Iterative Data Smoothing: Mitigating Reward Overfitting and Overoptimization in RLHF

Banghua Zhu [1] **Michael I. Jordan** [1] **Jiantao Jiao** [1]

Abstract

Reinforcement Learning from 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 ranking data. It is observed that the performance of the reward model degrades after one epoch of training, and optimizing too much against the learned reward model eventually hinders the true objective. This paper analyzes potential reasons behind the issues, and designs improved reward learning algorithm termed 'Iterative Data Smoothing' (IDS). The core idea is that during each training epoch, we not only update the model with the data, but also update the date using the model, replacing hard labels with soft labels. Our empirical findings highlight the superior performance of this approach over the traditional methods.

1. Introduction

Recent progress on Large Language Models (LLMs) is having a transformative effect not only in natural language processing but also more broadly in a range of AI applications (Radford et al., 2019; Chowdhery et al., 2022; Brown et al., 2020; Touvron et al., 2023; Bubeck et al., 2023; Schulman et al., 2022; OpenAI, 2023; Anthropic, 2023). A key ingredient in the roll-out of LLMs is the fine-tuning step, in which the models are brought into closer alignment with specific behavioral and normative goals. When no adequately fine-tuned, LLMs may exhibit undesirable and unpredictable behavior, including the fabrication of facts or the generation of biased and toxic content (Perez et al., 2022; Ganguli et al., 2022). The current approach towards mitigating such problems is to make use of reinforcement learning based on human assessments. In particular, Reinforcement Learning with Human Feedback (RLHF) proposes to use human assessments as a reward function from pairwise or multi-wise comparisons of model responses, and then fine-tune the language model based on the learned reward functions (Ziegler et al., 2019; Ouyang et al., 2022; Schulman et al., 2022).

Following on from a supervised learning stage, a typical RLHF protocol involves two main steps:

- **Reward learning:** Sample prompts from a prompt dataset and generate multiple responses for the same prompt. Collect human preference data in the form of pairwise or multi-wise comparisons of different responses. Train a reward model based on the preference data.
- **Policy learning:** Fine-tune the current LLM based on the learned reward model with reinforcement learning algorithms.

Although RLHF has been successful in practice (Bai et al., 2022; Ouyang et al., 2022; Dubois et al., 2023), it is not without flaws, and indeed the current reward training paradigm grapples with significant value-reward mismatches. There are two major issues with the current paradigm:

- **Reward overfitting:** During the training of the reward model, it has been observed that the test cross-entropy loss of the reward model can deteriorate after one epoch of training (Ouyang et al., 2022).
- **Reward overoptimization:** When training the policy model to maximize the reward predicted by the learned model, it has been observed that the ground-truth reward can increase when the policy is close in KL divergence to the initial policy, but decrease with continued training (Gao et al., 2023).

In this paper, we investigate these issues in depth. We simplify the formulation of RLHF to a multi-armed bandit problem and make attempt to explain and reproduce the overfitting and overoptimization phenomena. We leverage

[1] Department of EECS, University of California, Berkeley. Correspondence to: Banghua Zhu <banghua@berkeley.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

theoretical insights in the bandit setting to design new algorithms that help mitigate the issues and work well under practical fine-tuning scenarios.

1.1. Main Results

As our first contribution, we make the attempt to explain and analyze both reward overfitting and overoptimization from the simple setting of multi-armed bandit. We show that the **inadequacy of the cross-entropy loss for long-tailed preference datasets** can be one of the reasons for both overfitting and overoptimization. As illustrated in Figure 1, even a simple 3-armed bandit problem can succumb to overfitting and overoptimization when faced with such imbalanced datasets. Consider a scenario where we have three arms with true rewards given by $r_1^\star = 1, r_2^\star = r_3^\star = 0$, and the preference distribution is generated by the Bradley-Terry-Luce (BTL) model (Bradley & Terry, 1952), i.e. $\mathbb{P}(i \succ j) = \exp(r_i^\star)/(\exp(r_i^\star) + \exp(r_j^\star))$. Suppose our preference dataset compares the first and second arms 1000 times but only compares the first and third arm once, and let $n(i \succ j)$ denote the number of times that arm i is preferred over arm j. The standar empirical cross-entropy loss used in the literature for learning the reward model (Ouyang et al., 2022; Zhu et al., 2023a) can be written as follows:

$$-\sum_{i,j} n(i \succ j) \log \left(\frac{\exp(r_i)}{\exp(r_i) + \exp(r_j)} \right).$$

We know that the empirical values $n(1 \succ 2)$ and $n(2 \succ 1)$ concentrate around their means. However, we have with probability 0.73, $n(1 \succ 3) = 1$ and $n(3 \succ 1) = 0$, and with probability 0.27, $n(1 \succ 3) = 0$ and $n(3 \succ 1) = 1$. In either case, the minimizer of the empirical entropy loss will satisfy either $\hat{r}_1 - \hat{r}_3 = -\infty$ or $\hat{r}_1 - \hat{r}_3 = +\infty$. This introduces a huge effective noise when the coverage is imbalanced. Moreover, the limiting preference distribution is very different from the ground truth distribution, leading to reward overfitting. Furthermore, since there is 0.27 probability that $\hat{r}_1 - \hat{r}_3 = -\infty$, we will take arm 3 as the optimal arm instead of arm 1. This causes reward overoptimization during the stage of policy learning since the final policy converges to the wrong arm with reward zero.

To mitigate these effects, we leverage the pessimism mechanism from bandit learning to analyze and design a new algorithm, Iterative Data Smoothing (IDS), that simultaneously addresses both reward overfitting and reward overoptimization. The algorithm design is straightforward: in each epoch, beyond updating the model with the data, we also adjust the data using the model. Theoretically, we investigate the two phenomena in the tabular bandit case. We show that the proposed method, as an alternative to the lower-confidence-bound-based algorithm (Jin et al., 2021; Xie et al., 2021; Rashidinejad et al., 2021; Zhu et al., 2023a), learns the ground truth distribution for pairs that are compared enough times, and ignores infrequently covered comparisons thereby mitigating issues introduced by long-tailed data. Empirically, we present experimental evidence that the proposed method improves reward training in both bandit and neural network settings.

1.2. Related Work

RLHF and Preference-based Reinforcement Learning. RLHF, or Preference-based Reinforcement Learning (PbRL), has delivered significant empirical success in the fields of game playing, robot training, stock prediction, recommender systems, clinical trials and natural language processing (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). In the setting of the language models, there has been work exploring the efficient fine-tuning of the policy model (Snell et al., 2022; Song et al., 2023a; Yuan et al., 2023; Zhu et al., 2023b; Rafailov et al., 2023; Wu et al., 2023).

In the case of reward learning, Ouyang et al. (2022) notes that in general the reward can only be trained for one epoch in the RLHF pipeline, after which the test loss can go up. Gao et al. (2023) studies the scaling law of training the reward model, and notes that overoptimization is another problem in reward learning. To address the problem, Zhu et al. (2023a) propose a pessimism-based method that improves the policy trained from the reward model when the optimal reward lies in a linear family. It is observed in Song et al. (2023b) that the reward model tends to be identical regardless of whether the prompts are open-ended or closed-ended during the terminal phase of training, and they propose a prompt-dependent reward optimization scheme.

Another closely related topic is the problem of estimation and ranking from pairwise or K-wise comparisons. 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 agents in the tabular case, which is extended

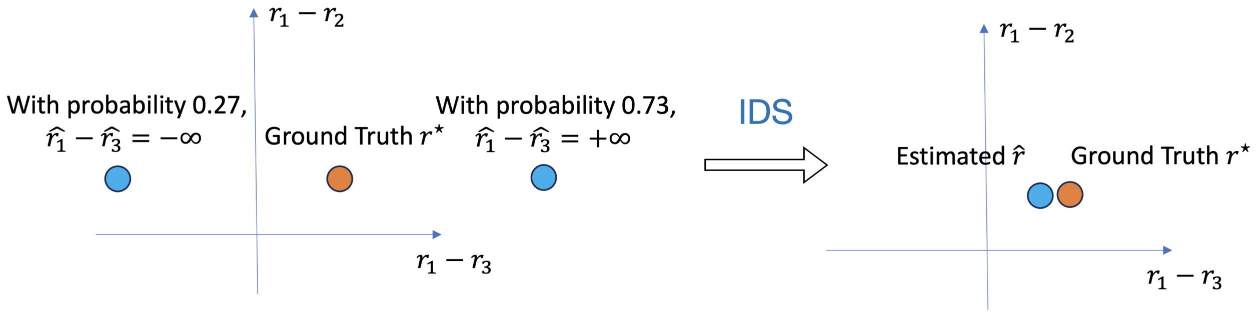

Figure 1: Illustration of the problem of the vanilla empirical cross-entropy minimization for learning the ground truth reward. With a small number of samples comparing arm 1 and 3, the minimization converges to a solution which assigns $\hat{r}_1 - \hat{r}_3 = -\infty$ with constant probability. With the proposed Iterative Data Smoothing (IDS) algorithm, the estimator is able to recover the ground truth reward.

to linear case and function approximation by the recent work of Pacchiano et al. (2021); Chen et al. (2022). Chatterji et al. (2022) studies a related setting where in each episode only binary feedback is received. Most of the theoretical work of learning from ranking focuses on regret minimization, while RLHF focuses more on the quality of the final policy.

Knowledge Distillation The literature of knowledge distillation focuses on transferring the knowledge from a teacher model to a student model (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). It is observed in this literature that the soft labels produced by the teacher network can help train a better student network, even when the teacher and student network are of the same size and structure (Hinton et al., 2015). Furlanello et al. (2018) present a method which iteratively trains a new student network after the teacher network achieves the smallest evaluation loss. Both our iterative data smoothing algorithm and these knowledge distillation methods learn from soft labels. However, iterative data smoothing iteratively updates the same 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 $\mu(a, a')$ to denote the probability of comparing a and a' in a preference dataset, and $\mu(a) = \sum_{a' \in \mathcal{A}} \mu(a, a')$ to denote the probability of comparing a with any other arms. 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. We use $a_1 \succ a_2$ to denote the event that the a_1 is more preferred compared to a_2.

2.1. General Formulation of RLHF

The key components in RLHF consist of two steps: reward learning and policy learning. We briefly introduce the general formulation of RLHF below.

In the stage of reward learning, one collects a preference dataset based on a prompt dataset and responses to the prompts. According to the formulation of Zhu et al. (2023a), for the i-th sample, a state (prompt) s_i is first sampled from some prompt distribution ρ. Given the state s_i, M actions (responses) $(a_i^{(1)}, a_i^{(2)}, \cdots, a_i^{(M)})$ are sampled from some joint distribution $\mathbb{P}(a^{(1)}, \cdots, a^{(M)} \mid s_i)$, Let $\sigma_i : [M] \mapsto [M]$ be the output of the human labeller, which is a permutation function that denotes the ranking of the actions. Here $\sigma_i(1)$ represents the most preferred action, and $\sigma_i(M)$ is the least preferred action. A common model for the distribution of σ under multi-ary comparisons is a Plackett-Luce model (Plackett, 1975; Luce, 2012). The Plackett-Luce model defines the probability of a state-action pair (s, a_i) being the largest among a given set $\{(s, a_i)\}_{i=1}^M$ as

$$\mathbb{P}(a_i \succ a_j, \forall j \neq i \mid s) = \frac{\exp(r^\star(s, a_i))}{\sum_{j=1}^M \exp(r^\star(s, a_j))},$$

where $r^\star : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$ is the ground-truth reward for the response given the prompt. Moreover, the probability of observing the permutation σ is defined as[1]

$$\mathbb{P}(\sigma \mid s, \{a^{(i)}\}_{i=1}^{M}) = \prod_{i=1}^{M} \frac{\exp(r^\star(s, a^{(\sigma(i))}))}{\sum_{j=i}^{M} \exp(r^\star(s, a^{(\sigma(j))}))}.$$

When $M = 2$, this reduces to the pairwise comparison considered in the Bradley-Terry-Luce (BTL) model (Bradley & Terry, 1952), which is used in existing RLHF algorithms. In this case, the permutation σ can be reduced to a Bernoulli random variable, representing whether one action is preferred compared to the other. Concretely, for each queried state-actions pair (s, a, a'), we observe a sample c from a Bernoulli distribution with parameter $\frac{\exp(r^\star(s,a))}{\exp(r^\star(s,a))+\exp(r^\star(s,a'))}$. Based on the observed dataset, the cross-entropy loss is minimized to estimate the ground-truth reward for the case of pairwise comparison. The minimizer of cross-entropy loss is the maximum likelihood estimator:

$$\hat{r}_{\mathsf{MLE}} \in \arg\min_{r} \mathcal{L}_{\mathsf{CE}}(\mathcal{D}, r),$$

$$\mathcal{L}_{\mathsf{CE}}(\mathcal{D}, r) = -\sum_{i=1}^{n} \log \left(\frac{y_i \cdot \exp(r(s_i, a_i^{(1)}))}{\exp(r(s_i, a_i^{(1)})) + \exp(r(s_i, a_i^{(2)}))} \right. \tag{1}$$

$$\left. + \frac{(1 - y_i) \cdot \exp(r(s_i, a_i^{(2)}))}{\exp(r(s_i, a_i^{(1)})) + \exp(r(s_i, a_i^{(2)}))} \right).$$

After learning the reward, we aim to learn the optimal policy under KL regularization with respect to an initial policy π_0 under some state (prompt) distribution ρ'.

$$\hat{\pi} = \arg\max_{\pi} \mathbb{E}_{s \sim \rho', a \sim \pi}[\hat{r}_{\mathsf{MLE}}(s, a)] - \lambda \cdot \mathbb{E}_{s \sim \rho'}[\mathsf{KL}(\pi(\cdot \mid s) \| \pi_0(\cdot \mid s))].$$

2.2. RLHF in Multi-Armed Bandits

To understand the overfitting and overoptimization problems, we simplify the RLHF problem to consider a single-state multi-armed bandit formulation with pairwise comparisons. Instead of fitting a reward model and policy model with neural networks, we fit a tabular reward model in a K-armed bandit problem.

Consider a multi-armed bandit problem with K arms. Each arm has a deterministic ground-truth reward $r^\star(k) \in \mathbb{R}, k \in [K]$. In this case, the policy becomes a distribution supported on the K arms $\pi \in \Delta([K])$. The sampling process for general RLHF reduces to 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 y_i following a distribution

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

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 $\mathcal{D} = \{a_i, a_i', y_i\}_{i=1}^{n}$. The tasks in RLHF for multi-armed bandit setting can be simplified as:

1. **Reward learning:** Estimate true reward $r^\star$ with a proxy reward $\hat{r}$ from the comparison dataset $\mathcal{D}$.

2. **Policy learning:** Find a policy $\pi \in \Delta([K])$ that maximizes 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 reasons behind overfitting and overoptimization.

2.3. Overfitting in Reward Learning

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

$$\hat{r}_{\mathsf{MLE}} = \arg\min_{r} \hat{\mathcal{L}}_{\mathsf{CE}}(\mathcal{D}, r), \text{ where} \tag{2}$$

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

By definition, $\hat{r}_{\mathsf{MLE}}$ is convergent point when we optimize the empirical cross entropy fully. Thus the population cross-entropy loss of $\hat{r}_{\mathsf{MLE}}$ is an indicator for whether overfitting exists during reward training.

We define the population cross entropy loss $\mathcal{L}_{\mathsf{CE}}(r)$ as

$$-\mathbb{E}_{(a,a') \sim \mu} \left[\frac{\exp(r^\star(a))}{\exp(r^\star(a)) + \exp(r^\star(a'))} \log \left(\frac{\exp(r(a))}{\exp(r(a)) + \exp(r(a'))} \right) + \frac{\exp(r^\star(a'))}{\exp(r^\star(a)) + \exp(r^\star(a'))} \log \left(\frac{\exp(r(a'))}{\exp(r(a)) + \exp(\hat{r}(a'))} \right) \right]. \tag{3}$$

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

[1] In practice, one may introduce an extra temperature parameter σ and replace all $r^\star$ with $r^\star/\sigma$. Here we take $\sigma = 1$.

Theorem 2.1 (Consistency of MLE, see, e.g., Theorem 6.1.3. of Hogg et al. (2013))**.** *Fix $r^\star(K) = \hat{r}(K) = 0$ for the uniqueness of the solution. For any fixed μ, and any given ground-truth reward $r^\star$, we have that $\hat{r}_{\mathsf{MLE}}$ converges in probability to $r^\star$; i.e., for any $\epsilon > 0$,*

$$\lim_{n \to +\infty} \mathbb{P}\left(\|\hat{r}_{\mathsf{MLE}} - r^\star\|_\infty \geq \epsilon \right) = 0.$$

Here we view $\hat{r}_{\mathsf{MLE}}$ and $r^\star$ as K-dimensional vectors.

The proof is deferred to Appendix D. This suggests that the overfitting phenomenon does not arise when we have an 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 2.2 (Reward overfitting of MLE in the non-asymptotic regime)**.** *Fix $r^\star(a) = \mathbb{1}(a = 1)$ and $\hat{r}(K) = 0$ for uniqueness of the solution. For any fixed $n > 500$, there exists some 3-armed bandit problem such that with probability at least 0.09,*

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

for any arbitrarily large C.

The proof is deferred to Appendix E. 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 $\mu(1, 2) = 1 - 1/n$, $\mu(1, 3) = 1/n$.

In this hard instance, there is constant probability that arm 3 is only compared with 1 once. And with constant probability, the observed comparison result between arm 1 and arm 3 will be different from the ground truth. The MLE will assign $r(3) = +\infty$ since the maximizer of $\log(\exp(x)/(1 + \exp(x)))$ is infinity when x is not bounded. Thus when optimizing the empirical cross entropy fully, the maximum likelihood estimator will result in a large population cross-entropy loss. We also validate this phenomenon in Section C.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. One can also extend the lower bound to more than 3 arms, where the probability of the loss being arbitrarily large will be increased to close to 1 instead of a small constant.

2.4. Overoptimization in Policy Learning

After obtaining 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 (reference) policy π_0, and optimizes 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, exhibiting the reward overoptimization phenomenon.

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

$$\max_{\pi \in \Delta([K])} \mathbb{E}_{a \sim \pi(\cdot)}[\hat{r}(a)] - \frac{1}{\lambda} \cdot \mathsf{KL}(\pi \| \pi_0). \tag{4}$$

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

$$\pi_\lambda(a) = \frac{\pi_0(a) \cdot \exp(\lambda \cdot \hat{r}(a))}{\sum_{a' \in \mathcal{A}} \pi_0(a') \cdot \exp(\lambda \cdot \hat{r}(a'))}. \tag{5}$$

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 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 of how the mismatch between $\hat{r}$ and $r^\star$ leads to the overoptimization phenomenon, which can be validated from the experiments in Section C. To simplify the analysis and provide better intuition, we focus on the case when $\lambda \to \infty$, i.e., when the optimal policy selects the best empirical arm without considering the KL constraint. 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 keep loosening the KL divergence constraint in Equation (4). Thus the performance of π_∞ is a good indicator of whether overoptimization exists during policy training. We thus define a notion fo sub-optimality to characterize the performance gap between the convergent policy and the optimal policy:

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

We know from Theorem 2.1 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 zero with an infinite number of samples.

However, as a corollary of Theorem 2.2 and a direct consequence of reward overfitting, π_∞ may have large sub-optimality in the non-asymptotic regime when trained from $\hat{r}_{\mathsf{MLE}}$.

Corollary 2.3 (Reward overoptimization of MLE in the non-asymptotic regime)**.** *Fix $r^\star(a) = \mathbb{1}(a = 1)$. For any fixed n, there exists some 3-armed bandit problem such that with probability at least 0.09,*

$$\mathsf{SubOpt}(\hat{\pi}_\infty) \geq 1.$$

The proof is deferred to Appendix F. This suggests that $\hat{r}_{\mathsf{MLE}}$ also leads to the reward overoptimization phenomenon in the non-asymptotic regime. In Section C, we conduct simulation in the exact same setting to verify the theoretical results.

3. Methods: Pessimistic MLE and Iterative Data Smoothing

The problem of overfitting and overoptimization 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 converge to a policy with vanishing sub-optimality 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

$$\hat{r}_{\mathsf{PE}}(a) = \hat{r}_{\mathsf{MLE}}(a) - \lambda \cdot \sqrt{\frac{1}{n}}, \tag{6}$$

where n is the total number of samples and $\lambda = \|(L + \epsilon I)_j^{-1/2}\|_2$ is the norm of the j-th column of the matrix $(L + \epsilon I)^{-1/2}$, where L is the matrix that satisfies $L_{a,a} = n(a)/n$, $L_{a,a'} = -n(a,a')/n, \forall a \neq a'$, and ϵ is a small constant. Intuitively, for those arms that are compared fewer times, we are more uncertain about their ground-truth reward value. Pessimistic MLE penalizes these arms by directly subtracting the length of lower confidence interval of their reward, ensuring that the arms that are less often compared will be less likely to be chosen. It is shown in Zhu et al. (2023a) that the sub-optimality of the policy optimizing $\hat{r}_{\mathsf{PE}}$ converges to zero under the following two conditions:

- The expected number of times that one compares optimal arm (or the expert arm to be compared with in the definition of sub-optimality) is lower bounded by some positive constant $\mu(a^\star) \geq C$.
- The parameterized reward family lies in a bounded space $|\hat{r}(a)| \leq B, \forall a \in [K]$.

This indicates that pessimistic MLE can help mitigate the reward overoptimization phenomenon. However, for real-world reward training paradigm, the neural network parameterized reward family may not be bounded. Furthermore, estimating the exact confidence interval for a neural-network parameterized model can be hard and costly. This prevents the practical use of pessimistic MLE, and calls for new methods that can potentially go beyond these conditions and apply to neural networks.

3.2. Iterative Data Smoothing

We propose a new algorithm, Iterative Data Smoothing (IDS), that shares similar insights as pessimistic MLE. Intuitively, pessimistic MLE helps mitigate the reward overoptimization issue by reducing the estimated reward for less seen arms. In IDS, we achieve this by updating the label of the data we train on.

As is shown in Algorithm 1, we initialize $y_{i,0}$ as the labels for the samples y_i. In the t-th epoch, we first update the model using the current comparison dataset with labels $\{y_{i,t}\}_{i=1}^n$. After the model is updated, we also update the data using the model by predicting the probability $\mathbb{P}(y_i = 1)$ for each comparison (a_i, a_i') using the current reward estimate $\hat{r}_{\theta_t}$. We update each label $y_{i,t}$ as a weighted combination of its previous value and the new predicted probability.

Intuitively, $y_{i,t}$ represents a proxy of the confidence level of labels predicted by interim model checkpoints. The idea is that as the model progresses through multiple epochs of training, it will bring larger change to rewards for frequently observed samples whose representation is covered well in the dataset. Meanwhile, for seldom-seen samples, the model will make minimal adjustments to the reward.

3.2.1. Benefit of one-step gradient descent

Before we analyze the IDS algorithm, we first discuss why training for one to two epochs in the traditional reward learning approach works well (Ouyang et al., 2022). We provide the following analysis of the one-step gradient update for the reward model. The proof is deferred to Appendix G.

Theorem 3.1. *Consider the same multi-armed bandit setting where the reward is initialized equally for all K arms. Then after one-step gradient descent, one has*

$$\forall a, a' \in [K], \hat{r}(a) - \hat{r}(a') = \\ \alpha \cdot (n_+(a) - n_-(a) - (n_+(a') - n_-(a'))),$$

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

```
**Algorithm 1** Iterative Data Smoothing ($\mathcal{D}, \theta_0, \alpha, \beta$)
  **Input:** The pairwise comparison dataset $\mathcal{D} = \{a_i, a_i', y_i\}_{i=1}^n$. A parameterized reward model family $\{r_\theta : \mathcal{A} \mapsto \mathbb{R} \mid \theta \in \Theta\}$ with initialization $\theta_0 \in \Theta$. Two step sizes $\alpha, \beta$. An empirical loss function
      $\mathcal{L}_\theta(\{y_i\}, \mathcal{D}) = -\frac{1}{n} \sum_{i=1}^n y_i \cdot \log \left( \frac{\exp(r_\theta(a_i))}{\exp(r_\theta(a_i)) + \exp(r_\theta(a_i'))} \right) + (1 - y_i) \cdot \log \left( \frac{\exp(r_\theta(a_i'))}{\exp(r_\theta(a_i)) + \exp(r_\theta(a_i'))} \right)$
  Initialize $t = 0$ and $y_{i,0} = y_i, \forall i \in [n]$.
  **while** $r_{\theta_t}$ does not converge **do**
      $\theta_{t+1} \leftarrow \theta_t - \alpha \cdot \nabla \mathcal{L}_\theta(\{y_{i,t}\}, \mathcal{D})$
      $y_{i,t+1} \leftarrow (1 - \beta) \cdot y_{i,t} + \beta \cdot \frac{\exp(r_{\theta_{t+1}}(a_i))}{\exp(r_{\theta_{t+1}}(a_i)) + \exp(r_{\theta_{t+1}}(a_i'))}$
      $t \leftarrow t + 1$
  **end while**
  **Return:** $r_{\theta_t}$
```

Remark 3.2. The result shows that why early stopping in the traditional reward learning works well in a simple setting. After one gradient step, the empirical best arm becomes the the arm whose absolute winning time is the largest. This can be viewed as another criterion besides pessimism that balances both the time of comparisons and the time of being chosen as the preferred arm. When the arm a 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 much more times. Thus the reward model will penalize those arms seen less. After updating the label with the model prediction, the label of less seen samples will be closer to zero, thus getting implicitly penalized.

3.2.2. Benefit of iterative data smoothing

Due to under-optimization, the estimator from a one-step gradient update might still be far from the ground-truth reward. We provide an analysis here why IDS can be better. Consider any two arms a, a' with $n(a, a')$ observations among n total observations. By computing the gradient, we can write the IDS algorithm as

$$\begin{aligned} \hat{r}_{t+1}(a) - \hat{r}_{t+1}(a') = \hat{r}_t(a) - \hat{r}_t(a') + \frac{\alpha \cdot n(a, a')}{n} \\ \cdot \Big((\hat{\mu}(a \succ a') \cdot y_t + \hat{\mu}(a \prec a') \cdot (1 - y_t)) \cdot \frac{\exp(\hat{r}_t(a'))}{\exp(\hat{r}_t(a)) + \exp(\hat{r}_t(a'))} \\ - (\hat{\mu}(a \prec a') \cdot y_t + \hat{\mu}(a \succ a') \cdot (1 - y_t)) \cdot \frac{\exp(\hat{r}_t(a))}{\exp(\hat{r}_t(a)) + \exp(\hat{r}_t(a'))} \Big) \\ y_{t+1} = (1 - \beta) \cdot y_t + \beta \cdot \frac{\exp(\hat{r}_{t+1}(a))}{\exp(\hat{r}_{t+1}(a)) + \exp(\hat{r}_{t+1}(a'))}, \end{aligned}$$

where we define $\hat{\mu}(a \succ a') = n(a \succ a')/n(a, a')$. One can see that the effective step size for updating $\hat{r}$ is $\alpha \cdot n(a, a')/n$, while the effective step size for updating y is β. Assume that we choose α, β, l, m such that

$$\alpha \cdot l/n \ll \beta \ll \alpha \cdot m/n.$$

Consider the following two scales:

- When there are sufficient observations, $n(a, a') \geq m$, we know that $\beta \ll \alpha \cdot n(a, a')/n$. In this case, the update step size of y_t is much slower than $\hat{r}_t$. One can approximately take $y_t \approx 0$ or 1 as unchanged during the update. Furthermore, since $n(a, a') \geq m$ is large enough, $\hat{\mu}$ concentrates around the ground truth μ. In this case, one can see that the reward converges to the ground truth reward $\hat{r}_t \to r^\star$.
- When the number of observations is not large, i.e., $n(a, a') \leq l$, we know that $\alpha \cdot l/n \ll \beta$. In this case, the update of $\hat{r}$ is much slower than y_t. When the $\hat{r}_0$ are initialized to be zero, y_t will first converge to $1/2$, leading to $\hat{r}_t(a) \approx \hat{r}_t(a')$ when t is large.

To formalize the above argument, we consider the following differential equations:

$$\begin{aligned} \dot{d}(t) = \alpha n \cdot \Bigg((\mu \cdot y(t) + (1 - \mu) \cdot (1 - y(t))) \cdot \frac{1}{1 + \exp(d(t))} \\ - ((1 - \mu) \cdot y(t) + \mu \cdot (1 - y(t))) \cdot \frac{\exp(d(t))}{1 + \exp(d(t))} \Bigg) \\ \dot{y}(t) = \beta \cdot \left(\frac{\exp(d(t))}{1 + \exp(d(t))} - y(t) \right). \end{aligned} \tag{7}$$

Here d represents the difference of reward between two arms a, a', and μ represents the empirical frequency $\hat{\mu}(a \succ a')$.

Let the initialization be $d(0) = 0, y(0) = 1$. We have the following theorem.

Theorem 3.3. *The differential equations in Equation (7) have one unique stationary point $d(t) = 0, y(t) = \frac{1}{2}$. On the other hand, for any α, β, n, T with $\beta T \leq \epsilon \ll 1 \ll \alpha n T$, one has*

$$\left|\frac{\exp(d(T))}{1+\exp(d(T))} - \mu\right| \leq \max(2(1-\exp(-\epsilon)), \exp(-\mu(1-\mu)\alpha n T))$$
$$y(T) \geq \exp(-\epsilon).$$

The proof is deferred to Appendix H. The above argument only proves convergence to the empirical measure μ. One can combine standard concentration argument to prove the convergence to the ground truth probability. The result shows that when choosing α, β carefully, for the pair of arms with a large number of comparisons, the difference of reward will be close to the ground truth during the process of training. As a concrete example, by taking $\alpha = n^{-1/2}, \beta = n^{-1}T^{-2}, \epsilon = \beta T$, we have

$$\left|\frac{\exp(d(T))}{1+\exp(d(T))} - \mu\right|$$
$$\leq \max(2n^{-1}T^{-1}, \exp(-\mu(1-\mu)n^{1/2}T)).$$

For those pairs of comparisons with a large sample size n, the estimated probability is close to the ground truth probability. On the other hand, for those pairs that are compared less often, the difference $d(t)$ is updated less frequently and remains close to the initialized values. Thus the algorithm implicitly penalizes the less frequently seen pairs, while still estimating the commonly seen pairs accurately. We also present an alternative formulation of IDS in Appendix B.

In summary, the IDS algorithm enjoys several benefits:

- For a sufficient number of observations, the estimated reward converges to the ground truth reward; while for an insufficient number of observations, the estimated reward remains largely unchanged at the initialization. Thus the reward model penalizes the less observed arms with higher uncertainty.
- 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 can be more effective than hard labels according to the literature on knowledge distillation (Hinton et al., 2015; Zhao & Zhu, 2023).

3.3. Experiments

Due to space limit, we provide one experimental result in Figure 2, and leave all the details of experiments to Appendix C, where we conduct simulation study on multi-armed bandit environment, and real-world experiments with the human-labeled Helpfulness and Harmlessnes (HH) dataset from Bai et al. (2022) and TLDR dataset[2], along with comparison with more baseline algorithms including Laplace Smoothing (Chen & Goodman, 1999). In Figure 2, we train a reward model with 1 Billion parameters using HH dataset, and fine-tune the language model with the reward model. One can see that after 1 epoch of training, the test loss of MLE begins to increase, while IDS enables continuous decrease of test loss for more than 3 epochs. Furthermore, when tuning the language model with the proxy reward model, the ground-truth reward for IDS grows higher than that of MLE.

4. Conclusions

We have presented analyses and methodology aimed at resolving the problems of overfitting and overoptimization for RLHF. We show that our proposed algorithm, IDS, helps mitigate these issues. While we identify the underlying source of reward overfitting and overoptimization as the variance of the human preference data, it is also possible that bias also contributes to these phenomena. In future work, it is interesting to pursue further theoretical analysis of the IDS algorithm, and explore potential applications beyond reward training in the generic domains of classification and prediction.

Acknowledgement

The authors would like to thank John Schulman for discussions and suggestions throughout the projects, which initiated the idea of reproducing reward overfitting and overoptimization in the bandit setting, and inspired the idea of combining soft labels during training for better training. The authors would also like to thank Lester Mackey for helpful discussions. The work is supported by NSF Cloudbank and the Mathematical Data Science program of the Office of Naval Research under grant number N00014-21-1-2840. Banghua Zhu and Jiantao Jiao were partially supported by NSF Grants IIS-1901252 and CIF-2211209. Michael I. Jordan was partially supported by NSF Grants IIS-1901252.

Impact Statement

Reinforcement Learning from Human Feedback (RLHF) is an important technique to improve the helpfulness and harmlessness of large language models. It tackles several critical social issues concerning the malicious usage

[2] `https://huggingface.co/datasets/CarperAI/openai_summarize_comparisons`

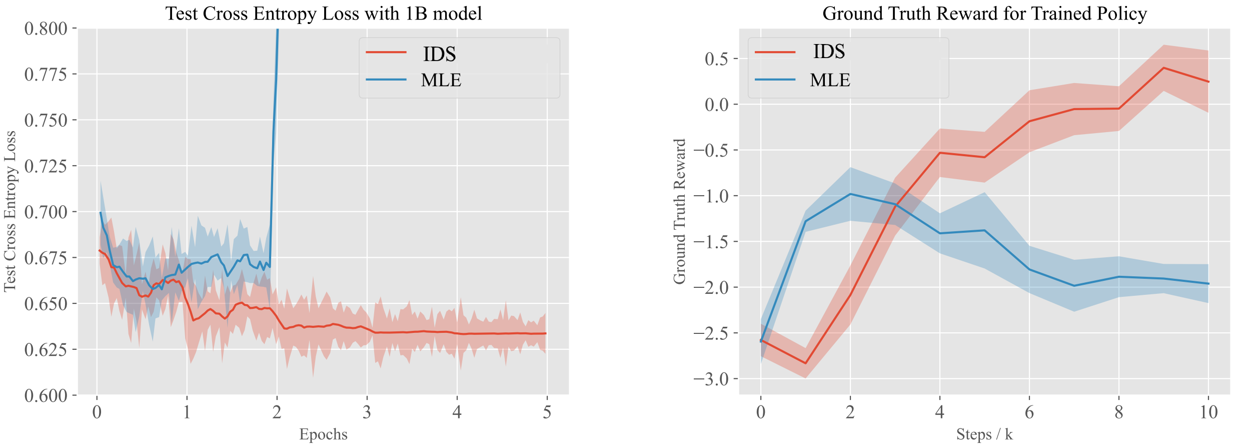

Figure 2: Comparisons of MLE and IDS when the reward is parameterized by a neural network.

of language models such as spreading knowledge that may harm humanity and society, academic misconduct of students, creation of fake news, and circulation of misinformation. By overcoming the overfitting and overoptimization issues in the current RLHF scheme, one can make better use of expensive human preference datasets, training more accurate reward models and less harmful language models. In conclusion, our paper has positive social impacts.

References

Ailon, N., Karnin, Z. S., and Joachims, T. Reducing dueling bandits to cardinal bandits. In *ICML*, volume 32, pp. 856–864, 2014.

Anthropic. Model card and evaluations for claude models, 2023. URL https://www-files.anthropic.com/production/images/Model-Card-Claude-2.pdf. Accessed: Sep. 27,2023.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

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

Brown, D., Goo, W., Nagarajan, P., and Niekum, S. Extrapolating beyond suboptimal demonstrations via inverse reinforcement learning from observations. In *International Conference on Machine Learning*, pp. 783–792. PMLR, 2019.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Chatterji, N. S., Pacchiano, A., Bartlett, P. L., and Jordan, M. I. On the theory of reinforcement learning with once-per-episode feedback, 2022.

Chen, S. F. and Goodman, J. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–394, 1999.

Chen, X., Zhong, H., Yang, Z., Wang, Z., and Wang, L. 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.

Cheng, X., Rao, Z., Chen, Y., and Zhang, Q. Explaining knowledge distillation by quantifying the knowledge. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12925–12935, 2020.

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

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H. W., Sutton, C., Gehrmann, S., et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017a.

Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., and Amodei, D. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pp. 4299–4307, 2017b.

Dubois, Y., Li, X., Taori, R., Zhang, T., Gulrajani, I., Ba, J., Guestrin, C., Liang, P., and Hashimoto, T. B. Alpacafarm: A simulation framework for methods that learn from human feedback. *arXiv preprint arXiv:2305.14387*, 2023.

Faury, L., Abeille, M., Calauzènes, C., and Fercoq, O. Improved optimistic algorithms for logistic bandits. In *International Conference on Machine Learning*, pp. 3052–3060. PMLR, 2020.

Furlanello, T., Lipton, Z., Tschannen, M., Itti, L., and Anandkumar, A. Born again neural networks. In *International Conference on Machine Learning*, pp. 1607–1616. PMLR, 2018.

Gajane, P., Urvoy, T., and Clérot, F. 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.

Ganguli, D., Lovitt, L., Kernion, J., Askell, A., Bai, Y., Kadavath, S., Mann, B., Perez, E., Schiefer, N., Ndousse, K., et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*, 2022.

Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. *arXiv preprint arXiv:2210.10760*, 2022.

Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.

Ghoshal, S. and Saha, A. 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.

Glaese, A., McAleese, N., Trębacz, M., Aslanides, J., Firoiu, V., Ewalds, T., Rauh, M., Weidinger, L., Chadwick, M., Thacker, P., et al. Improving alignment of dialogue agents via targeted human judgements. *arXiv preprint arXiv:2209.14375*, 2022.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hogg, R. V., McKean, J. W., Craig, A. T., et al. *Introduction to mathematical statistics*. Pearson Education India, 2013.

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

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

Jin, Y., Yang, Z., and Wang, Z. Is pessimism provably efficient for offline RL? In *International Conference on Machine Learning*, pp. 5084–5096. PMLR, 2021.

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

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

Kupcsik, A., Hsu, D., and Lee, W. S. Learning dynamic robot-to-human object handover from human feedback. In *Robotics research*, pp. 161–176. Springer, 2018.

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

MacGlashan, J., Ho, M. K., Loftin, R., Peng, B., Wang, G., Roberts, D. L., Taylor, M. E., and Littman, M. L. Interactive learning from policy-dependent human feedback. In *International Conference on Machine Learning*, pp. 2285–2294. PMLR, 2017.

Menick, J., Trebacz, M., Mikulik, V., Aslanides, J., Song, F., Chadwick, M., Glaese, M., Young, S., Campbell-Gillingham, L., Irving, G., et al. Teaching language models to support answers with verified quotes. *arXiv preprint arXiv:2203.11147*, 2022.

Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

Novoseller, E. R., Sui, Y., Yue, Y., and Burdick, J. W. Dueling posterior sampling for preference-based reinforcement learning. *arXiv preprint arXiv:1908.01289*, 2019.

OpenAI. Gpt-4 technical report, 2023.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

Pacchiano, A., Saha, A., and Lee, J. Dueling rl: reinforcement learning with trajectory preferences. *arXiv preprint arXiv:2111.04850*, 2021.

Park, W., Kim, D., Lu, Y., and Cho, M. Relational knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3967–3976, 2019.

Perez, E., Huang, S., Song, F., Cai, T., Ring, R., Aslanides, J., Glaese, A., McAleese, N., and Irving, G. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*, 2022.

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

Qiu, Z., Ma, X., Yang, K., Liu, C., Hou, J., Yi, S., and Ouyang, W. Better teacher better student: Dynamic prior knowledge for knowledge distillation. *arXiv preprint arXiv:2206.06067*, 2022.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

Ramamurthy, R., Ammanabrolu, P., Brantley, K., Hessel, J., Sifa, R., Bauckhage, C., Hajishirzi, H., and Choi, Y. Is reinforcement learning (not) for natural language processing?: Benchmarks, baselines, and building blocks for natural language policy optimization. *arXiv preprint arXiv:2210.01241*, 2022.

Rashidinejad, P., Zhu, B., Ma, C., Jiao, J., and Russell, S. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems*, 34:11702–11716, 2021.

Romero, A., Ballas, N., Kahou, S. E., Chassang, A., Gatta, C., and Bengio, Y. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

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

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

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

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

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

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Schulman, J., Zoph, B., Kim, C., Hilton, J., Menick, J., Weng, J., Uribe, J. F. C., Fedus, L., Metz, L., Pokorny, M., et al. Chatgpt: Optimizing language models for dialogue. *OpenAI blog*, 2022.

Shah, N., Balakrishnan, S., Bradley, J., Parekh, A., Ramchandran, K., and Wainwright, M. Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence. In *Artificial Intelligence and Statistics*, pp. 856–865. PMLR, 2015.

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

Snell, C., Kostrikov, I., Su, Y., Yang, M., and Levine, S. Offline rl for natural language generation with implicit language q learning. *arXiv preprint arXiv:2206.11871*, 2022.

Song, F., Yu, B., Li, M., Yu, H., Huang, F., Li, Y., and Wang, H. Preference ranking optimization for human alignment. *arXiv preprint arXiv:2306.17492*, 2023a.

Song, Z., Cai, T., Lee, J. D., and Su, W. J. Reward collapse in aligning large language models. *arXiv preprint arXiv:2305.17608*, 2023b.

Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

Tian, Y., Krishnan, D., and Isola, P. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

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

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

Wirth, C., Akrour, R., Neumann, G., and Fürnkranz, J. A survey of preference-based reinforcement learning methods. *The Journal of Machine Learning Research*, 18 (1):4945–4990, 2017.

Wu, J., Ouyang, L., Ziegler, D. M., Stiennon, N., Lowe, R., Leike, J., and Christiano, P. Recursively summarizing books with human feedback. *arXiv preprint arXiv:2109.10862*, 2021.

Wu, T., Zhu, B., Zhang, R., Wen, Z., Ramchandran, K., and Jiao, J. Pairwise proximal policy optimization: Harnessing relative feedback for llm alignment, 2023.

Xie, T., Cheng, C.-A., Jiang, N., Mineiro, P., and Agarwal, A. Bellman-consistent pessimism for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 34:6683–6694, 2021.

Xu, Y., Wang, R., Yang, L., Singh, A., and Dubrawski, A. Preference-based reinforcement learning with finite-time guarantees. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (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.

Yim, J., Joo, D., Bae, J., and Kim, J. 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.

Yuan, Z., Yuan, H., Tan, C., Wang, W., Huang, S., and Huang, F. Rrhf: Rank responses to align language models with human feedback without tears. *arXiv preprint arXiv:2304.05302*, 2023.

Yue, Y. and Joachims, T. 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.

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

Yue, Y., Broder, J., Kleinberg, R., and Joachims, T. The k-armed dueling bandits problem. *Journal of Computer and System Sciences*, 78(5):1538–1556, 2012.

Zhao, B., Cui, Q., Song, R., Qiu, Y., and Liang, J. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pp. 11953–11962, 2022.

Zhao, Q. and Zhu, B. Towards the fundamental limits of knowledge transfer over finite domains. *arXiv preprint arXiv:2310.07838*, 2023.

Zhu, B., Jiao, J., and Jordan, M. I. Principled reinforcement learning with human feedback from pairwise or k-wise comparisons. *arXiv preprint arXiv:2301.11270*, 2023a.

Zhu, B., Sharma, H., Frujeri, F. V., Dong, S., Zhu, C., Jordan, M. I., and Jiao, J. Fine-tuning language models with advantage-induced policy alignment. *arXiv preprint arXiv:2306.02231*, 2023b.

Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

Zoghi, M., Whiteson, S., Munos, R., Rijke, M. d., 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.

Zoghi, M., Whiteson, S. A., De Rijke, M., and Munos, R. 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. Extension to Multi-wise Comparison

Here we discuss potential extensions from pairwise comparisons to multi-wise comparison. When there is M ranked responses for each prompt, there are two losses that one can choose from, namely MLE_2 and MLE_M from Zhu et al. (2023a).

$$
\begin{aligned}
&\hat{\theta}_{\mathsf{MLE}_2} \in \arg\min_r \mathcal{L}_2(\mathcal{D}, r),\\
\text{where } &\mathcal{L}_2(\mathcal{D}, r) = -\frac{1}{n}\sum_{i=1}^n \sum_{j=1}^M \sum_{k=j+1}^M \log\left(\frac{\exp(r(s_i, a_i^{(\sigma_i(j))}))}{\exp(r(s_i, a_i^{(\sigma_i(j))})) + \exp(r(s_i, a_i^{(\sigma_i(k))}))}\right)\\
&\hat{\theta}_{\mathsf{MLE}_M} \in \arg\min_r \mathcal{L}_M(\mathcal{D}, r),\\
\text{where } &\mathcal{L}_M(\mathcal{D}, r) = -\frac{1}{n}\sum_{i=1}^n \sum_{j=1}^M \log\left(\frac{\exp(r(s_i, a_i^{(\sigma_i(j))}))}{\sum_{k=j}^M \exp(r(s_i, a_i^{(\sigma_i(k))}))}\right).
\end{aligned}
$$

Here we discuss how to incorporate the iterative data smoothing algorithm for the two losses above.

The loss MLE_2 splits the M-wise comparisons into pairwise comparisons, thus it is straightforward to predict the new label $y_i^{j,k}$ for each pair of the comparisons between j-th and k-th response. The loss used for iterative data smoothing can be written as

$$
\begin{aligned}
\mathcal{L}_2^{\mathsf{DR}}(\mathcal{D}, r) = -\frac{1}{n}\sum_{i=1}^n \sum_{j=1}^M \sum_{k=j+1}^M y_i^{\sigma_i(j),\sigma_i(k)} &\log\left(\frac{\exp(r(s_i, a_i^{(\sigma_i(j))}))}{\exp(r(s_i, a_i^{(\sigma_i(j))})) + \exp(r(s_i, a_i^{(\sigma_i(k))}))}\right)\\
+ (1 - y_i^{\sigma_i(j),\sigma_i(k)}) &\log\left(\frac{\exp(r(s_i, a_i^{(\sigma_i(k))}))}{\exp(r(s_i, a_i^{(\sigma_i(j))})) + \exp(r(s_i, a_i^{(\sigma_i(k))}))}\right).
\end{aligned}
$$

$$
y_{i,t+1}^{j,k} = (1-\beta)\cdot y_{i,t}^{j,k} + \beta \cdot \frac{\exp(r_{\theta_{t+1}}(s_i, a_i^j))}{\exp(r_{\theta_{t+1}}(s_i, a_i^j)) + \exp(r_{\theta_{t+1}}(s_i, a_i^k))}.
$$

On the other hand, adapting the loss MLE_M for iterative data smoothing requires more efforts since it requires changing the ranking labels to soft labels. The design of MLE_M decomposes the probability of the observed ranking to the product of the probability that each response is the most preferred one among the rest of the responses. One of the options is to directly change the labels for the current rankings by the following update rules:

$$
\mathcal{L}_M(\mathcal{D}, r) = -\frac{1}{n}\sum_{i=1}^n \sum_{j=1}^M y_i^{\sigma_i(j)} \log\left(\frac{\exp(r(s_i, a_i^{(\sigma_i(j))}))}{\sum_{k=j}^M \exp(r(s_i, a_i^{(\sigma_i(k))}))}\right).
$$

And the update rule for the labels y_i is

$$
y_{i,t+1}^{\sigma_i(j)} = (1-\beta)\cdot y_{i,t}^{\sigma_i(j)} + \beta \cdot \frac{\exp(r_{\theta_{t+1}}(s_i, a_i^{(\sigma_i(j))}))}{\sum_{k=j}^M \exp(r_{\theta_{t+1}}(s_i, a_i^{(\sigma_i(k))}))}.
$$

However, the above update method does not directly reduce to the the case of pairwise comparisons when setting $M = 2$. In order to recover the pairwise loss, one needs to consider all possible rankings and get the soft labels for all the rankings. The loss will become

$$
\mathcal{L}'_M(\mathcal{D}, r) = -\frac{1}{n}\sum_{i=1}^n \sum_{\sigma\in\Pi(M)} \sum_{j=1}^M y_i^{j,\sigma} \log\left(\frac{\exp(r(s_i, a_i^{(\sigma(j))}))}{\sum_{k=j}^M \exp(r(s_i, a_i^{(\sigma(k))}))}\right).
$$

Here $\Pi(M)$ is the set of all permutations of the M elements. And the label is initialized as $y_{i,0}^{j,\sigma} = 1$ if $\sigma = \sigma_i$, and 0 otherwise. And the update rule for the labels y_i is

$$
y_{i,t+1}^{j,\sigma} = (1-\beta)\cdot y_{i,t}^{j,\sigma} + \beta \cdot \frac{\exp(r_{\theta_{t+1}}(s_i, a_i^{(\sigma(j))}))}{\sum_{k=j}^M \exp(r_{\theta_{t+1}}(s_i, a_i^{(\sigma(k))}))}.
$$

The loss is consistent with the pairwise cross entropy loss when $M = 2$. However, it requires enumerating over all possible permutations, which are not very efficient when M is large. It requires more study to decide which loss is more appropriate for M-wise iterative data smoothing.

B. An Alternative Formulation of Iterative Data Smoothing

Besides the formulation shown in Algorithm 1, we also propose an alternative formulation that directly multiplies a confidence c_i in front of the original loss for each sample, as is shown in Algorithm 2. We note here that although the algorithm has better asymptotic convergence result, its performance in practice is not as good as Algorithm 1.

Algorithm 2 Iterative Data Smoothing V2 $(\mathcal{D}, \theta_0, \alpha, \beta)$

```
  Input: The pairwise comparison dataset $\mathcal{D} = \{a_i, a'_i, y_i\}_{i=1}^n$. A parameterized reward model family $\{r_\theta : \mathcal{A} \mapsto \mathbb{R} \mid \theta \in \Theta\}$ with initialization $\theta_0 \in \Theta$. Two step sizes $\alpha, \beta$. An empirical loss function
      $\mathcal{L}_\theta(\{c_i\}, \mathcal{D}) = -\frac{1}{n}\sum_{i=1}^n \max(2c_i - 1, 0) \cdot \left(y_i \cdot \log\left(\frac{\exp(r_\theta(a_i))}{\exp(r_\theta(a_i)) + \exp(r_\theta(a'_i))}\right) + (1 - y_i) \cdot \log\left(\frac{\exp(r_\theta(a'_i))}{\exp(r_\theta(a_i)) + \exp(r_\theta(a'_i))}\right)\right).$
  Initialize $t = 0$ and $c_{i,0} = 1, \forall i$.
  while $r_{\theta_t}$ does not converge do
      $\theta_{t+1} \leftarrow \theta_t - \alpha \cdot \nabla \mathcal{L}_\theta(\{c_{i,t}\}, \mathcal{D})$
      $c_{i,t+1} \leftarrow (1 - \beta) \cdot c_{i,t} + \beta \cdot \frac{\exp(r_{\theta_{t+1}}(a_i))}{\exp(r_{\theta_{t+1}}(a_i)) + \exp(r_{\theta_{t+1}}(a'_i))}$
      $t \leftarrow t + 1$
  end while
  Return: $r_{\theta_t}$
```

We multiply a $\max(2c_i - 1, 0)$ in front of the loss for each sample as an approximation of how confident the current model predicts the preference label. When the reward is approximately similar, the coefficient goes to 0, putting less weights on those samples. Below we show that asymptotically, the new iterative data smoothing V2 algorithm is better at preserving the preference distribution compared with the original version.

Theorem B.1. *Consider the multi-armed bandit problem with the number of samples going to infinity and a fixed sampling distribution μ. Assume that $\mu(a, a') > 0$ for any $a, a' > 0$. Then we have*

- *Any stationary point for Algorithm 1 satisfies $\forall a, a', \hat{r}(a) = \hat{r}(a')$;*
- *There is one stationary point for Algorithm 2 that satisfies*

$$\forall a, a', \hat{r}(a) - \hat{r}(a') = r^\star(a) - r^\star(a').$$

Proof. The stationary points for Algorithm 1 and 2 are the points where the gradients equal 0. For Algorithm 1, this is equivalent to $\hat{y} = \frac{\exp(\hat{r}(a))}{\exp(\hat{r}(a)) + \exp(\hat{r}(a'))}$, and

$$\begin{aligned}
&(\mu(a \succ a') \cdot \hat{y} + \mu(a \prec a') \cdot (1 - \hat{y})) \cdot \frac{\exp(\hat{r}(a'))}{\exp(\hat{r}(a')) + \exp(\hat{r}(a'))} \\
=&(\mu(a \prec a') \cdot \hat{y} + \mu(a \succ a') \cdot (1 - \hat{y})) \cdot \frac{\exp(\hat{r}(a))}{\exp(\hat{r}(a)) + \exp(\hat{r}(a'))}.
\end{aligned}$$

Here $\hat{y}$ can be different for different (a, a'). Solving the above equation gives that the only stationary point is $\hat{y} = 1/2$ and $\hat{r}(a) - \hat{r}(a') = 0$.

On the other hand, for Algorithm 2, the stationary point condition is equivalent to $\hat{c}(a,a') = \frac{\exp(\hat{r}(a))}{\exp(\hat{r}(a))+\exp(\hat{r}(a'))}$, and

$$\sum_{a'} \max(2\hat{c}(a,a')-1,0)\cdot\left(\mu(a\succ a')\cdot\frac{\exp(\hat{r}(a'))}{\exp(\hat{r}(a'))+\exp(\hat{r}(a'))} - \mu(a\prec a')\cdot\frac{\exp(\hat{r}(a))}{\exp(\hat{r}(a))+\exp(\hat{r}(a'))}\right) = 0.$$

Thus one can verify that $\hat{r}(a)-\hat{r}(a') = r^\star(a) - r^\star(a')$ satisfies the stationary condition. This proves the result. □

Remark B.2. Although the asymptotic stationary points of Algorithm 1 do not contain the ground truth, the two-scale analysis discussed in Section 3.2.2 shows that when one of the step size is much larger than the other such that one of the updates in $\hat{y}$ or $\hat{r}$ is slower (and thus does not hit the stationary point), the reward still converges to the ground truth for those sufficiently observed arms. However, preliminary experiments on Algorithm 2 show that the result is worse than that of Algorithm 1, and also suffer from reward overfitting. This together with the failure of MLE may suggest that asymptotic result does not reflect the practical performance with smaller sample size compared with number of parameters.

Remark B.3. The condition of $\mu(a,a') > 0$ can be relaxed to that the comparison graph induced by the Laplace matrix L is connected, since the reward is identifiable in this case (Shah et al., 2015).

C. Experiments

In this section, we present the results of experiments with both multi-armed bandits and neural networks.

C.1. Multi-Armed Bandit

In the bandit setting, we focus on the hard example constructed in Theorem 2.2. We take total samples $n = 60$ and the number of arms K as 10 and 20. We compare the performance of the vanilla MLE, pessimistic MLE and IDS in both the reward learning phase and the policy learning phase.

In the reward learning phase, we run stochastic gradient descent with learning rate 0.01 on the reward model for multiple epochs and monitor how the loss changes with respect to the number of training epochs. For pessimistic MLE, we subtract the confidence level in the reward according to Equation (6). For IDS, we take the two step sizes as $\alpha = 0.01, \beta = 0.001$. As is shown in left part of Figure 3, both MLE and pessimistic MLE suffer from reward overfitting, while the test cross-entropy loss for the IDS algorithm continues to decrease until convergence. Since the training loss changes with the updated labels, we plot the population cross-entropy loss which is averaged over all pairs of comparisons.

In the right part of the figure, we plot the KL-reward tradeoff when training a policy based on the learned reward. We vary the choice of λ in Equation (5) to derive the optimal policy under diverse levels of KL constraint, where we take the reference policy π_0 as the uniform policy. One can see that IDS is able to converge to the optimal reward when KL is large, while both MLE and pessimistic MLE suffer from overoptimization.

We remark here that the reason pessimistic MLE suffers from both overfitting and overoptimization might be due to the design of unbounded reward in the multi-armed bandit case. When the reward family is bounded, pessimistic MLE is also guaranteed to mitigate the overoptimization issue. Furthermore, we only run one random seed for this setting to keep the plot clean since the KL-reward trade-off heavily depends on the observed samples.

C.2. Neural Network

We also conduct experiments with neural networks. We use the human-labeled Helpfulness and Harmlessnes (HH) dataset from Bai et al. (2022).[3] We take `Dahoas/pythia-125M-static-sft`[4] as the 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 it with a linear layer. When fine-tuning the language model, we use the proximal policy optimization (PPO) algorithm (Schulman et al., 2017).

We take a fully-trained 6B reward model `Dahoas/gptj-rm-static` trained from the same dataset based on `EleutherAI/gpt-j-6b` as the ground truth. We use the model to label the comparison samples using the BTL model (Bradley & Terry, 1952). And we train the 125M, 1B and 3B reward model with the new labeled comparison samples.

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

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

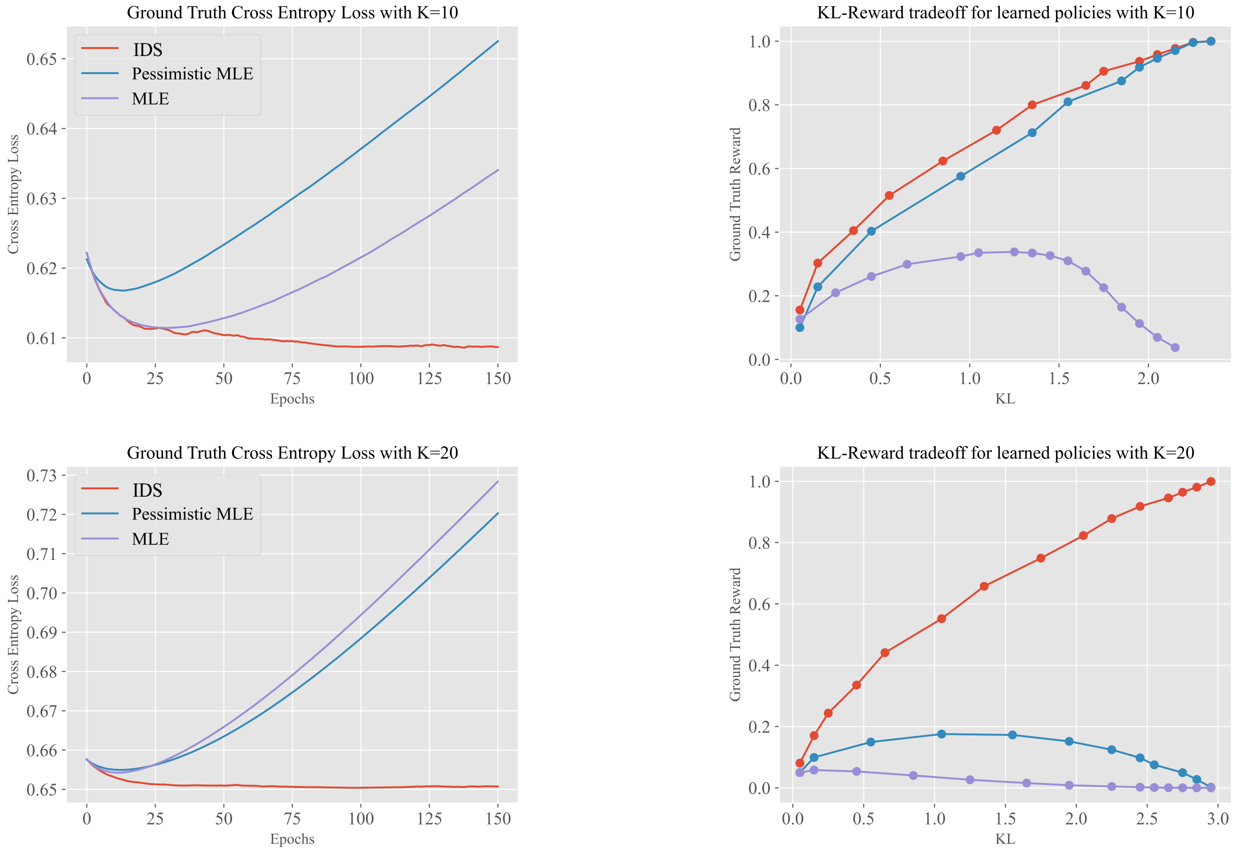

Figure 3: Comparisons of the three methods in the multi-armed bandit setting.

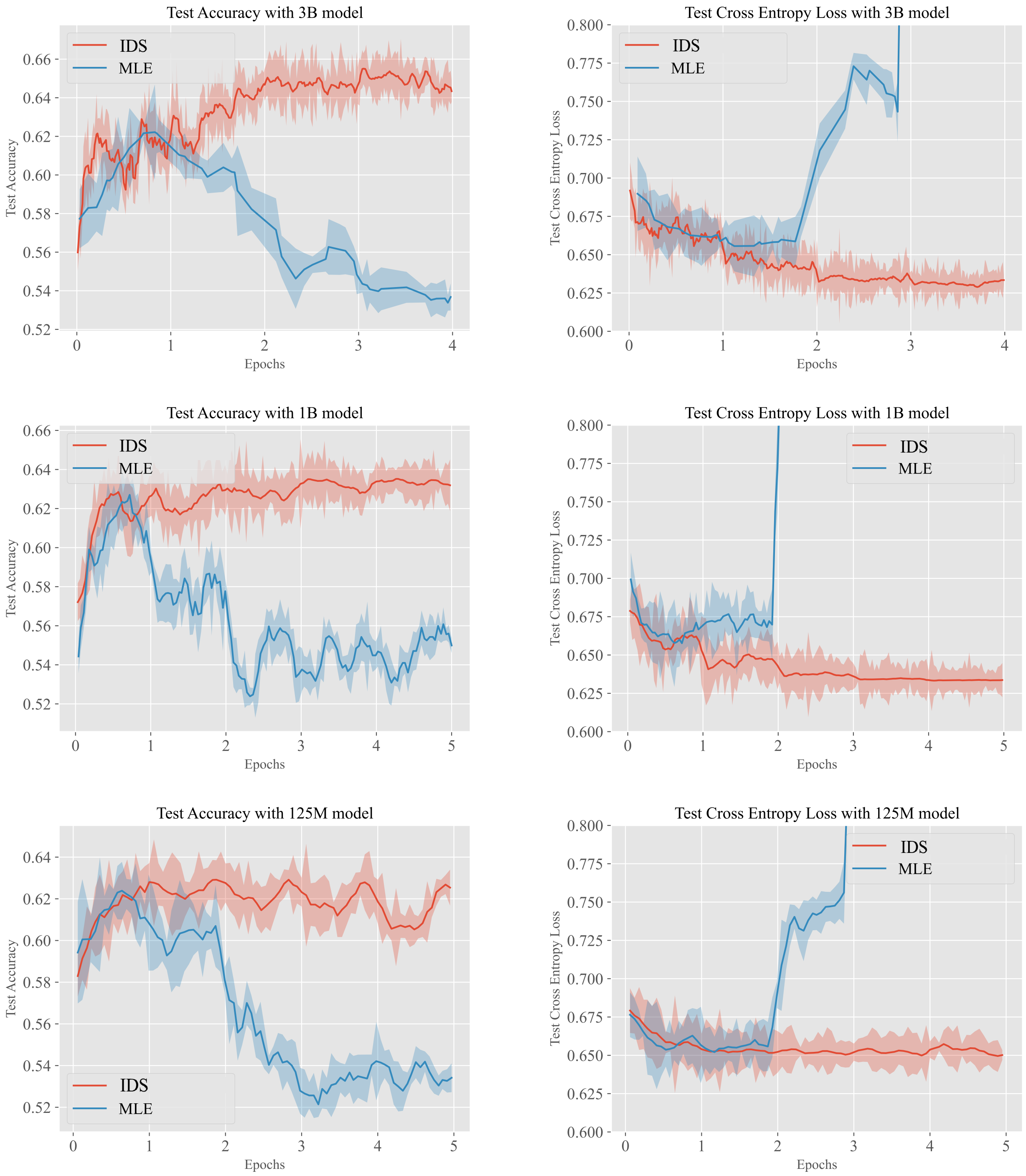

Figure 4: Comparisons of MLE and IDS when the reward is parameterized by a neural network.

The reward training results are shown in Figure 4. One can see that the MLE begins to overfit after 1-2 epochs, while the loss of the IDS algorithm continues to decrease stably until convergence.

For both MLE and IDS algortihms, we take the reward with the smallest evaluation loss and optimize a policy against the selected reward model. We compare results for policy learning as shown in Figure 5. One can see that MLE suffers from reward overoptimization with few thousand steps, while the ground truth reward continues to grow when using our IDS algorithm. We select step sizes $\alpha = 10^{-5}$ and $\beta = 0.7$ for all experiments. We observe that larger model leads to more improvement after one epoch, potentially due to more accurate estimation of the labels. We provide more details of the experiment along with the experiments on a different dataset, TLDR, in Appendix C.3.

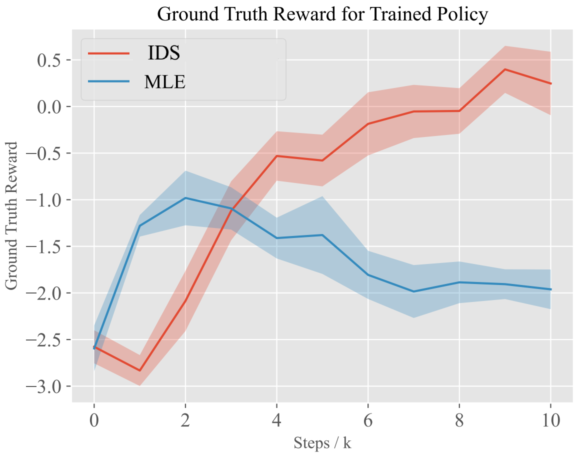

Figure 5: Comparison of MLE and IDS for policy learning

In the implementation, we find that it is helpful to restore the best checkpoint at the end of each epoch. This is due to that an inappropriate label $\{y_i\}_{i=1}^n$ at certain epoch may hurt the performance of the model. To prevent overfitting to the test set, we choose a large validation and test dataset, and we select the best checkpoint according to the smallest loss in the validation set, and plot the loss on the test set. During the whole training procedure including checkpoint restoration, we do not use any of the sample in the test set.

We also compare the algorithm with Laplace Smoothing, which simply replaces the hard label $y_i = 1$ with $y_i = 1 - \alpha$, and minimize the following loss function.

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

In our experiments, we conduct hyperparameter search and take $\alpha = 0.05$. The comparison is listed in Figure 6. One can see that Laplace Smoothing helps makes the convergence point slightly better than MLE, but still much worse than that of IDS.

C.3. Additional Experiments on TLDR

The hyper-parameters for the neural network experiments are listed in table 1.

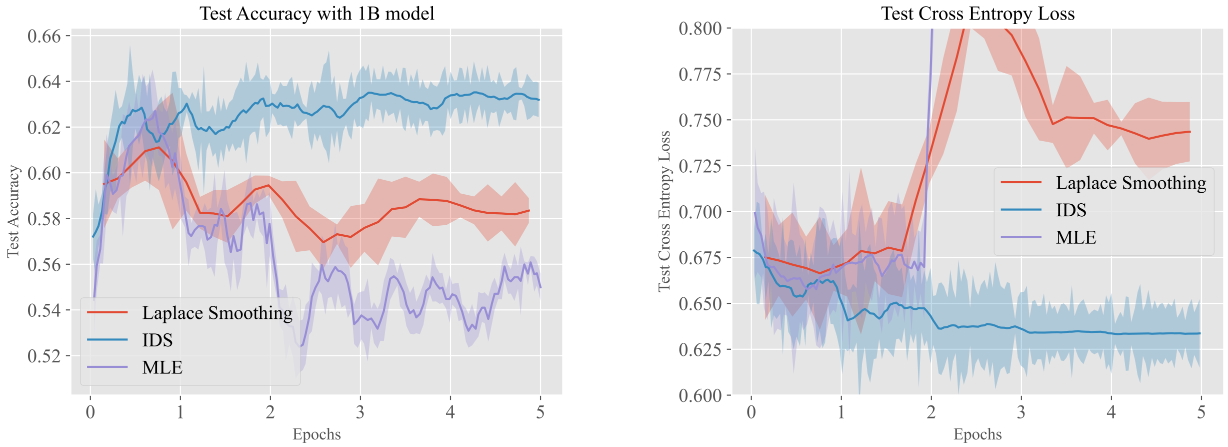

Figure 6: Comparisons of MLE, Laplace Smoothing and IDS.

Model	Parameter	Value
Reward model	learning rate α	10^{-5}
	label update parameter β	0.7
	batch size	128
	eval & save steps	100
Policy model	max sequence length	1024
	max output length	500
	generation temperature	1.0
	batch size	64
	fixed KL coefficient	0.001
	number of rollouts	128
	PPO epochs	4
	value coefficient	0.5
	GAE coefficient λ	0.95
	discount factor	1.0
	clip range	2

Table 1: Hyper-parameters for the neural network experiments

We also include additional experiments on a different dataset, TLDR[5], in Figure 7 and 8 of this section. The settings follow the same as HH in Section C.2. One can see that in the case of TLDR, the test accuracy does not drop significantly like HH. However, even with small difference in loss and the accuracy, the resulting policy reward difference is still significant.

D. Proof of Theorem 2.1

Proof. Let $\mathbb{P}_r(a, a', c) = \mathbb{P}_r(a \succ a')$ if $c = 1$, and $\mathbb{P}_r(a' \succ a)$ if $c = 0$ be the density function of the observations. According to Theorem 6.1.3. of Hogg et al. (2013), it suffices to verify the following conditions for the consistency of MLE:

- The CDFs are distinct, i.e. $\mathbb{P}_r(a, a', c) = \mathbb{P}_{r'}(a, a', c)$ almost everywhere implies that $r = r'$. This is true since the distribution is supported on discrete space, and the equality implies that $r(i) - r(j) = r(i)' - r(j)'$ for any i, j, and $r(M) = r'(M) = 1$.

- The PDFs have common support for all r. This is true since the probability is positive for any a, a', c.

[5] https://huggingface.co/datasets/CarperAI/openai_summarize_comparisons

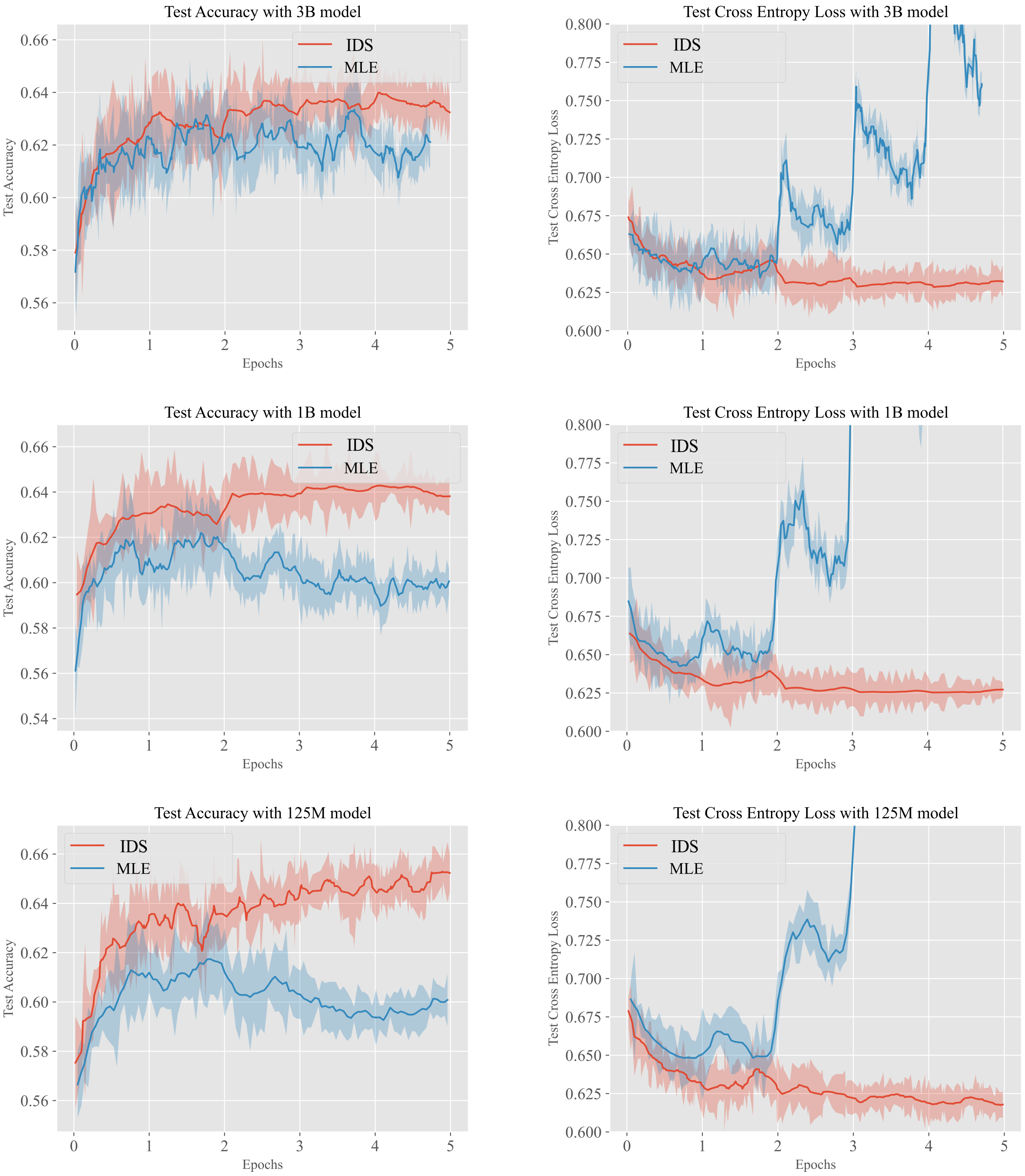

Figure 7: Comparisons of MLE and Iterative Data Smoothing when the reward is parameterized by a neural network.

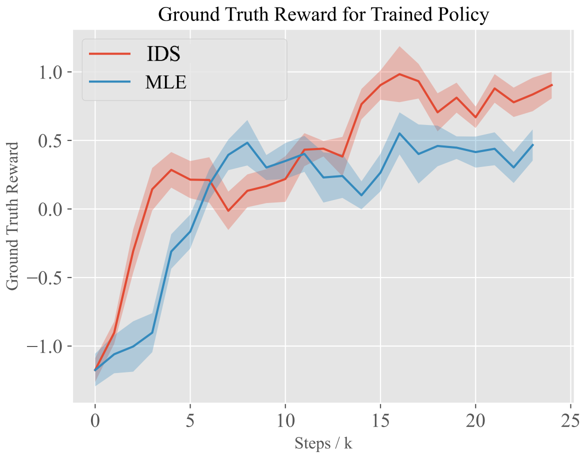

Figure 8: Comparison of MLE and Iterative Data Smoothing for policy learning.

- The point $r^\star$ is an interior point in $\mathbb{R}^K$. This is true by definition, since any open ball of radius ϵ around $r^\star$ is a subset of the space.

□

E. Proof of Theorem 2.2

Proof. The construction is in similar spirit to (Rashidinejad et al., 2021) and (Zhu et al., 2023a). Consider a bandit problem where $r^\star(a) = \mathbb{1}(a = 1)$. For any fixed n, we set $\mu(1, 2) = 1 - 1/n$, $\mu(1, 3) = 1/n$.

In this hard instance, there is constant probability that arm 3 is only compared with arm 1 once. Concretely, we have

$$\mathbb{P}(n(1, 3) = 1) = n \cdot (1 - \mu(1, 3))^{n-1} \cdot \mu(1, 3) = (1 - 1/n)^{n-1}.$$

When $n \geq 500$, we have $\mathbb{P}(n(1, 3) = 1) \geq 0.36$. Under this case, we know that arm 3 is preferred with probability at least $\exp(r(3))/(\exp(r(1)) + \exp(r(3))) > 0.26$. When there is only one comparison between arm 1 and 3, and arm 3 is preferred, the MLE assigns $r(3)$ as infinity. Even when the reward for arm 1 is estimated perfectly, this leads to a population cross-entropy loss arbitrarily large.

□

F. Proof of Corollary 2.3

Proof. The proof follows immediately from Theorem 2.2. Under the same construction, we know that $\hat{r}_{\mathsf{MLE}}(3) = +\infty$ with probability at least 0.09. Thus, the sub-optimality of the resulting optimal policy is at least 1. □

G. Proof of Theorem 3.1

Proof. Let $\hat{r}_i$ be the reward for the i-th arm, and $\hat{r} = [\hat{r}_1, \hat{r}_2, \cdots, \hat{r}_K]$ as the vector for the reward. One can calculate the gradient of the reward as

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

Here the last equality is due to that all the reward is initialized at the same value. And $n_+(i)$ (or $n_-(i)$) refers the total number of winning (or losing) of arm i in the observations.

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

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

This proves the result.

□

H. Proof of Theorem 3.3

Proof. From the differential equations in (7), we know that

$$\frac{\dot{y}(t)}{y(t)} \geq -\beta.$$

Taking integration on both sides give us

$$y(t) \geq \exp(-\beta t) \geq \exp(-\epsilon).$$

Now we set a Lyapunov function $V(t) = \left(\frac{\exp(d(t))}{1+\exp(d(t))} - \mu \right)^2$. We know that

$$\begin{aligned}
\dot{V}(t) &= 2 \left(\frac{\exp(d(t))}{1 + \exp(d(t))} - \mu \right) \cdot \frac{\exp(d(t))}{(1 + \exp(d(t)))^2} \cdot \dot{d}(t) \\
&= 2\alpha n \left(\frac{\exp(d(t))}{1 + \exp(d(t))} - \mu \right) \cdot \frac{\exp(d(t))}{(1 + \exp(d(t)))^2} \\
&\quad \cdot \left((\mu \cdot y(t) + (1 - \mu) \cdot (1 - y(t))) \cdot \frac{1}{1 + \exp(d(t))} - ((1 - \mu) \cdot y(t) + \mu \cdot (1 - y(t))) \cdot \frac{\exp(d(t))}{1 + \exp(d(t))} \right) \\
&= 2\alpha n \left(\frac{\exp(d(t))}{1 + \exp(d(t))} - \mu \right) \cdot \frac{\exp(d(t))}{(1 + \exp(d(t)))^2} \cdot \left((2\mu - 1) \cdot y(t) + 1 - \mu - \frac{\exp(d(t))}{1 + \exp(d(t))} \right) \\
&= -2\alpha n \cdot \frac{\exp(d(t))}{(1 + \exp(d(t)))^2} \cdot \left(\frac{\exp(d(t))}{1 + \exp(d(t))} - \mu \right)^2 \\
&\quad + 2\alpha n \cdot \frac{\exp(d(t))}{(1 + \exp(d(t)))^2} \cdot \left(\frac{\exp(d(t))}{1 + \exp(d(t))} - \mu \right) \cdot (2\mu - 1) \cdot (y(t) - 1) \\
&\overset{(i)}{\leq} 2\alpha n \cdot \frac{\exp(d(t))}{(1 + \exp(d(t)))^2} \cdot \left(-\left(\frac{\exp(d(t))}{1 + \exp(d(t))} - \mu \right)^2 + 1 - \exp(-\epsilon) \right) \\
&= 2\alpha n \cdot \frac{\exp(d(t))}{(1 + \exp(d(t)))^2} \cdot (-V(t) + 1 - \exp(-\epsilon)).
\end{aligned}$$

Here (i) uses the fact that $y(t), \mu, \frac{\exp(d(t))}{1+\exp(d(t))} \in [0,1]$. Now consider two scenarios. The first is that for any time $t \in [0, T]$, one always has $V(t) \geq 2(1-\exp(-\epsilon))$. In this case, we know that

$$\begin{aligned}\dot{V}(t) &\leq 2\alpha n \cdot \frac{\exp(d(t))}{(1+\exp(d(t)))^2} \cdot (-V(t) + 1 - \exp(-\epsilon)) \\ &\leq -\alpha n \cdot \frac{\exp(d(t))}{(1+\exp(d(t)))^2} \cdot V(t) \\ &\leq 0. \end{aligned} \tag{8}$$

This shows that $V(t)$ is a non-increasing function. Without loss of generality, assume that $\mu \geq 1/2$. We know that

$$V(t) \leq V(t'), \forall t > t'. \tag{9}$$

Now we prove that there must be $\frac{\exp(d(t))}{1+\exp(d(t))} \leq \mu$. If one can find some t_0 such that $\frac{\exp(d(t))}{1+\exp(d(t))} > \mu$, by the continuity of $\frac{\exp(d(t))}{1+\exp(d(t))}$ and the fact that $\frac{\exp(d(0))}{1+\exp(d(0))} = 1/2$, one can find some $t_1 < t_0$ such that $\frac{\exp(d(t_1))}{1+\exp(d(t_1))} = \mu$. This gives that

$$V(t_1) = 0 < V(t_0),$$

which contradicts Equation (9). Thus we know that $\frac{\exp(d(t))}{1+\exp(d(t))} \leq \mu$ holds for any t. Furthermore, since we know that $V(t)$ is non-increasing, we know that $\frac{\exp(d(t))}{1+\exp(d(t))} \geq 1/2$. This also implies that

$$\frac{\exp(d(t))}{(1+\exp(d(t)))^2} \geq \mu(1-\mu).$$

Similarly, we can prove the same condition holds when $\mu < 1/2$. Thus we have

$$\frac{\dot{V}(t)}{V(t)} \leq -\mu(1-\mu)\alpha n.$$

By integrating over t on both sides, we have

$$V(t) \leq \exp(-\mu(1-\mu)\alpha n t) \cdot V(0) \leq \exp(-\mu(1-\mu)\alpha n t).$$

Here the last inequality uses the fact that $V(0) \in [0,1]$.

On the other hand, assume that at some time point $t_0 \in [0, T]$, we have $V(t_0) < 2(1-\exp(-\epsilon))$. When $V(T) > 2(1-\exp(-\epsilon))$, by the continuity of the function $V(\cdot)$, we know that there exists some t_1 such that $V(t_1) = 2(1-\exp(-\epsilon))$, and for any $t \in [t_1, T]$, $V(t) \geq 2(1-\exp(-\epsilon))$. From Equation (8), we know that in the regime of $t \in [t_1, T]$, $V(t)$ is non-increasing. This contradicts with the fact that $V(T) > V(t_1)$. Thus we know that

$$V(T) \leq 2(1-\exp(-\epsilon)).$$

□