DATA REFINEMENT: OVERCOMING REWARD OVERFITTING AND OVER-OPTIMIZATION IN RLHF

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement Learning with Human Feedback (RLHF) is a pivotal technique that aligns language models closely with human-centric values. The initial phase of RLHF involves learning human values using a reward model from pairwise or K-wise comparisons. It is observed that the performance of the reward model degrades after one epoch of training, and optimizing too much against such reward model eventually hinders the true objective. This paper delves into these issues, using the theoretical insights to introduce improved reward learning algorithms termed "data refinement". The core idea is that during each training epoch, we not only update the model with the data, but also refine the data using the model, replacing hard labels with soft labels. This helps reduce the noise introduced by the imbalanced data coverage. Our empirical findings highlight the superior performance of this approach over the traditional methods.

1 INTRODUCTION

Large Language Models (LLMs) have made remarkable progress, profoundly influencing the AI community (Chowdhery et al., 2022; Brown et al., 2020; Touvron et al., 2023; Bubeck et al., 2023). However, without careful fine-tuning, LLMs are likely to express unintended and unpredictable behavior. These include fabricating facts, generating biased or toxic text, and even harmful content to humans (Perez et al., 2022; Ganguli et al., 2022). As large language models grow in capability, aligning them with human values becomes paramount. As an initial step towards this target, Reinforcement Learning with Human Feedback (RLHF) proposes to first learn the human value as a reward function from pairwise or K-wise comparisons of model responses, and then fine-tune the language model based on the learned reward function to align with human values and societal norms Ziegler et al. (2019); Ouyang et al. (2022).

A typical deployment of RLHF for language modeling includes the following steps:

- **Supervised Fine-tuning:** Fine-tune a pre-trained Large Language Model (LLM) using supervised training.
- **Reward Learning:** Collect human preference data in the format of pairwise or multi-wise comparisons of different responses. Train a reward model based comparison data using Maximum Likelihood Estimator (MLE).
- **Policy Learning:** Fine-tune the existing LLM based on the learned reward model using Proximal Policy Optimization (PPO).

Although RLHF has achieved great empirical success Bai et al. (2022), the current reward training paradigm grapples with significant value-reward mismatches. In the practical training of the reward model, it is observed in Ouyang et al. (2022) that the reward function deteriorates after one epoch of training. Furthermore, Gao et al. (2022) introduces an over-optimization phenomenon, where optimizing the policy too much with the learned reward can hurt the ground-truth reward.

In this paper, we investigate the two phenomenons. We start with the multi-armed bandit setting to reproduce the two issues. Based on our theoretical analysis, we attribute both phenomenons to the imbalanced data coverage for training reward model in the non-asymptotic regime. Although the maximum likelihood estimator converges asymptotically to the ground-truth reward, it may fail to converge in the non-asymptotic region.

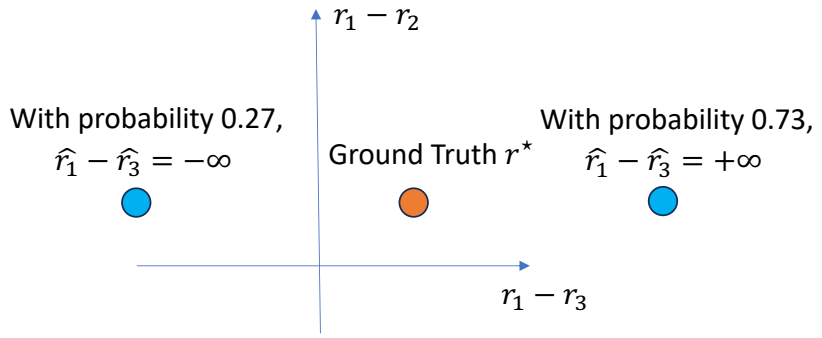

Figure 1: Illustration of the problem of MLE for learning the ground truth reward. With a small number of samples comparing arm 1 and 3, the MLE converges to a solution which assigns $\hat{r}_1 - \hat{r}_3 = -\infty$ with constant probability.

As is shown in Figure 1, in a simple 3-armed bandit problem, when the arm 1 and arm 3 is only compared once, there is constant probability that their comparison result is inconsistent with the ground-truth, leading to the estimated reward to be infinity for sub-optimal arms. We show that such issue leads to both reward overfitting in the reward learning phase and reward over-optimization in the policy learning phase.

To mitigate these effects, we leverage the theoretical insights from pessimism-based ideas to design better reward learning algorithms, data refinement, that simultaneously improve both reward overfitting and reward over-optimization. The algorithm design is straightforward: in each epoch, beyond updating the model with the data, we also update the data with the model in order to remove the noisy data with a capable model. Theoretically, we investigate the two phenomenons in the tabular bandit case, showing that the proposed data refinement methods, as an alternative to lower confidence bound algorithm, shares similar insight and resolves the two issues. Empirically, we show strong empirical evidence that the proposed method improves the reward training with both bandit case and neural network parameterization.

1.1 Related Work

RLHF and Preference-based Reinforcement Learning. RLHF, or Preference-based Reinforcement Learning, has delivered significant empirical success in the fields of game playing, robot training, stock-prediction, recommender systems, clinical trials, large language models etc. (Novoseller et al., 2019; Sadigh et al., 2017; Christiano et al., 2017b; Kupcsik et al., 2018; Jain et al., 2013; Wirth et al., 2017; Knox & Stone, 2008; MacGlashan et al., 2017; Christiano et al., 2017a; Warnell et al., 2018; Brown et al., 2019; Shin et al., 2023; Ziegler et al., 2019; Stiennon et al., 2020; Wu et al., 2021; Nakano et al., 2021; Ouyang et al., 2022; Menick et al., 2022; Glaese et al., 2022; Gao et al., 2022; Bai et al., 2022; Ganguli et al., 2022; Ramamurthy et al., 2022). There have been work exploring the efficient fine-tuning of policy model Snell et al. (2022); Song et al. (2023); Yuan et al. (2023); Zhu et al. (2023b). However, the investigation of reward learning is still missing, with the exception of Zhu et al. (2023a).

Learning and Estimation from Pairwise Comparison and Ranking. The problem of estimation and ranking from pairwise or K-wise comparisons has been studied extensively in the literature. In the literature of *dueling bandit*, one compares two actions and aims to minimize regret based on pairwise comparisons (Yue et al., 2012; Zoghi et al., 2014b; Yue & Joachims, 2009; 2011; Saha & Krishnamurthy, 2022; Ghoshal & Saha, 2022; Saha & Gopalan, 2018a; Ailon et al., 2014; Zoghi et al., 2014a; Komiyama et al., 2015; Gajane et al., 2015; Saha & Gopalan, 2018b; 2019; Faury et al., 2020). Novoseller et al. (2019); Xu et al. (2020) analyze the sample complexity of dueling RL under the tabular case, which is extended to linear case and function approximation by the recent work Pacchiano et al. (2021); Chen et al. (2022). Chatterji et al. (2022) studies a close setting where in each episode only binary feedback is received. However, most of the work focuses on regret minimization. We take a first step towards the theoretical analysis for function approximation for

K-wise comparisons with policy learning as the target. Zhu et al. (2023a) analyzes the sample complexity of RLHF in the offline setting.

Knowledge distillation There have been a family of knowledge distillation methods that try to use trained models to update existing or new models (Hinton et al., 2015; Furlanello et al., 2018; Cho & Hariharan, 2019; Zhao et al., 2022; Romero et al., 2014; Yim et al., 2017; Huang & Wang, 2017; Park et al., 2019; Tian et al., 2019; Tung & Mori, 2019; Qiu et al., 2022; Cheng et al., 2020). Notably, Furlanello et al. (2018) proposes born-again network, which iteratively trains a new student neural network after the teacher network achieves the smallest evaluation loss. Both our data refinement idea and knowledge distillation idea rely on soft label update. However, data refinement iteratively update the single model and data, while knowledge distillation method usually focuses on transferring knowledge from one model to the other.

2 FORMULATION

We begin with the notation that we use in the paper. Then we introduce the general formulation of RLHF, along with our simplification in the multi-armed bandit case.

Notations. We use calligraphic letters for sets, e.g., $\mathcal{S}$ and $\mathcal{A}$. Given a set $\mathcal{S}$, we write $|\mathcal{S}|$ to represent the cardinality of $\mathcal{S}$. We use $[K]$ to denote the set of integers from 1 to K. We use $\mu(a)$ to denote the probability of comparing a with any other arms, and $\mu(a, a')$ to denote the probability of comparing a and a'. Similarly, we use $n(a), n(a, a')$ to denote the number of samples that compare a with any other arms, and the number of samples that compare a with a', respectively.

2.1 RLHF IN MULTI-ARMED BANDIT

To understand the overfitting and over-optimization problem, we simplify the RLHF problem by considering a single-state multi-armed bandit formulation. Instead of fitting a reward model and policy model with a complex neural network, we fit a tabular reward model for a K-armed bandit problem. In this case, the policy becomes a distribution supported on the K arms $\pi \in \Delta([K])$.

We first formulate the corresponding reward learning and policy learning problem under this setting, phenomenon of reward overfitting and reward over-optimization in the simplified setting. Then we analyze the hard instance, connect them to the theory of pessimism for offline bandit and RL problems. We also propose another surrogate solution for the problem, named data refinement, that is easier to extend to the case of neural network.

Consider a multi-armed bandit problem with K arms, i.e. $\mathcal{A} = [K]$. Each arm has a deterministic ground-truth reward $r^\star(k) \in \mathbb{R}, k \in [K]$. Let the sampling process be the following: we first sample two actions a_i, a_i' from a joint distribution $\mu \in \Delta([K] \times [K])$, and then observe a binary comparison variable c_i following a distribution

$$\mathbb{P}(c_i = 1) = \frac{\exp(r^\star(a_i))}{\exp(r^\star(a_i)) + \exp(r^\star(a_i'))}, \quad \mathbb{P}(c_i = 0) = 1 - \mathbb{P}(c_i = 1).$$

Assume that we are given n samples, which are sampled $i.i.d.$ from the above process. Let $n(a, a')$ be the total number of comparisons between actions a and a' in the n samples. Let the resulting dataset be $\mathcal{D} = \{a_i, a_i', c_i\}_{i=1}^n$. The task in RLHF is:

1. **Reward Learning:** Estimate the true reward $r^\star$ with a proxy reward $\hat{r}$ from the comparison dataset $\mathcal{D}$.
2. **Policy Learning:** Train a policy $\pi \in \Delta([K])$ by maximizing the proxy reward under KL constraints.

In the next two sections, we discuss separately the reward learning phase and policy learning phase, along with the reason behind overfitting and over-optimization.

2.2 OVERFITTING IN REWARD LEARNING

For reward learning, the commonly used maximum likelihood estimator is the estimator that minimizes empirical cross entropy loss:

$$\hat{r}_{\mathsf{MLE}} = \arg\min \mathcal{L}_{\mathsf{CE}}(\mathcal{D}), \text{ where} \tag{1}$$

$$\mathcal{L}_{\mathsf{CE}}(\mathcal{D}, \hat{r}) = \frac{1}{n}\sum_{i=1}^{n}\left(\mathbb{1}(c_i = 1)\log\left(\frac{\exp(\hat{r}(a_i))}{\exp(\hat{r}(a_i)) + \exp(\hat{r}(a_i'))}\right) + \mathbb{1}(c_i = 0)\log\left(\frac{\exp(\hat{r}(a_i'))}{\exp(\hat{r}(a_i)) + \exp(\hat{r}(a_i'))}\right)\right)$$

By definition, $\hat{r}_{\mathsf{MLE}}$ is the reward that the reward learning procedure converges to. Thus the performance of $\hat{r}_{\mathsf{MLE}}$ is a good indicator whether overfitting exists during reward training.

We define the population cross entropy loss as

$$\begin{aligned}
\mathcal{L}_{\mathsf{CE}}(r) = \mathbb{E}_{(a,a')\sim\mu, c\sim\mathsf{Ber}(\frac{\exp(r^\star(a))}{\exp(r^\star(a))+\exp(r^\star(a)')})}\Bigg[&\mathbb{1}(c=1)\log\left(\frac{\exp(r(a))}{\exp(r(a))+\exp(r(a'))}\right)\\
&+\mathbb{1}(c=0)\log\left(\frac{\exp(r(a'))}{\exp(r(a))+\exp(\hat{r}(a_i'))}\right)\Bigg]\\
= \mathbb{E}_{(a,a')\sim\mu}\Bigg[&\frac{\exp(r^\star(a))}{\exp(r^\star(a))+\exp(r^\star(a)')}\log\left(\frac{\exp(r(a))}{\exp(r(a))+\exp(r(a'))}\right)\\
&+\frac{\exp(r^\star(a'))}{\exp(r^\star(a))+\exp(r^\star(a)')}\log\left(\frac{\exp(r(a'))}{\exp(r(a))+\exp(\hat{r}(a_i'))}\right)\Bigg]
\end{aligned}$$

For a fixed pairwise comparison distribution μ, it is known that maximum likelihood estimator $\hat{r}_{\mathsf{MLE}}$ converges to the ground truth reward $r^\star$ as the number of samples n goes to infinity.

Theorem 1 (Hastie et al. (2009))**.** *For any fixed μ, and any given ground-truth reward $r^\star$, the limit of the expected excess loss is given by*

$$\lim_{n\to\infty}\mathbb{E}_{\mathcal{D}}[\mathcal{L}_{\mathsf{CE}}(\hat{r}_{\mathsf{MLE}}) - \mathcal{L}_{\mathsf{CE}}(r^\star)] \to 0.$$

This suggests that the overfitting phenomenon will not exist when we have infinite number of samples. However, in the non-asymptotic regime when the comparison distribution μ may depend on n, one may not expect convergent result for MLE. We have the following theorem:

Theorem 2. *Fix $r^\star(a) = \mathbb{1}(a = 1)$. For any fixed n, there exists some μ such that with probability at least 0.1,*

$$\mathcal{L}_{\mathsf{CE}}(\hat{r}_{\mathsf{MLE}}) - \mathcal{L}_{\mathsf{CE}}(r^\star) \geq C$$

for any arbitrarily large C.

The proof is deferred to Appendix A. Below we provide a intuitive explanation. The constructed hard instance is a bandit where $r^\star(a) = \mathbb{1}(a = 1)$. For any fixed n, we set $\mu(a_1, a_2) = 1 - (K - 2)/n$, $\mu(a_1, a_k) = 1/n$ for any $2 \leq k \leq K$ and the rest pairs with 0.

In this hard instance, for each arm $a \in \{2, 3, \cdots, K\}$, there is constant probability that a is only compared with 1 once. And with constant probability, the comparison between arm 1 and arm a will be flipped. Since a is only compared with 1 for one time, the MLE will assign $r(a)$ as infinity when the reward function is not bounded, due to the same reason shown in Figure 1. Thus when optimizing the empirical cross entropy to the end, the maximum likelihood estimator will result in arbitrary large cross entropy loss. We also validate this phenomenon in Section 4.1 with simulated experiments.

This lower bound instance simulates the high-dimensional regime where the number of samples is comparable to the dimension, and the data coverage is unbalanced across dimensions.

2.3 OVER-OPTIMIZATION IN POLICY LEARNING

After getting the estimated reward function $\hat{r}$, we optimize the policy $\pi \in \Delta([K])$ to maximize the estimated reward. In RLHF, one starts from an initial policy π_0, and optimize the new policy π to maximize the estimated reward $\hat{r}$ under some constraint in KL divergence between π and π_0. It is observed in Gao et al. (2022) that as we continue optimizing the policy to maximize the estimated reward, the true reward of the policy will first increase then decrease, thus leading to Goodharting / reward over-optimization phenomenon.

Consider the following policy optimization problem for a given reward model $\hat{r}$:

$$\max_{\pi \in \Delta([K])} \mathbb{E}_{a \sim \pi(\cdot)}[\hat{r}(a)] - \frac{1}{\lambda} \cdot \mathsf{KL}(\pi \| \pi_0). \tag{2}$$

Assuming that the policy gradient method converges to the optimal policy for the above policy optimization problem, which has a closed-form solution as

$$\pi_\lambda(a) = \frac{\pi_0(a) \cdot \exp(\lambda \cdot \hat{r}(a))}{\sum_{a' \in \mathcal{A}} \pi_0(a') \cdot \exp(\lambda \cdot \hat{r}(a'))} \tag{3}$$

The optimal policy is a function of the estimated reward. Thus the over-optimization problem also reduces to the quality of the estimated reward.

In the tabular case, we can derive a closed form solution for how the KL divergence and ground-truth reward change with respect to λ, thus completely characterizing the reward-KL tradeoff. We can compute the KL divergence and ground-truth reward of the policy as

$$\begin{aligned}
\mathsf{KL}(\pi_\lambda \| \pi_0) &= \frac{\sum_{a \in \mathcal{A}} \pi_0(a) \cdot \exp(\lambda \cdot \hat{r}(a)) \cdot \log(\exp(\lambda \cdot \hat{r}(a)) / (\sum_{a' \in \mathcal{A}} \pi_0(a') \cdot \exp(\lambda \cdot \hat{r}(a'))))}{\sum_{a' \in \mathcal{A}} \pi_0(a') \cdot \exp(\lambda \cdot \hat{r}(a'))} \\
&= \frac{\sum_{a \in \mathcal{A}} \pi_0(a) \cdot \exp(\lambda \cdot \hat{r}(a)) \cdot \lambda \cdot \hat{r}(a)}{\sum_{a' \in \mathcal{A}} \pi_0(a') \cdot \exp(\lambda \cdot \hat{r}(a'))} - \log \left(\sum_{a' \in \mathcal{A}} \pi_0(a') \cdot \exp(\lambda \cdot \hat{r}(a')) \right), \\
\mathbb{E}_{a \sim \pi_\lambda}[r^\star(a)] &= \frac{\sum_{a \in \mathcal{A}} \pi_0(a) \cdot \exp(\lambda \cdot \hat{r}(a)) \cdot \lambda \cdot r^\star(a)}{\sum_{a' \in \mathcal{A}} \pi_0(a') \cdot \exp(\lambda \cdot \hat{r}(a'))}.
\end{aligned}$$

The above equation provides a precise characterization on how the mismatch between $\hat{r}$ and $r^\star$ leads to the over-optimization phenomenon. To simplify the analysis and provide better intuitions, we focus on the case when $\lambda \to \infty$, i.e. when the optimal policy selects the best empirical arm. In this case, the final policy reduces to the empirical best arm $\pi_\infty(a) = \mathbb{1}(a = \arg\max_{a'} \hat{r}(a'))$.

By definition, π_∞ is the convergent policy when we optimize Equation (2) to the end. Thus the performance of π_∞ is a good indicator whether over-optimization exists during policy training. We define the sub-optimality as below to characterize the performance gap between the convergent policy and the optimal policy.

$$\mathsf{SubOpt}(\hat{\pi}) \coloneqq \mathbb{E}[\max_a r^\star(a) - r^\star(\hat{\pi})].$$

We know by Theorem 1 that asymptotically, the MLE for reward $\hat{r}_{\mathsf{MLE}}$ converges to the ground truth reward $r^\star$. As a direct result, when using the MLE as reward, the sub-optimality of the policy π_∞ also converges to 0 with infinite number of samples.

However, similar to the case of reward learning overfitting, π_∞ may have arbitrarily bad sub-optimality in the non-asymptotic regime when trained from $\hat{r}_{\mathsf{MLE}}$.

Theorem 3. *Fix $r^\star(a) = \mathbb{1}(a = 1)$. For any fixed n, there exists some μ such that with probability at least 0.1,*

$$\mathbb{E}[\mathsf{SubOpt}(\hat{\pi}_\infty)] \geq 1.$$

The proof is deferred to Appendix B. This suggests that $\hat{r}_{\mathsf{MLE}}$ also leads to the reward over-optimization phenomenon in the non-asymptotic regime. In Section 4, we conduct simulation in the exact same setting to verify the loss.

3 METHODS: PESSIMISTIC MLE AND DATA REFINEMENT

The overfitting and over-optimization issue calls for a design of better and practical reward learning algorithm that helps mitigate both issues. We first discuss the pessimistic MLE algorithm in Zhu et al. (2023a), which is shown to guarantee a good convergent policy under good coverage assumption.

3.1 PESSIMISTIC MLE

In the tabular case, the pessimistic MLE corrects the original MLE by subtracting a confidence interval. Precisely, we have

$$\hat{r}_{\mathsf{PE}}(a) = \hat{r}_{\mathsf{MLE}}(a) - C \cdot \sqrt{\frac{1}{n(a)}}.$$

Intuitively, for those arms that are compared less times, we are more uncertain about their ground-truth reward value. Pessimistic MLE penalizes these arms by directly subtracting the confidence interval of each arm, making sure that the arms that are less often compared will not be chosen. It is shown in Zhu et al. (2023a) that the sub-optimality of the policy optimizing $\hat{r}_{\mathsf{PE}}$ converges to 0 under the following two conditions:

- The expected number of times that one compares optimal arm $\mu(a^\star)$ is lower bounded by some positive constant.
- The parameterized reward family lies in a bounded space, or with bounded moments.

Thus pessimistic MLE helps mitigate the reward over-optimization phenomenon when the conditions hold. However, for real-world reward training paradigm, the neural network is not bounded. Furthermore, estimating the exact confidence interval for a neural-network parameterized model is hard. These prevent the practical use of pessimistic MLE, and call for new methods that can potentially go beyond these conditions.

3.2 DATA REFINEMENT

We propose a new algorithm, data refinement, that leverages similar insights from pessimistic MLE. Intuitively, pessimistic-MLE helps mitigate the reward over-optimization issue by reducing the estimated reward for less seen arms. In data refinement, we achieve this by updating the label of the data we train on.

Algorithm 1 Data Refinement

Input: The pairwise comparison dataset $\mathcal{D} = \{a_i, a_i', c_i\}_{i=1}^n$. An parameterized reward model $\{r_\theta : \mathcal{A} \mapsto [0,1] \mid \theta \in \Theta\}$ with initialization $\theta_0 \in \Theta$. An empirical loss function

$$\mathcal{L}_\theta(\{c_i\}, \mathcal{D}) = \frac{1}{n}\sum_{i=1}^{n} c_i \log\left(\frac{\exp(r_\theta(a_i))}{\exp(r_\theta(a_i)) + \exp(r_\theta(a_i'))}\right) + (1-c_i)\cdot\log\left(\frac{\exp(r_\theta(a_i'))}{\exp(r_\theta(a_i)) + \exp(r_\theta(a_i'))}\right),$$

while r_{θ_k} does not converge **do**

$$\begin{aligned}
\theta_{k+1} &\leftarrow \theta_k - \lambda \cdot \nabla \mathcal{L}_\theta(\{c_{i,k}\}, \mathcal{D}) \\
c_{i,k+1} &\leftarrow \alpha \cdot c_{i,k} + (1-\alpha) \cdot \frac{\exp(r_\theta(a_i))}{\exp(r_\theta(a_i)) + \exp(r_\theta(a_i'))} \\
k &\leftarrow k+1
\end{aligned}$$

end while
Return: θ_k

As is shown in Algorithm 1, in each epoch, we first update the model using the current comparison dataset. After the model is updated, we also use the model to update the data by predicting the

probability of $\mathbb{P}(c_i = 1)$ for each comparison (a_i, a_i') using the current reward estimate $\hat{r}_{\theta_k}$. We update each label $c_{i,k}$ by weighting its previous value and the new predicted probability.

We first show the following theorem on one-step gradient update of the reward model.

Theorem 4. *Assume that the reward is initialized equally for all K arms. Then after one-step gradient descent, one has*

$$\forall a, a' \in [K], r(a) - r(a') = \lambda \cdot (n_+(a) - n_-(a) - (n_+(a') - n_-(a'))),$$

where $n_+(a), n_-(a)$ refers to the total number of times that a is preferred and not preferred.

The proof is deferred to Appendix C. The result shows that after one-step gradient, the empirical best arm becomes the Condorcet winner. When the arm is only compared few times, the difference $n_+(a') - n_-(a')$ will be bounded by the total number of comparisons, which will be smaller than those that have been compared more times. Thus the model will assign less probability for those arms seen less. After updating the label with the model prediction, the label of less seen samples will be closer to 0, thus getting implicitly penalized.

The data refinement algorithm enjoys several benefits:

- It is easy to combine with neural networks, allowing arbitrary parametrization of the reward model.
- It utilizes the soft labels starting from the second epoch, which has been validated to be more effective than hard labels in the literature of knowledge distillation (Hinton et al., 2015).
- As we show in the Theorem above, when only updated for one step, the reward model is resistant to the noise introduced by unbalanced comparison dataset. Thus, the reweighting of the data help mitigates the issue of over-fitting and over-optimization.

4 Experimental Analysis

In this section, we conduct experiments on both synthetic dataset in the multi-armed bandit setting and real world dataset with neural network parameterized reward family.

4.1 Multi-armed bandit setting

In the bandit setting, we focus on the hard example constructed in Theorem 2. We take total samples $n = 60$ and the number of arms K as 10 and 20. We compare the performance of vanilla MLE, pessimistic MLE and data refinement in both reward learning phase and policy learning phase.

As is shown in left part of Figure 2, in the reward learning phase, both MLE and pessimistic MLE suffers from overfitting, while data refinement algorithm continues to decrease the ground-truth cross entropy loss until convergence. In the right part, we replicate the policy in Equation (3) and plot the tradeoff between the ground truth reward and the KL divergence between the current policy and a uniform policy as an initial policy π_0. One can see that data refinement is able to converge to the optimal reward when KL is large, while both MLE and pessimistic MLE suffer from over-optimization.

We remark here that the reason that pessimistic MLE suffers from both overfitting and over-optimization is due to the design of unbounded reward in the multi-armed bandit case. When the reward family is bounded, pessimistic MLE is also guarantees to mitigate the over-optimization issue.

4.2 Neural Network

We also conduct experiments with neural network in the real RLHF setting. We use the human-labeled Helpfulness and Harmlessnes (HH) dataset from Bai et al. (2022).[1]. We take

[1] https://huggingface.co/datasets/Dahoas/static-hh

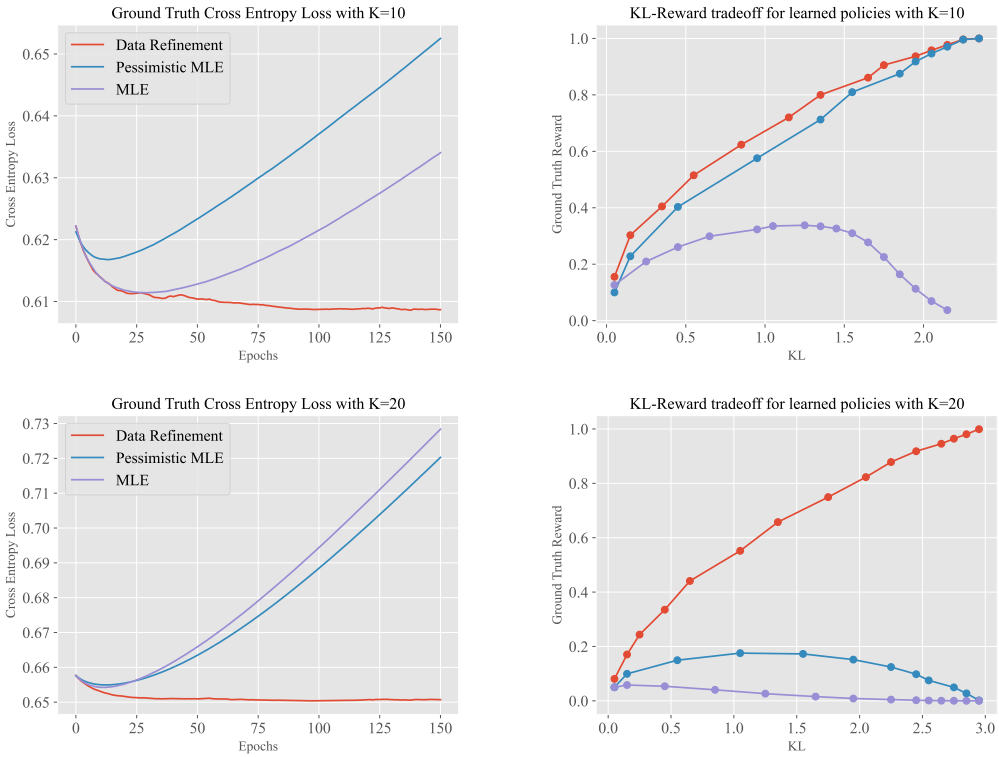

Figure 2: Comparisons of the three methods in the multi-armed bandit setting.

`Dahoas/pythia-125M-static-sft`[2] as policy model with three different reward models of size 125M, 1B and 3B. When training reward model, we take a supervised fine-tuned language model, remove the last layer and replace with a linear layer.

We take a trained 6B reward model as the ground truth. We use the 6B reward model to label the comparisons samples. And we train the 125M, 1B and 3B reward model with the new labeled comparison samples. The reward training results are shown in Figure 3. One can see that the MLE begins to overfit after 1-2 epochs, while data refinement continues to grow stably until convergence. The policy learning result is shown in Figure 4. One can see that MLE also suffers from reward-overoptimization with few steps, while the ground truth reward continues to grow when using our data refinement algorithm.

5 Conclusion and Future Work

In this paper, we provide analysis and solution towards the problem of overfitting and over-optimization in reward training for RLHF. We show that the data refinement algorithm helps mitigate the two issues for both multi-armed bandit and neural network simulated experiments. As part of the future work, we are excited to see formal theoretical analysis for the data refinement algorithm, along with its potential applications beyond reward training in the generic domain of classification and prediction.

[2] `https://huggingface.co/Dahoas/pythia-125M-static-sft`

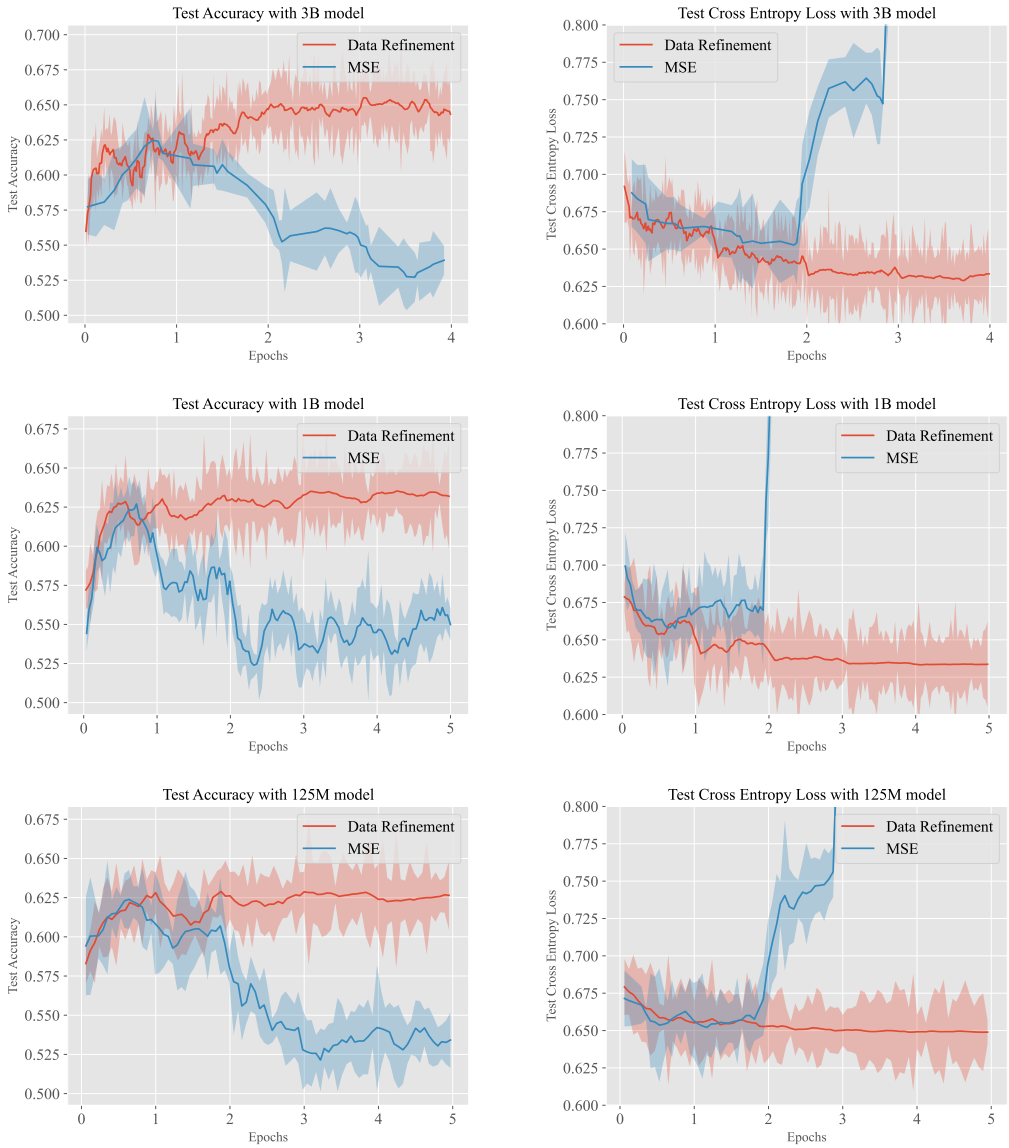

Figure 3: Comparisons of MLE and Data Refinement when the reward is parameterized by a neural network.

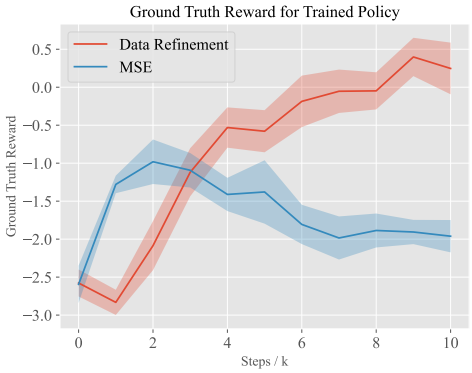

Figure 4: Comparison of MLE and Data Refinement for policy learning

REFERENCES

Nir Ailon, Zohar Shay Karnin, and Thorsten Joachims. Reducing dueling bandits to cardinal bandits. In *ICML*, volume 32, pp. 856–864, 2014.

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

Daniel Brown, Wonjoon Goo, Prabhat Nagarajan, and Scott Niekum. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International Conference on Machine Learning*, pp. 783–792. PMLR, 2019.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Niladri S. Chatterji, Aldo Pacchiano, Peter L. Bartlett, and Michael I. Jordan. On the theory of reinforcement learning with once-per-episode feedback, 2022.

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

Xu Cheng, Zhefan Rao, Yilan Chen, and Quanshi Zhang. Explaining knowledge distillation by quantifying the knowledge. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12925–12935, 2020.

Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4794–4802, 2019.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

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

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pp. 4299–4307, 2017b.

Louis Faury, Marc Abeille, Clément Calauzènes, and Olivier Fercoq. Improved optimistic algorithms for logistic bandits. In *International Conference on Machine Learning*, pp. 3052–3060. PMLR, 2020.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, pp. 1607–1616. PMLR, 2018.

Pratik Gajane, Tanguy Urvoy, and Fabrice Clérot. A relative exponential weighing algorithm for adversarial utility-based dueling bandits. In *Proceedings of the 32nd International Conference on Machine Learning*, pp. 218–227, 2015.

Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*, 2022.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. *arXiv preprint arXiv:2210.10760*, 2022.

Suprovat Ghoshal and Aadirupa Saha. Exploiting correlation to achieve faster learning rates in low-rank preference bandits. In *International Conference on Artificial Intelligence and Statistics*, pp. 456–482. PMLR, 2022.

Amelia Glaese, Nat McAleese, Maja Trębacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Maribeth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, et al. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*, 2022.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Zehao Huang and Naiyan Wang. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219*, 2017.

Ashesh Jain, Brian Wojcik, Thorsten Joachims, and Ashutosh Saxena. Learning trajectory preferences for manipulators via iterative improvement. In *Advances in neural information processing systems*, pp. 575–583, 2013.

W Bradley Knox and Peter Stone. Tamer: Training an agent manually via evaluative reinforcement. In *7th IEEE International Conference on Development and Learning*, pp. 292–297. IEEE, 2008.

Junpei Komiyama, Junya Honda, Hisashi Kashima, and Hiroshi Nakagawa. Regret lower bound and optimal algorithm in dueling bandit problem. In *COLT*, pp. 1141–1154, 2015.

Andras Kupcsik, David Hsu, and Wee Sun Lee. Learning dynamic robot-to-human object handover from human feedback. In *Robotics research*, pp. 161–176. Springer, 2018.

James MacGlashan, Mark K Ho, Robert Loftin, Bei Peng, Guan Wang, David L Roberts, Matthew E Taylor, and Michael L Littman. Interactive learning from policy-dependent human feedback. In *International Conference on Machine Learning*, pp. 2285–2294. PMLR, 2017.

Jacob Menick, Maja Trebacz, Vladimir Mikulik, John Aslanides, Francis Song, Martin Chadwick, Mia Glaese, Susannah Young, Lucy Campbell-Gillingham, Geoffrey Irving, et al. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*, 2022.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Ellen R Novoseller, Yanan Sui, Yisong Yue, and Joel W Burdick. Dueling posterior sampling for preference-based reinforcement learning. *arXiv preprint arXiv:1908.01289*, 2019.

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

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

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3967–3976, 2019.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*, 2022.

Zengyu Qiu, Xinzhu Ma, Kunlin Yang, Chunya Liu, Jun Hou, Shuai Yi, and Wanli Ouyang. Better teacher better student: Dynamic prior knowledge for knowledge distillation. *arXiv preprint arXiv:2206.06067*, 2022.

Rajkumar Ramamurthy, Prithviraj Ammanabrolu, Kianté Brantley, Jack Hessel, Rafet Sifa, Christian Bauckhage, Hannaneh Hajishirzi, and Yejin Choi. Is reinforcement learning (not) for natural language processing?: Benchmarks, baselines, and building blocks for natural language policy optimization. *arXiv preprint arXiv:2210.01241*, 2022.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

Dorsa Sadigh, Anca D Dragan, Shankar Sastry, and Sanjit A Seshia. Active preference-based learning of reward functions. In *Robotics: Science and Systems*, 2017.

Aadirupa Saha and Aditya Gopalan. Battle of bandits. In *Uncertainty in Artificial Intelligence*, 2018a.

Aadirupa Saha and Aditya Gopalan. Active ranking with subset-wise preferences. *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018b.

Aadirupa Saha and Aditya Gopalan. PAC Battling Bandits in the Plackett-Luce Model. In *Algorithmic Learning Theory*, pp. 700–737, 2019.

Aadirupa Saha and Akshay Krishnamurthy. Efficient and optimal algorithms for contextual dueling bandits under realizability. In *International Conference on Algorithmic Learning Theory*, pp. 968–994. PMLR, 2022.

Daniel Shin, Anca D Dragan, and Daniel S Brown. Benchmarks and algorithms for offline preference-based reward learning. *arXiv preprint arXiv:2301.01392*, 2023.

Charlie Snell, Ilya Kostrikov, Yi Su, Mengjiao Yang, and Sergey Levine. Offline rl for natural language generation with implicit language q learning. *arXiv preprint arXiv:2206.11871*, 2022.

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. Preference ranking optimization for human alignment. *arXiv preprint arXiv:2306.17492*, 2023.

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.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 1365–1374, 2019.

Garrett Warnell, Nicholas Waytowich, Vernon Lawhern, and Peter Stone. Deep tamer: Interactive agent shaping in high-dimensional state spaces. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Christian Wirth, Riad Akrour, Gerhard Neumann, and Johannes Fürnkranz. A survey of preference-based reinforcement learning methods. *The Journal of Machine Learning Research*, 18(1):4945–4990, 2017.

Jeff Wu, Long Ouyang, Daniel M Ziegler, Nisan Stiennon, Ryan Lowe, Jan Leike, and Paul Christiano. Recursively summarizing books with human feedback. *arXiv preprint arXiv:2109.10862*, 2021.

Yichong Xu, Ruosong Wang, Lin Yang, Aarti Singh, and Artur Dubrawski. Preference-based reinforcement learning with finite-time guarantees. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 18784–18794. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/d9d3837ee7981e8c064774da6cdd98bf-Paper.pdf.

Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4133–4141, 2017.

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

Yisong Yue and Thorsten Joachims. Interactively optimizing information retrieval systems as a dueling bandits problem. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1201–1208. ACM, 2009.

Yisong Yue and Thorsten Joachims. Beat the mean bandit. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 241–248, 2011.

Yisong Yue, Josef Broder, Robert Kleinberg, and Thorsten Joachims. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 11953–11962, 2022.

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*, 2023a.

Banghua Zhu, Hiteshi Sharma, Felipe Vieira Frujeri, Shi Dong, Chenguang Zhu, Michael I Jordan, and Jiantao Jiao. Fine-tuning language models with advantage-induced policy alignment. *arXiv preprint arXiv:2306.02231*, 2023b.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

Masrour Zoghi, Shimon Whiteson, Remi Munos, Maarten de Rijke, et al. Relative upper confidence bound for the k-armed dueling bandit problem. In *JMLR Workshop and Conference Proceedings*, number 32, pp. 10–18. JMLR, 2014a.

Masrour Zoghi, Shimon A Whiteson, Maarten De Rijke, and Remi Munos. Relative confidence sampling for efficient on-line ranker evaluation. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pp. 73–82. ACM, 2014b.

A Proof of Theorem 2

The construction is in similar spirit to Zhu et al. (2023a). Consider a bandit problem where $r^\star(a) = \mathbb{1}(a = 1)$. For any fixed n, we set $\mu(a_1, a_2) = 1 - (K-2)/n$, $\mu(a_1, a_k) = 1/n$ for any $2 \le k \le K$ and the rest pairs with 0.

In this hard instance, for each arm $a \in \{2, 3, \cdots, K\}$, there is constant probability that a is only compared with 1 once. And with constant probability, the comparison between arm 1 and arm a will be flipped. To see this, consider the following event

$$\mathcal{E}_1 \coloneqq \{n(2) = 1\}.$$

We have

$$\mathbb{P}(\mathcal{E}_1) = n \cdot \mu(1)^{n-1} \cdot \mu(2) = (1 - 1/n)^{n-1}.$$

As long as n is sufficiently large (say $n \ge 500$), we have $\mathbb{P}(\mathcal{E}_1) \ge 0.36$.

Under this case, we know that arm 2 is preferred with probability at least $\exp(r(2))/(\exp(r(1)) + \exp(r(2))) > 0.36$. When there is only one comparison between arm 1 and 2, and arm 2 is preferred, the MLE assigns ∞ to arm 2, leading to arbitrarily large reward. This finishes the proof.

B Proof of Theorem 3

The proof exactly follows that of Theorem 2. Under the same construction, we know that $\hat{r}_{\mathsf{MLE}}(2) = +\infty$ with probability at least 0.1. Thus, the sub-optimality of the resulting optimal policy is at least 1.

C Proof of Theorem 4

One can calculate the reward gradient as

$$\begin{aligned}
\nabla_{\hat{r}_i} \mathcal{L}_{\mathsf{CE}}(\mathcal{D}, \hat{r}) &= \frac{1}{n} \sum_{i=1}^{n} \left(\mathbb{1}(c_i = 1) \log \left(\frac{\exp(\hat{r}(a_i))}{\exp(\hat{r}(a_i)) + \exp(\hat{r}(a_i'))} \right) + \mathbb{1}(c_i = 0) \log \left(\frac{\exp(\hat{r}(a_i'))}{\exp(\hat{r}(a_i)) + \exp(\hat{r}(a_i'))} \right) \right) \\
&= \frac{1}{n} \sum_{i=1}^{n} \left(\mathbb{1}(c_i = 1) \left(\frac{\exp(\hat{r}(a_i'))}{\exp(\hat{r}(a_i)) + \exp(\hat{r}(a_i'))} \right) - \mathbb{1}(c_i = 0) \left(\frac{\exp(\hat{r}(a_i))}{\exp(\hat{r}(a_i)) + \exp(\hat{r}(a_i'))} \right) \right) \\
&= \frac{1}{2} \cdot (n_+(i) - n_-(i)).
\end{aligned}$$

Here the last equality is due to that all the reward is initialized at the same value.

We assume all the reward is initialized at 0 without loss of generality. After one step gradient, we have

$$\hat{r}(i) = \lambda(n_+(i) - n_-(i)).$$

This proves the result.