POLICY OPTIMIZATION IN RLHF: THE IMPACT OF OUT-OF-PREFERENCE DATA

Ziniu Li*
The Chinese University of Hong Kong, Shenzhen
Shenzhen Research Institute of Big Data

Tian Xu* **& Yang Yu**†
National Key Laboratory for Novel Software Technology, Nanjing University
School of Artificial Intelligence, Nanjing University
Polixir.ai

ABSTRACT

Aligning agents with human preferences is important. This paper examines two classes of alignment methods. The first class operates without explicitly learning a reward model from preference data, with Direct Preference Optimization (Rafailov et al., 2023) emerging as a prominent method within this class. The second class involves methods that explicitly learn a reward function and utilize it to optimize policy on prompts-only data, with Proximal Policy Optimization (Schulman et al., 2017) standing out as a popular choice. Within this class, we investigate a notable approach that leverages a large amount of prompts and responses, extending beyond those present in the preference dataset. Experiments demonstrate that this approach outperforms other methods on synthetic contextual bandits tasks. Additionally, we provide an analysis of policy optimization errors for these methods and draw connections with other related research areas, such as imitation learning and reinforcement learning. In essence, our research highlights the importance of integrating out-of-preference data into the policy optimization. The code is available at `https://github.com/liziniu/policy_optimization`.

1 INTRODUCTION

Developing trustworthy agents requires alignment with human preferences (Russell & Norvig, 2010). A standard practice involves providing a human preference dataset for the agent to learn from. According to utility theory (Fishburn et al., 1979), preference is connected with a certain reward function. Currently, there are two kinds of alignment methods:

- The first class of methods, referred to as the reward-model-free policy optimization (**RMF-PO**) in this paper, does not explicitly learn a reward model but directly optimizes the language model from preference annotations. This class of methods includes popular algorithms such as Direct Preference Optimization (DPO) (Rafailov et al., 2023) and Identity Policy Optimization (IPO) (Azar et al., 2023).
- The second class of methods, referred to as reward-model-based policy optimization (**RMB-PO**) in this paper, is exemplified by the Reinforcement Learning from Human Feedback (RLHF) framework (Christiano et al., 2017; Stiennon et al., 2020; OpenAI, 2023), where RL methods (such as PPO (Schulman et al., 2017) or ReMax (Li et al., 2023)) are employed. Specifically, this class of methods trains a reward model from preference data and then optimizes the language model to improve its responses to prompts. Notably, it exploits out-of-preference data, that is, prompts and responses beyond those in the preference dataset.

While both approaches are able to improve performance by leveraging preference data, the superiority of one method over the other remains an open question, crucial for driving future advancements.

*Equal contribution. Author ordering is determined by coin flip. Emails: `ziniuli@link.cuhk.edu.cn` and `xut@lamda.nju.edu.cn`

†Corresponding author. Email: `yuy@nju.edu.cn`

2 MAIN RESULTS

We analyze the errors in the language model's optimization within the framework of contextual bandits (Lattimore & Szepesvári, 2020). In this framework, the language model is viewed as a *policy*, with prompts and responses corresponding to *states* and *actions*, respectively.

Proposition 1. *Define the reward evaluation error* $\varepsilon_r = \sup_{(s,a)} |\widehat{r}(s,a) - r(s,a)|$ *of a learned reward function* $\widehat{r}$*, the state distribution estimation error* $\varepsilon_s = \sup_\pi \sup_{r:\mathcal{S}\times\mathcal{A}\to[0,1]} \mathbb{E}_s\mathbb{E}_{a\sim\pi(\cdot|s)}[r(s,a)] - \widehat{\mathbb{E}}_s\mathbb{E}_{a\sim\pi(\cdot|s)}[r(s,a)]$*, and the action distribution estimation error* $\varepsilon_a = \sup_\pi \sup_{r:\mathcal{S}\times\mathcal{A}\to[0,1]} \sup_s \mathbb{E}_{a\sim\pi(\cdot|s)}[r(s,a)] - \widehat{\mathbb{E}}_{a\sim\pi(\cdot|s)}[r(s,a)]$*. Here* $\widehat{\mathbb{E}}_s$ *and* $\widehat{\mathbb{E}}_a$ *denote the finite-sample estimations of expectation under the state and action distributions, respectively. Consider* $\beta = 0$ *(i.e., no KL regularization), then we have*

$$r(\pi_r^\star) - r(\widehat{\pi}_{\text{RMB-PO}}) \le 2\varepsilon_r + 2\varepsilon_s,$$
$$r(\pi_r^\star) - r(\widehat{\pi}_{\text{RMF-PO}}) \le 2\varepsilon_r + 2\varepsilon_s + 2\varepsilon_a,$$

where $r(\pi) = \mathbb{E}_s\mathbb{E}_{a\sim\pi(\cdot|s)}[r(s,a)]$ *is the evaluation performance of a policy* π*.*

The details of the RMB-PO and RMF-PO approaches are provided in Appendix A, and the proof is given in Appendix B. The theory highlights that RMB-PO methods can reduce the action distribution estimation error and may generalize well (refer to the illustration in Figure 3 in the Appendix).

We provide empirical evidence for the above theory. We study a linear bandit task, where we have $r(s,a) = \phi_r(s,a)^\top \theta_r^\star$, with $\phi_r(s,a) \in \mathbb{R}^d$ denoting the feature representation and $\theta_r^\star \in \mathbb{R}^d$ as the parameter to learn. We provide the feature map ϕ_r to the reward model. For the policy, we consider $\pi(a|s) = \exp(\phi_\pi(s,a)^\top\theta_\pi)/\sum_{a'}\exp(\phi_\pi(s,a')^\top\theta_\pi)$. We examine two scenarios: $\phi_\pi = \phi_r$ (indicating no feature mismatch between the reward and policy models) and the converse case $\phi_\pi \neq \phi_r$. The latter case corresponds to practical scenarios in which the policy model and the reward model have different feature representations (due to architectures or training).

We display the optimality gap $|r(\pi^\star) - r(\widehat{\pi})|$ (the smaller, the better) in Figure 1 and Figure 2. A variant of RMB-PO, called RMB-PO+, which leverages states beyond those in the preference dataset to further reduce state distribution estimation error as in Proposition 1, is also examined. In fact, RMB-PO+ is much closer to the setting described in (Ouyang et al., 2022).

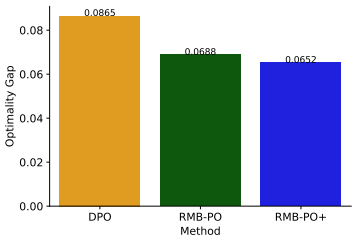


Figure 1: Optimality gap with $\phi_r = \phi_\pi$.

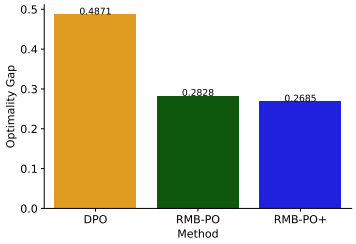


Figure 2: Optimality gap with $\phi_r \neq \phi_\pi$.

We find that even though the policy model is provided with a good feature (e.g., in Figure 1), RMB-PO methods can benefit from policy-generated data. This conclusion is also true for neural bandit tasks (results are given in Appendix C.2). Thus, we believe it is crucial to exploit the out-of-preference data, even when the two models share the same good feature. Otherwise, the policy might conflict with the reward model and perform poorly in the out-of-distribution regime. For a better understanding, see the visualization of learned policy distribution in Figure 6 in the Appendix.

Our study relates to imitation learning (Osa et al., 2018), focusing on learning policies from expert demonstrations. Within this field, there are two primary methods: behavioral cloning (BC) (Pomerleau, 1991) and adversarial imitation learning (AIL) (Ho & Ermon, 2016). Typically, BC is considered less effective than AIL, which enhances performance by incorporating "out-of-expert-data" through sampling new trajectories. This principle aligns with RMB-PO approaches and effectively addresses distribution shifts, an area where AIL is theoretically proven to excel (Xu et al., 2020; 2022). Further discussion is applicable to transition-model-based reinforcement learning (see Appendix D).

In summary, our results underscore the importance of optimizing policies on out-of-preference data and the power of using a reward model to provide supervision signals.

URM STATEMENT

The authors acknowledge that at least one key author of this work meets the URM criteria of ICLR 2024 Tiny Papers Track. For example, the author Ziniu Li is outside the age range of 30-50 years.

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.

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.

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*, pp. 4299–4307, 2017.

Steven Diamond and Stephen Boyd. Cvxpy: A python-embedded modeling language for convex optimization. *The Journal of Machine Learning Research*, 17(1):2909–2913, 2016.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

Peter C Fishburn, Peter C Fishburn, et al. *Utility theory for decision making*. Krieger NY, 1979.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 10835–10866, 2023.

Seyed Kamyar Seyed Ghasemipour, Richard S. Zemel, and Shixiang Gu. A divergence minimization perspective on imitation learning methods. In *Proceedings of the 3rd Conference on Robot Learning*, pp. 1259–1277, 2019.

Danijar Hafner, Timothy P. Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *Proceedings of the 8th International Conference on Learning Representations*, 2020.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems 29*, pp. 4565–4573, 2016.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in neural information processing systems 32*, pp. 12498–12509, 2019.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, 2015.

John Langford and Tong Zhang. The epoch-greedy algorithm for multi-armed bandits with side information. *Advances in Neural Information Processing Systems 20*, 2007.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

Ziniu Li, Tian Xu, Yushun Zhang, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. Remax: A simple, effective, and efficient method for aligning large language models. *arXiv preprint arXiv:2310.10505*, 2023.

Tyler Lu, Dávid Pál, and Martin Pál. Contextual multi-armed bandits. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pp. 485–492, 2010.

Fan-Ming Luo, Tian Xu, Xingchen Cao, and Yang Yu. Reward-consistent dynamics models are strongly generalizable for offline reinforcement learning. *arXiv preprint arXiv:2310.05422*, 2023.

Yuping Luo, Huazhe Xu, Yuanzhi Li, Yuandong Tian, Trevor Darrell, and Tengyu Ma. Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees. In *Proceedings of the 7th International Conference on Learning Representations*, 2019.

OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J. Andrew Bagnell, Pieter Abbeel, and Jan Peters. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotic*, 7(1-2): 1–179, 2018.

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*, pp. 27730–27744, 2022.

Dean Pomerleau. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation*, 3(1):88–97, 1991.

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.

Stuart J Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. London, 2010.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *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.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Nino Vieillard, Tadashi Kozuno, Bruno Scherrer, Olivier Pietquin, Rémi Munos, and Matthieu Geist. Leverage the average: an analysis of kl regularization in reinforcement learning. *Advances in Neural Information Processing Systems*, 33:12163–12174, 2020.

Tian Xu, Ziniu Li, and Yang Yu. Error bounds of imitating policies and environments. In *Advances in Neural Information Processing Systems 33*, pp. 15737–15749, 2020.

Tian Xu, Ziniu Li, Yang Yu, and Zhi-Quan Luo. Understanding adversarial imitation learning in small sample regime: A stage-coupled analysis. *arXiv preprint arXiv:2208.01899*, 2022.

Zhewei Yao, Reza Yazdani Aminabadi, Olatunji Ruwase, Samyam Rajbhandari, Xiaoxia Wu, Ammar Ahmad Awan, Jeff Rasley, Minjia Zhang, Conglong Li, Connor Holmes, et al. Deepspeed-chat: Easy, fast and affordable rlhf training of chatgpt-like models at all scales. *arXiv preprint arXiv:2308.01320*, 2023.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems 33*, pp. 14129–14142, 2020.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric Xing, et al. Lmsys-chat-1m: A large-scale real-world llm conversation dataset. *arXiv preprint arXiv:2309.11998*, 2023.

A PROBLEM FORMULATION

We consider the so-called contextual bandits (Langford & Zhang, 2007; Lu et al., 2010) formulation, which serves mathematical models for alignment. Let s and a be the state and action, respectively. We aim to obtain a decision policy π that acts optimally in terms of reward maximization:

$$\pi_r^\star \leftarrow \operatorname*{argmax}_\pi \mathbb{E}_{s\sim\rho(\cdot)}\mathbb{E}_{a\sim\pi(\cdot|s)}\left[r(s,a)\right], \tag{1}$$

where the symbol ρ denotes the state distribution, and r is the ground truth reward function. We omit the subscript r in $\pi_r^\star$ when the context is clear. For language models, the term "states" refers to prompts, while "actions" denote responses. The language model functions as the decision-making policy. It is worth noting that terminologies may be used interchangeably.

In the context of alignment, the difficulty is that the reward function is unknown but only preferences over two actions are observed. Typically, the Bradley-Terry assumption (Bradley & Terry, 1952) is used:

$$\mathbb{P}\left(a > a'|s\right) = \frac{\exp(r(s,a))}{\exp(r(s,a)) + \exp(r(s,a'))},$$

where the symbol $a > a'$ means that a is more preferred compared with a'. Given a preference dataset $D_{\text{pref}} = \{(s_i, a_i, a_i')\}_{i=1}^n$, where $a_i > a_i'$ is assumed without loss of generality, the reward learning objective, derived via maximum likelihood estimation, is

$$\widehat{r} \leftarrow \operatorname*{argmax}_r \sum_{i=1}^n \log\left(\sigma\left(r(s_i,a_i) - r(s_i,a_i')\right)\right), \quad (s_i,a_i,a_i') \sim D_{\text{pref}}. \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid function. This objective encourages the reward function to give a high score for the positively preferred data (s,a) and a low score for the negative preferred data (s,a').

Let π_{ref} be a reference policy model and $\beta > 0$ be a hyper-parameter. Ideally, we may want to optimize the policy with this recovered reward function in population:

$$\pi_{\widehat{r}}^\star \leftarrow \operatorname*{argmax}_\pi \mathbb{E}_{s\sim\rho(\cdot)}\left\{\mathbb{E}_{a\sim\pi(\cdot|s)}\left[\widehat{r}(s,a)\right] - \beta D_{\text{KL}}(\pi(\cdot|s), \pi_{\text{ref}}(\cdot|s))\right\}.$$

Here, the Kullback–Leibler (KL) penalty aims to mitigate the reward hacking and over-optimization issue (Gao et al., 2023). We remark that, in practice, we do not know the distribution $\rho(\cdot)$ and typically employ Monte Carlo approximations. That is, we use finite samples to approximate the population distribution and its expectation. There are two kinds of practical approaches, which we elaborate on below. Please also see Figure 3.

Reward-model-free Optimization Approaches: One direct idea is to use state-action pairs from the preference dataset for policy optimization:

$$\widehat{\pi}_{\text{RMF-PO}} \leftarrow \operatorname*{argmax}_\pi \sum_{i=1}^n \sum_{a\in\{a_i,a_i'\}} \widehat{r}(s_i,a) - \beta D_{\text{KL}}\left(\pi(a|s_i), \pi_{\text{ref}}(a|s_i)\right), \quad (s_i,a_i,a_i') \sim D_{\text{pref}}. \tag{3}$$

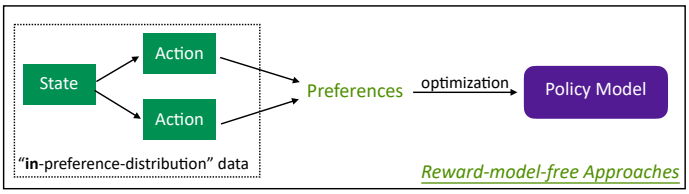
This approach approximate the state and action distributions by finite samples from the preference dataset. By using tools from KL-regularized optimization (see e.g., (Vieillard et al., 2020)), Rafailov et al. (2023) showed that procedures in Eq. (2) and Eq. (3) could be integrated into a single objective:

$$\widehat{\pi}_{\text{DPO}} \leftarrow \operatorname*{argmax}_\pi \sum_{i=1}^n \log\sigma\left(\beta\log\frac{\pi(a_i|s_i)}{\pi_{\text{ref}}(a_i|s_i)} - \beta\log\frac{\pi(a_i'|s_i)}{\pi_{\text{ref}}(a_i'|s_i)}\right), \quad (s_i,a_i,a_i') \sim D_{\text{pref}}. \tag{4}$$

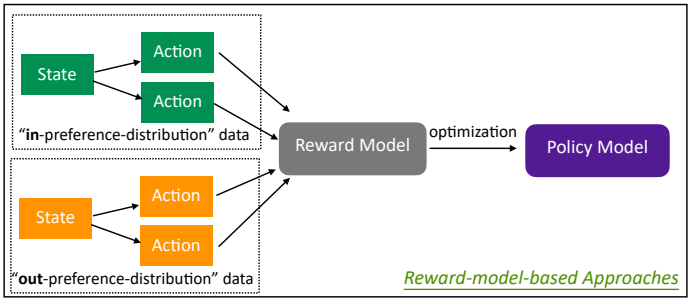
The resultant algorithm is named Direct Preference Optimization (DPO). Since this approach does not require explicitly training a reward model, it is considered as reward-model-free optimization.

Reward-model-based Optimization Approaches: The vanilla Reward-Model-Based Policy Optimization (RMB-PO) approach also leverages states from the preference dataset but samples actions from the policy model:

$$\widehat{\pi}_{\text{RMB-PO}} \leftarrow \operatorname*{argmax}_\pi \sum_{i=1}^n \mathbb{E}_{a\sim\pi(\cdot|s_i)}\left[\widehat{r}(s_i,a)\right] - \beta D_{\text{KL}}(\pi(\cdot|s_i), \pi_{\text{ref}}(\cdot|s_i)), \quad s_i \sim D_{\text{pref}}. \tag{5}$$



(a) Illustration for reward-model-free approaches.



(b) Illustration for reward-model-based approaches.

Figure 3: Illustration for policy optimization methods. For reward-model-based approaches, the reward model learning procedure is not plotted for ease of presentation.

We consider the exact action expectation $\mathbb{E}_{a\sim\pi(\cdot|s)}[\widehat{r}(s,a)]$ in the above formulation, and this expectation can be approximated by sampling multiple actions. This approximation error can be mitigated by computational power, and we do not consider this error in this paper.

In (Ouyang et al., 2022; Touvron et al., 2023), a variant of RMB-PO, refereed to as RMB-PO+ in this paper, further leverages a new, *preference-free* dataset $D_{\text{pref-free}} = \{s_j\}_{j=1}^m$:

$$\widehat{\pi}_{\text{RMB-PO+}} \leftarrow \operatorname*{argmax}_{\pi} \sum_{j=1}^{m} \mathbb{E}_{a\sim\pi(\cdot|s_j)}\left[\widehat{r}(s_j,a)\right] - \beta D_{\text{KL}}(\pi(\cdot|s_j), \pi_{\text{ref}}(\cdot|s_j)),\ \ s_j \sim D_{\text{pref-free}}. \tag{6}$$

Note that the dataset $D_{\text{pref-free}}$ is cheap to obtain and usually $m \geq n$ (Ouyang et al., 2022). One particular example of such data in language model's application is the `lmsys-chat-1m` dataset (Zheng et al., 2023), which has 1 million prompts from real users without preference annotations.

We note that there is no single learning objective for reward-model-based approaches. This is because the technique in (Rafailov et al., 2023) requires that the reward and policy learning objectives have the same training distribution, a condition that is not met for reward-model-based approaches. In practice, policy optimization in Eq. (5) and Eq. (6) can be conducted by policy gradient methods such as PPO (Schulman et al., 2017) and ReMax (Li et al., 2023).

B Theoretical Analysis

In this section, we present a preliminary analysis of errors in the optimization methods. At a high level, we identify three types of errors:

1) the reward evaluation error $|\widehat{r}(s,a) - r(s,a)|$;
2) the estimation error when using finite samples to calculate the expectation $\mathbb{E}_{a\sim\pi(\cdot|s)}[\cdot]$;
3) the estimation error when using finite samples to calculate the expectation $\mathbb{E}_{s\sim\rho(\cdot)}[\cdot]$.

The first error primarily results from finite preference data and diminishes to zero as the preference data size increases indefinitely. This error exists in all optimization methods. Compared with DPO, RMB-PO aims to mitigate the second error, while RMB-PO+ further reduces the third error. We note

that RMB-PO and RMB-PO+ do not increase the sample complexity of preference data but only incur additional computational steps. The theory is presented in Proposition 1 and the proof is given below.

Proof. We first consider the reward error:

$$\begin{aligned}
&r(\pi_r^\star) - r(\widehat{\pi}_{\text{RMB-PO}}) \\
&= \mathbb{E}_s\mathbb{E}_{a\sim\pi_r^\star}[r(s,a)] - \mathbb{E}_s\mathbb{E}_{a\sim\widehat{\pi}_{\text{RMB-PO}}}[r(s,a)] \\
&= \mathbb{E}_s\mathbb{E}_{a\sim\pi_r^\star}[r(s,a)] - \mathbb{E}_s\mathbb{E}_{a\sim\pi_r^\star}[\widehat{r}(s,a)] + \mathbb{E}_s\mathbb{E}_{a\sim\pi_r^\star}[\widehat{r}(s,a)] - \mathbb{E}_s\mathbb{E}_{a\sim\widehat{\pi}_{\text{RMB-PO}}}[r(s,a)] \\
&\le \varepsilon_r + \mathbb{E}_s\mathbb{E}_{a\sim\pi_r^\star}[\widehat{r}(s,a)] - \mathbb{E}_s\mathbb{E}_{a\sim\widehat{\pi}_{\text{RMB-PO}}}[\widehat{r}(s,a)] \\
&\quad + \mathbb{E}_s\mathbb{E}_{a\sim\widehat{\pi}_{\text{RMB-PO}}}[\widehat{r}(s,a)] - \mathbb{E}_s\mathbb{E}_{a\sim\widehat{\pi}_{\text{RMB-PO}}}[r(s,a)] \\
&\le 2\varepsilon_r + \mathbb{E}_s\mathbb{E}_{a\sim\pi_r^\star}[\widehat{r}(s,a)] - \mathbb{E}_s\mathbb{E}_{a\sim\widehat{\pi}_{\text{RMB-PO}}}[\widehat{r}(s,a)]. \qquad (7)
\end{aligned}$$

Then we consider the state distribution estimation error:

$$\begin{aligned}
&\mathbb{E}_s\mathbb{E}_{a\sim\pi_r^\star}[\widehat{r}(s,a)] - \mathbb{E}_s\mathbb{E}_{a\sim\widehat{\pi}_{\text{RMB-PO}}}[\widehat{r}(s,a)] \\
&= \mathbb{E}_s\mathbb{E}_{a\sim\pi_r^\star}[\widehat{r}(s,a)] - \widehat{\mathbb{E}}_s\mathbb{E}_{a\sim\pi_r^\star}[\widehat{r}(s,a)] + \widehat{\mathbb{E}}_s\mathbb{E}_{a\sim\pi_r^\star}[\widehat{r}(s,a)] - \mathbb{E}_s\mathbb{E}_{a\sim\widehat{\pi}_{\text{RMB-PO}}}[\widehat{r}(s,a)] \\
&\le \varepsilon_s + \widehat{\mathbb{E}}_s\mathbb{E}_{a\sim\pi_r^\star}[\widehat{r}(s,a)] - \widehat{\mathbb{E}}_s\mathbb{E}_{a\sim\widehat{\pi}_{\text{RMB-PO}}}[\widehat{r}(s,a)] \\
&\quad + \widehat{\mathbb{E}}_s\mathbb{E}_{a\sim\widehat{\pi}_{\text{RMB-PO}}}[\widehat{r}(s,a)] - \mathbb{E}_s\mathbb{E}_{a\sim\widehat{\pi}_{\text{RMB-PO}}}[\widehat{r}(s,a)] \\
&\le \varepsilon_s + 0 + \varepsilon_s. \qquad (8)
\end{aligned}$$

Combining (7) and (8) proves the first result in Proposition 1. For the second result, we may replace $\widehat{\pi}_{\text{RMB-PO}}$ with $\widehat{\pi}_{\text{RMF-PO}}$ in the above proof to obtain:

$$r(\pi_r^\star) - r(\widehat{\pi}_{\text{RMF-PO}}) \le 2\varepsilon_r + 2\varepsilon_s + \widehat{\mathbb{E}}_s\mathbb{E}_{a\sim\pi_r^\star}[\widehat{r}(s,a)] - \widehat{\mathbb{E}}_s\mathbb{E}_{a\sim\widehat{\pi}_{\text{RMF-PO}}}[\widehat{r}(s,a)]. \qquad (9)$$

Then we consider the action distribution estimation error:

$$\begin{aligned}
&\widehat{\mathbb{E}}_s\mathbb{E}_{a\sim\pi_r^\star}[\widehat{r}(s,a)] - \widehat{\mathbb{E}}_s\mathbb{E}_{a\sim\widehat{\pi}_{\text{RMF-PO}}}[\widehat{r}(s,a)] \\
&= \widehat{\mathbb{E}}_s\mathbb{E}_{a\sim\pi_r^\star}[\widehat{r}(s,a)] - \widehat{\mathbb{E}}_s\widehat{\mathbb{E}}_{a\sim\pi_r^\star}[\widehat{r}(s,a)] + \widehat{\mathbb{E}}_s\widehat{\mathbb{E}}_{a\sim\pi_r^\star}[\widehat{r}(s,a)] - \widehat{\mathbb{E}}_s\mathbb{E}_{a\sim\widehat{\pi}_{\text{RMF-PO}}}[\widehat{r}(s,a)] \\
&\le \varepsilon_a + \widehat{\mathbb{E}}_s\widehat{\mathbb{E}}_{a\sim\pi_r^\star}[\widehat{r}(s,a)] - \widehat{\mathbb{E}}_s\widehat{\mathbb{E}}_{a\sim\widehat{\pi}_{\text{RMF-PO}}}[\widehat{r}(s,a)] + \qquad (10) \\
&\widehat{\mathbb{E}}_s\widehat{\mathbb{E}}_{a\sim\widehat{\pi}_{\text{RMF-PO}}}[\widehat{r}(s,a)] - \widehat{\mathbb{E}}_s\mathbb{E}_{a\sim\widehat{\pi}_{\text{RMF-PO}}}[\widehat{r}(s,a)] \\
&\le \varepsilon_a + 0 + \varepsilon_a \qquad (11)
\end{aligned}$$

Combining (9) and (11) proves the second result in Proposition 1. □

C EXPERIMENTS

In this section, we conduct numerical experiments to validate the improvement of RMB-PO and RMB-PO+ by better stochastic approximation. All of our experiments are run with 10 different random seeds (2021-2030), and the averaged results are reported[1]. Note that we set π_{REF} to be a policy with a uniform action distribution in all experiments and $\beta = 0.01$ for all methods. Besides, we use a policy with a uniform action distribution to collect the preference data.

C.1 LINEAR BANDIT

We study a linear bandit task, where we have $r(s,a) = \phi_r(s,a)^\top \theta_r^\star$, with $\phi_r(s,a) \in \mathbb{R}^d$ denoting the feature representation and $\theta_r^\star \in \mathbb{R}^d$ as the parameter. In this case, the reward learning optimization problem is convex, so we use CVXPY (Diamond & Boyd, 2016) to find the solution $\widehat{r}$. In particular, we use the feature map $\phi_r(s,a)$ and the parameter $\theta_r^\star$ as

$$\phi_r(s,a) = \left((a+1)\cdot\cos(s\cdot\pi), \frac{1}{a+1}\cdot\sin(s\cdot\pi)\right)^\top, \quad \theta_r^\star = (1,2)^\top,$$

[1]We exclude the worst and best results to make a robust estimation of the performance.

where $s \in \mathcal{S} = [0, 1]$ and $a \in \mathcal{A} = \{0, 1, 2, 3\}$. A uniform distribution over $\mathcal{S}$ is studied. For the policy, we consider the parameterization

$$\pi(a|s) = \frac{\exp(\phi_\pi(s, a)^\top \theta_\pi)}{\sum_{a'} \exp(\phi_\pi(s, a')^\top \theta_\pi)},$$

with $\phi_\pi(s, a)$ and θ_π both in $\mathbb{R}^2$. In this case, the policy optimization problem is a non-convex problem, but the gradient domination condition holds (Agarwal et al., 2021). We use the gradient ascent method with the AdaGrad optimizer (Duchi et al., 2011) (a step size of 0.1 is used).

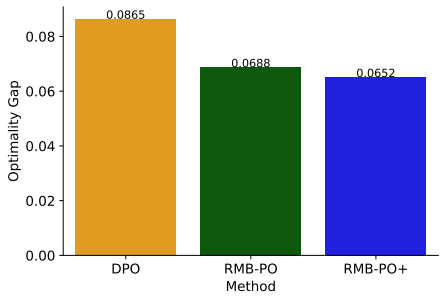


Figure 4: Optimality gap with $\phi_\pi = \phi_r$.

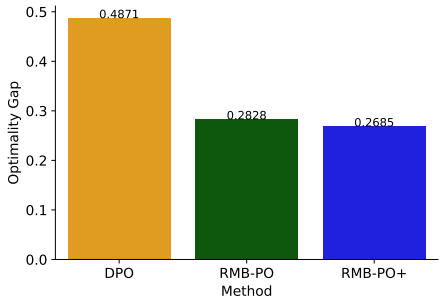


Figure 5: Optimality gap with $\phi_\pi \neq \phi_r$.

We examine two scenarios. In the first scenario, there is no feature mismatch between the reward and policy models, i.e., $\phi_\pi = \phi_r$. In the second, we use a different feature map for policy:

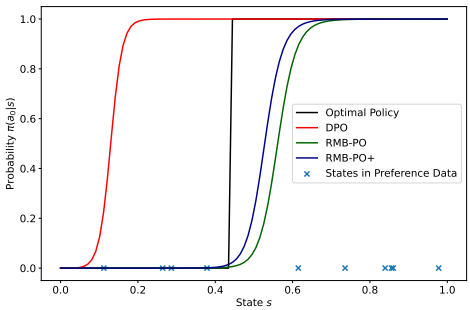
$$\phi_\pi(s, a) = \left((a + 1) \cdot \sin(s \cdot \pi), \frac{1}{a + 1} \cdot \cos(s \cdot \pi)\right)^\top.$$

We believe that in scenarios where $\phi_\pi \neq \phi_r$, RMB-PO approaches could exhibit more promising performance than RMF-PO approaches. This is because, in such cases, the policy and reward models may align well by learning from preference data. However, in out-of-preference-distribution scenarios, they may extrapolate and generalize quite differently due to mismatches in representations. Nevertheless, RMB-PO approaches could use out-of-preference-distribution data to mitigate these mismatches and tend to perform well. The case where $\phi_\pi \neq \phi_r$ will be revisited in later neural bandit experiments, where the policy model and reward model typically utilize distinct architectures and learn distinct representations.

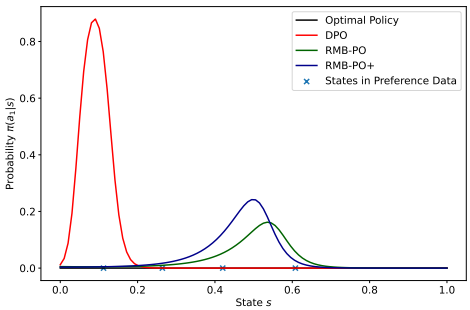
In our experiments, we set the size of preference data to be $n = 20$ and the size of preference-free data to be $m = 10n$, resulting in training accuracy of the reward model ranging from 60% to 80% over 10 experiments. We display the optimality gap $|r(\pi^\star) - r(\widehat{\pi})|$ (the smaller, the better) in Figure 4 and Figure 5, where $r(\pi)$ is the evaluation performance of a policy π, i.e., $r(\pi) = \mathbb{E}_{s\sim\rho(\cdot)}\mathbb{E}_{a\sim\pi(\cdot|s)}[r(s, a)]$ (in our experiments, we use 5000 sampled states to approximate this expectation).

From Figure 4, we see that even though the policy model is provided with a good feature (e.g., in Figure 4), RMB-PO methods can benefit from out-of-preference data. In the case where $\phi_\pi \neq \phi_r$ in Figure 5, we find that RMB-PO+ is better than RMB-PO by leveraging additional preference-free data. Thus, we believe it is crucial to learn the optimal action (as inferred by the reward model) on out-of-preference data, even when the two models share the same good feature.

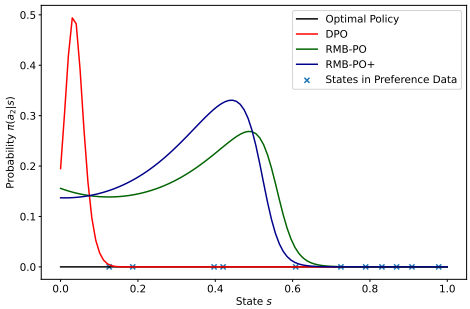
To gain a better understanding, we also visualize the learned policy distribution in the $\phi_\pi \neq \phi_r$ setting; see Figure 6. To observe the training distribution coverage, we plot the states from the preference dataset. Additional states used in RMB-PO+ almost cover the entire state space but are not shown for readability reasons. From the reported curves, we observe that DPO aligns well with the optimal policy in the regions covered by preference data, and RMB-PO(+) methods tend to perform better than DPO in the out-of-distribution regime not covered by the preference data.



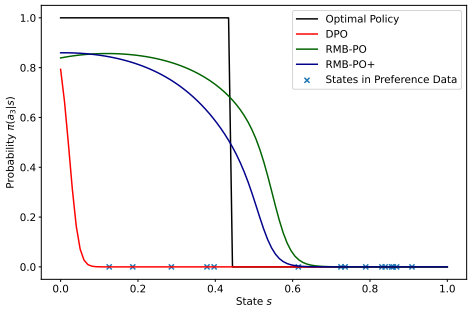
(a) Action a_0.



(b) Action a_1.



(c) Action a_2.



(d) Action a_3.

Figure 6: Probabilities of four actions a_0, a_1, a_2 and a_3. Results illustrate that RMB-PO(+) methods leverage out-of-preference data to better learn the policy distribution on out-of-distribution states and improve the generalization performance.

Following the same setup, we provide ablation studies regarding the size of preference-free data used in RMB-PO+. See the results in Figure 7 and Figure 8. We find that the previous conclusions still hold true.

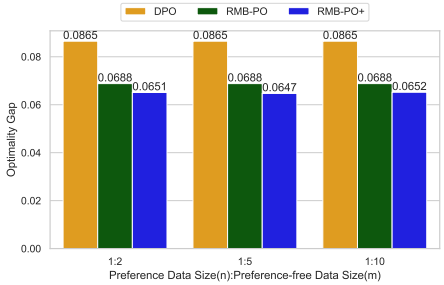


Figure 7: Optimality gap with $\phi_\pi = \phi_r$.

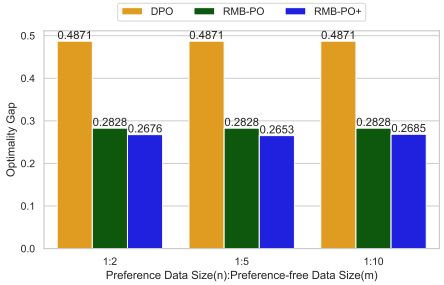
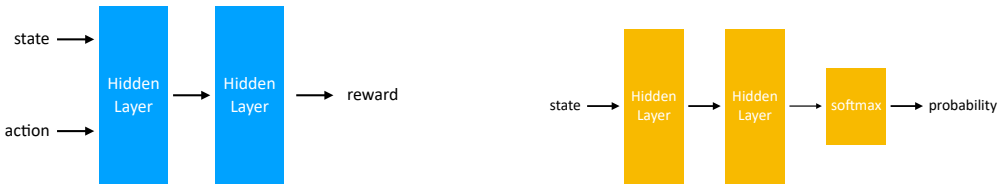


Figure 8: Optimality gap with $\phi_\pi \neq \phi_r$.

C.2 Neural Bandit

In this section, we study a neural bandit problem. Specifically, we study the case where $r(s, a) = f_{\theta_r^\star}(s, a)$, with $f_{\theta_r^\star}$ being a fixed 1-hidden-layer multi-layer perceptron (MLP) neural network, having a hidden size of 64. For reward learning, we use a 2-hidden-layer MLP with a hidden size of 64, and the policy network is also a 2-hidden-layer MLP with a hidden size of 64. We consider a continuous state space $\mathcal{S} = [-1, 1]^{50}$ and a discrete action space $\mathcal{A} = \{0, 1, 2, \ldots, 9\}$. The state distribution ρ is uniform and one-hot feature representation for actions is used.

We note that, unlike in the linear bandit case where we could fix the feature representations of the reward and policy models to be the same, in this case, the feature representations of the reward and policy models are purely learned from the given data. The architectures of the reward and policy



(a) Reward neural network. (b) Policy neural network.

Figure 9: Architectures of the reward and policy models.

models are shown in Figure 9. All neural networks are optimized using the Adam optimizer (Kingma & Ba, 2015) with a step size of 10^{-3}.

We run experiments with varying sizes of preference-free data m while fixing the preference data size at $n = 50$. We report the results in Figure 10. First, we observe that RMB-PO and RMB-PO+ significantly outperform DPO. Furthermore, simply using a preference-free data size that is twice as large already improves performance over RMB-PO, and further scaling does not help too much.

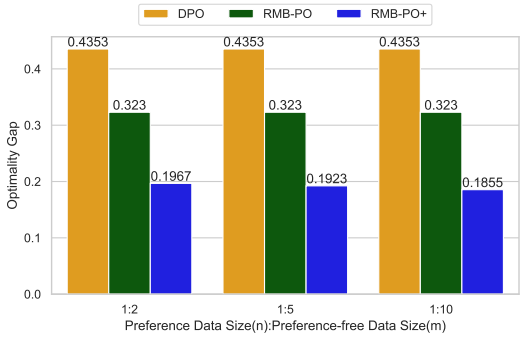


Figure 10: Optimality gap of learned policies in the neural bandit task.

D Discussion

Our research is related to imitation learning (Osa et al., 2018), which aims to learn a policy from expert demonstrations. A popular approach to achieve this goal is through behavioral cloning (BC) (Pomerleau, 1991), which trains a policy model by maximizing the likelihood of expert data. Note that the working mechanism of BC is quite similar to DPO, as in Eq. (4), where the likelihood of positively preferred actions is increased and that of negatively preferred actions is decreased:

$$\pi_{\mathrm{BC}} \leftarrow \operatorname*{argmax}_{\pi} \sum_{i=1}^{n} \log \pi(a_i|s_i), \quad (s_i, a_i) \sim D_{\mathrm{E}},$$

where D_{E} is the expert dataset.

Ghasemipour et al. (2019) showed that another class of imitation methods, known as adversarial imitation learning (AIL) methods, (such as GAIL (Ho & Ermon, 2016)), usually performs better than BC. In particular, AIL methods leverage a recovered reward function to perform policy optimization on "out-of-expert-data" through online interaction, significantly improving performance. Following the formulation in (Xu et al., 2022), the training objective of reward-model-based AIL can be re-formulated as

$$\pi_{\mathrm{AIL}} \leftarrow \operatorname*{argmin}_{\pi} \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \left| d_{\pi}(s,a) - \widehat{d}_{\mathrm{E}}(s,a) \right|,$$

where $\widehat{d}_{\mathrm{E}}$ is the empirical state-action distribution estimated from D_{E}, and $d_{\pi}(s,a)$ is obtained from online interaction. For the optimization objective of AIL, it utilizes states beyond those in the expert dataset (reflected in the summation over all state-action pairs). We notice that Xu et al. (2022)

theoretically proved that AIL can outperform BC in terms of addressing the distribution shift issue with optimization on "out-of-expert data". The idea of recovering a reward function and using it to perform extensive policy optimization is quite similar to the framework of RLHF.

Additionally, our research is related to transition-model-based reinforcement learning (RL) methods, where the goal is to find an optimal policy through interactions with environments. Many empirical successes suggest that transition-model-based approaches are superior in terms of sample complexity (Luo et al., 2019; Janner et al., 2019). We do not aim to present a detailed discussion since RL involves lots of concepts and notations. Instead, we would like to highlight that our findings align with the understanding that additional policy optimization on transition-model-generated data is helpful. We would like to refer readers to (Hafner et al., 2020; Schrittwieser et al., 2020; Yu et al., 2020; Luo et al., 2023) for the effect of data augmentation in transition-model-based RL methods.

Finally, we note that compared with reward-model-free methods such as DPO (Rafailov et al., 2023), reward-model-based policy optimization (RMB-PO) methods do not require extra preference annotation. For applications such as language models, training and storing a reward model has been shown to be highly efficient, as demonstrated in (Yao et al., 2023). The primary challenge in RMB-PO lies in the huge action space during policy optimization. However, this issue can be effectively addressed by computationally efficient methods like those proposed by (Dong et al., 2023; Li et al., 2023). Notably, Li et al. (2023) showed that optimizing the language model with prompts-only data can improve performance, a setting that cannot achieved by reward-model-free approaches such as DPO.