accuracy of the reward model learned by APO is much higher than the one learned via random selection baseline even when APO's sample budget is only 5% of the data and random baseline's 40% illustrating the suboptimality gap as shown in Theorem 3.2. Next, we compare the performance of the aligned models trained via PPO using the respective reward models. For APO, we use the reward model trained on a sample budget of 10% while for random baseline it is the highest accuracy reward model (corresponding to 40% sample). From Figure 1 it is evident that APO outperforms the random baseline by a 60 : 40 win rate demonstrating the efficiency of the proposed method. Hyperparameters are given in Appendix F.

E.2. Results on Single-Turn Dialogue Task

In this experiment, we use Anthropic helpful and harmless preference dataset (Bai et al., 2022), and gemma-2b[2] (Team et al., 2024) language model. We first collect all the prompts with single-turn dialogues. Then we split this collection into two in $80 : 20$ ratio. On the larger 80%-collection, we make inference with a Mistral-7b reward model[3], which serves as the latent reward model r^*. Then we obtain reward differences between chosen and rejected responses in this collection. Based on these, we separate the dataset into three buckets: (i) reward difference between -1 to 1 ($\sim$ 7500 samples), (ii) reward difference between 1 to 3 ($\sim$ 6500 samples), and (iii) reward difference more than 3 ($\sim$ 3300 samples). These buckets contain data points which are progressively easier to classify. With this, we create a train set of $8,000$ samples containing 4500 from (i), 2500 from (ii) and 1000 samples from (iii). Such a training dataset highlights the importance of selecting prompts more carefully to obtain useful information during learning. Randomly sampling prompts to collect feedback is more likely to hurt the performance in such a setting. For the test set, we sample 2000 data points from the smaller 20%-collection set aside previously.

Reward Evaluation. We compare the reward model learnt by our algorithm APO with that of random sampling baseline. After the reward models are trained, we compute rewards for *chosen* and *rejected* responses in the test dataset. We define reward evaluation accuracy as the $\%$ of the test samples for which the reward of *chosen* response is higher than that of *rejected* response. We plot reward accuracy as a function of training samples for small (5) and large (20) no. of batches keeping sample budget same. The results are shown in Fig. 1. We observe that APO outperforms the random-sampling baseline. We also see that the reward accuracy increases with increasing number of epochs for a given sample budget. We

[2] Specifically the instruction-tuned version: https://huggingface.co/google/gemma-2b-it

[3] https://huggingface.co/Ray2333/reward-model-Mistral-7B-instruct-Unified-Feedback

only vary no. of samples till 4000 as we want to compare performances under budget constraint.

Win Rate. We compare the win rate of the APO-policy with that of the random-sampling baseline. We train the reward models on 2000 samples over 20 epochs, and then the final policies are trained using PPO. We then generate responses from these policies for all the 2000 prompts in test set (except 21 prompts of length greater than 250 tokens, due to compute bottleneck), and obtain a reward score for each generation using the Mistral-7b reward model. Win rate of APO over random baseline is computed as the percentage of samples for which reward of APO-policy generation is higher than that of the random-sampling based policy. From Fig. 1, we observe APO outperforms random baseline by a $60 : 40$ win rate. Thus we observe significant improvement in win rate by using APO over random sampling baseline. Hyperparameters are given in Appendix F.

F. Hyperparameter Details

Any hyperparameters not explicitly mentioned use the default values in the TRL library.[4]

F.1. Experiments on Controlled Sentiment Generation

The hyperparameters for the experiments are outlined in Table 1.

Table 1. Hyperparameters used in our experiment

Parameter	Value
regularizer in APO	1e-5
beta	0.1
learning rate	1.41e-5
batch size	16
max length	512
max prompt length	128

F.2. Experiment with Anthropic Dataset

All experiments in this section were performed using one A100 (80 GB) GPU. For reward learning for APO, we use regularizer $\lambda = 1 \times 10^{-5}$. The learning rate for both APO and random is set to 1×10^{-2} with weight decay of 1×10^{-5}. After every epoch of data collection, the training step on the logistic loss is run for 10 epochs.

The details for the PPO configuration is same for both APO and random. The details are given in Table 2.

Table 2. Hyperparameters used in PPO Training

Parameter	Value
learning rate	1e-4
lora-rank	8
lora_alpha	32
lora_dropout	0.05
batch size	16
mini batch size	8
max new tokens	256
top_k	80
top_p	1
temperature	1.1
do_sample	True

[4] huggingface.co/docs/trl/index

F.3. Additional Experimental Comparison with AE-DPO ((Mehta et al., 2023))

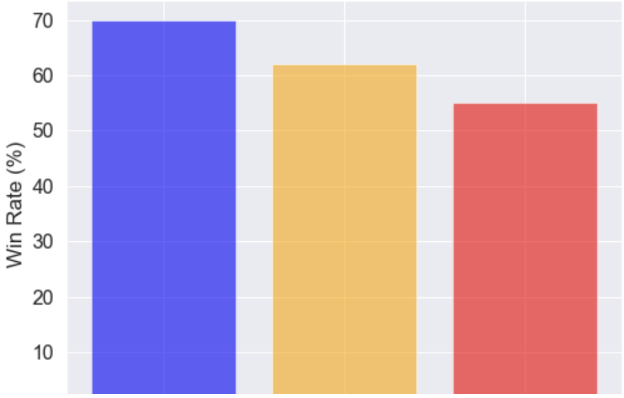

Figure 3. Win rates of APO (left), AE-DPO ((Mehta et al., 2023)) (middle) and random baselines (right) compared against SFT model on the IMDb controlled generation task. We can see that APO outperforms all the other methods.