Optimal Design for Human Feedback

Subhojyoti Mukherjee[*]
ECE Department
UW-Madison
Wisconsin, Madison
smukherjee27@wisc.edu

Anusha Lalitha
AWS AI Labs
Santa Clara
USA

Kousha Kalantari
AWS AI Labs
Santa Clara
USA

Aniket Deshmukh
AWS AI Labs
Santa Clara
USA

Ge Liu
AWS AI Labs
Santa Clara
USA

Yifei Ma
AWS AI Labs
Santa Clara
USA

Branislav Kveton
AWS AI Labs
Santa Clara
USA

Abstract

Learning of preference models from human feedback has been central to recent advances in artificial intelligence. Motivated by the cost of obtaining high-quality human annotations, we study the problem of data collection for learning preference models. The key idea in our work is to generalize the optimal design, a method for computing information gathering policies, to ranked lists. To show the generality of our ideas, we study both absolute and relative feedback on the lists. We design efficient algorithms for both settings and analyze them. We prove that our preference model estimators improve with more data and so does the ranking error under the estimators. Finally, we experiment with several synthetic and real-world datasets to show the statistical efficiency of our algorithms.

1 Introduction

Reinforcement learning with human feedback (RLHF) has been shown to be effective in aligning and fine-tuning *large language models (LLMs)* [44, 74, 41, 18, 102, 82, 31]. The difference from classic *reinforcement learning (RL)* [86] is that the learner learns from human feedback, which is expressed in the form of preferences among different potential choices [94, 12, 55, 3, 78]. The human feedback allows LLMs to be adapted beyond the distribution of data that was used for their pre-training and generate answers that are more preferred by humans [18]. The feedback can be incorporated by learning a preference model. When the human decides between two choices, the *Bradley-Terry-Luce (BTL)* model [13] can be used. When it is among multiple choices, the *Plackett-Luce (PL)* model [71, 61] can be used. Learning of a good preference model can be seen as ranking answers to questions, a well-known setting within learning to rank. Numerous works have explored this topic, in both offline [16] and online [73, 47, 88, 85, 52] settings.

To effectively learn preference models from human feedback, we study efficient methods for data collection. We formalize this problem as follows. We have a set of L lists representing questions, each containing K items representing answers. The goal of the learner is to learn to rank all items in all lists. The learner can query humans for feedback. Each query is a question with K answers represented as a list. The human provides feedback on it. We study two settings: absolute and ranking feedback. In the absolute feedback setting, a human provides noisy values for all items in the list. This setting is motivated by how human annotators assign relevance scores in search [34, 63]. The

[*]Work conducted during an internship at Amazon.

Preprint. Under review.

ranking feedback is motivated by learning preference models in RLHF [44, 74, 41, 18, 102, 82, 20]. In this setting, a human ranks all items in the list, which indicates their preference. While $K = 2$ is arguably the most common case, we study $K \geq 2$ to provide a more general insight on the problem, and also to allow for a higher-capacity communication channel with the human [111]. The learner has a budget for the number of queries. To learn efficiently within the budget, they need to ask for feedback on the most informative lists, which allows them to learn to rank all other lists. Our main contribution is an efficient algorithm for computing the distribution over the most informative lists.

Our work touches on many classic topics and recent works. Learning of reward models from human feedback is at the center of RLHF [70] and its recent popularity has led to major theory developments, including analyses of regret minimization in RLHF [19, 91, 94, 98, 69, 80]. These works propose and analyze adaptive algorithms that interact with the environment to learn highly-rewarding policies. In practice, though, such policies are hard to deploy because they may over-explore initially, which harms user experience. They may also need to be recomputed frequently [24, 87]. Zhu et al. [111] studied RLHF from ranking feedback in the offline setting with a fixed dataset. We focus on collecting an *informative dataset for offline learning to rank* with both absolute and ranking feedback. The data logging problem is framed as an optimal design. The optimal design aims to find a distribution over the most informative choices that minimizes uncertainty in some metric [72, 26]. This distribution generally solves an optimization problem and has some desirable properties, like sparsity. The main technical contribution of this work is a matrix generalization of the Kiefer-Wolfowitz theorem [46], which allows us to formulate optimal designs over ranked lists and solve them efficiently. Optimal designs have become a standard tool in exploration [42, 51, 43, 64, 38]. We contribute to these works by proposing the *first pure exploration algorithm for ranked lists based on optimal designs*. More precisely, our setting can be viewed as fixed-budget *best-arm identification (BAI)* [6, 101, 8, 50], where the best arm corresponds to all lists being correctly ranked.

We make the following contributions:

(1) We develop a novel approach for logging data for learning to rank from human feedback. The key idea is to generalize the Kiefer-Wolfowitz theorem [46] to matrices (Section 3), which then allows us to compute information-gathering policies for ranked lists.

(2) In the absolute feedback model, we propose an algorithm that uses an optimal design to collect absolute human feedback (Section 4.1). A least-squares estimator is then used to learn from it. This combination is both computationally and statistically efficient. Specifically, we bound the estimation errors of the algorithm (Section 4.2) and the resulting ranking loss (Section 4.3), and show that both decrease with the sample size.

(3) In the ranking feedback model, we propose an algorithm that uses an optimal design to collect ranking human feedback (Section 5.1). A *maximum likelihood estimator (MLE)* [111] is then used to learn from it. This combination is both computationally and statistically efficient, and we bound the resulting estimation errors and ranking loss in Sections 5.2 and 5.3, respectively. These results mimic the absolute feedback setting and show the generality of our proposed framework.

(4) We compare our algorithms to multiple baselines on several synthetic and real-world datasets. We observe that our algorithms achieve a lower ranking loss than the baselines.

2 Setting

Notation: Let $[K] = \{1, \dots, K\}$. Let $\triangle^L$ be the probability simplex over $[L]$. For any distribution $\pi \in \triangle^L$, we have $\sum_{i=1}^L \pi(i) = 1$. Let $\Pi_2(K) = \{(j, k) : j < k; j, k \in [K]\}$ be the set of pairs over $[K]$ where the first entry is lower than the second one. Let $\|\mathbf{x}\|_{\mathbf{A}}^2 = \mathbf{x}^\top \mathbf{A} \mathbf{x}$ for any positive-definite $\mathbf{A} \in \mathbb{R}^{d \times d}$ and $\mathbf{x} \in \mathbb{R}^d$. We use $\widetilde{O}$ for the big-O notation up to logarithmic factors. Specifically, for any function f, we write $\widetilde{O}(f(n))$ if it is $O(f(n) \log^k f(n))$ for some $k > 0$. Let Supp denote the support of a distribution or random variable.

Setup: We learn to rank L lists with K items. An item $k \in [K]$ in list $i \in [L]$ is represented by its feature vector $\mathbf{x}_{i,k} \in \mathcal{X}$, where $\mathcal{X} \subseteq \mathbb{R}^d$ is the set of all feature vectors. The relevance of items is determined by their mean rewards. The mean reward of item k in list i is $\mathbf{x}_{i,k}^\top \boldsymbol{\theta}_*$, where $\boldsymbol{\theta}_* \in \mathbb{R}^d$ is an unknown parameter. Without loss of generality, we assume that $\mathbf{x}_{i,j}^\top \boldsymbol{\theta}_* > \mathbf{x}_{i,k}^\top \boldsymbol{\theta}_*$ for any $j < k$ in any list i. Therefore, the original order of the items is optimal. The learner does not know it. The learner interacts with the lists over n rounds. At round t, they select a list I_t and the human labeler provides stochastic feedback on it. Our goal is to design a policy for selecting the lists such that the learner learns the optimal order of all items in all lists after n rounds. Our setting resembles *best arm identification (BAI)* [15, 6, 101, 8] with a fixed budget where the goal is to identify the arm with the highest mean reward in a stochastic bandit.

Feedback Model: We study two models of human feedback, absolute and ranking:

(1) In the *absolute feedback model*, at any round t, the human labeler provides a reward for each item in list I_t selected by the learner. Specifically, the learner observes noisy rewards

$$y_{t,k} = \mathbf{x}_{I_t,k}^\top \boldsymbol{\theta}_* + \eta_{t,k} \,, \tag{1}$$

for all $k \in [K]$ in list I_t, where $\eta_{t,k}$ is independent zero-mean 1-sub-Gaussian noise. Therefore, the reward provided by the human labeler is stochastic, with mean $\mathbf{x}_{I_t,k}^\top \boldsymbol{\theta}_*$ and additive independent noise. This is similar to the document-based click model [22] in learning to rank.

(2) In the *ranking feedback model*, at any round t, the human labeler orders all K items in list I_t selected by the learner. Specifically, the learner observes a permutation over all K items in list I_t sampled from the *Plackett-Luce (PL)* model [71, 61]. This feedback model has been studied before [111, 39]. Let $\sigma_t : [K] \to [K]$ be the permutation provided by the human labeler at round t, where $\sigma_t(k)$ is the index of the k-th ranked item. Then the PL model generates σ_t with probability

$$\mathbb{P}\left(\sigma_t\right) = \prod_{k=1}^{K} \frac{\exp\left(\mathbf{x}_{I_t,\sigma_t(k)}^\top \boldsymbol{\theta}_*\right)}{\sum_{j=k}^{K} \exp\left(\mathbf{x}_{I_t,\sigma_t(j)}^\top \boldsymbol{\theta}_*\right)} \,. \tag{2}$$

The PL model provides a probabilistic ranking using the underlying mean rewards with unknown parameter $\boldsymbol{\theta}_*$. Because the feedback at round t is with independent noise, in both (1) and (2), any list can be observed multiple times and we do need to assume that $n \leq L$.

Objective: At the end of n rounds, the learner outputs a permutation $\widehat{\sigma}_{n,i} : [K] \to [K]$ for each list $i \in [L]$, where $\widehat{\sigma}_{n,i}(k)$ is the item placed at position k in list i. Our evaluation metric is the *ranking loss* after n rounds, which we define as

$$\mathrm{R}_n = \sum_{i=1}^{L} \sum_{j=1}^{K} \sum_{k=j+1}^{K} \mathbb{I}\{\widehat{\sigma}_{n,i}(j) > \widehat{\sigma}_{n,i}(k)\} \,. \tag{3}$$

In plain English, the ranking loss is the number of incorrectly ordered pairs of items in permutation $\widehat{\sigma}_{n,i}$, summed over all lists $i \in [L]$. It can also be viewed as the Kendall tau rank distance [45] between the optimal order of items in all lists and that according to $\widehat{\sigma}_{n,i}$. We note that other ranking metrics exist, such as the *normalized discounted cumulative gain (NDCG)* [90] and *mean reciprocal rank (MRR)* [89]. These consider both the order of items and their relevance scores. We believe that our analyses can be extended to these metrics and leave this for future work.

We introduce optimal designs [72, 26] next. This allows us to minimize the expected ranking loss within a budget of n rounds efficiently.

3 Optimal Design and Matrix Kiefer-Wolfowitz

This section introduces a unified approach to data collection for both absolute and ranking feedback. First, we note that to learn the optimal order of items in all lists, the learner needs to estimate the unknown parameter $\boldsymbol{\theta}_*$ well. In this work, the learner uses a *maximum-likelihood estimator (MLE)* to obtain an estimate $\widehat{\boldsymbol{\theta}}_n$ of $\boldsymbol{\theta}_*$. After that, it orders the items in all lists according to their estimated mean rewards $\mathbf{x}_{i,k}^\top \widehat{\boldsymbol{\theta}}_n$ in descending order to obtain the permutation $\widehat{\sigma}_{n,i}$. If $\widehat{\boldsymbol{\theta}}_n$ could minimize the prediction error $(\mathbf{x}_{i,k}^\top(\boldsymbol{\theta}_* - \widehat{\boldsymbol{\theta}}_n))^2$ for all items $k \in [K]$ in list i, then the permutation $\widehat{\sigma}_{n,i}$ would be closer to the optimal order. Moreover, by minimizing the maximum prediction error over all lists, the learner can learn the optimal order in all lists and minimize the ranking loss in (3). Therefore, we are concerned with optimizing the *maximum prediction error*

$$\max_{i \in [L]} \sum_{\mathbf{a} \in \mathbf{A}_i} \left(\mathbf{a}^\top \left(\boldsymbol{\theta}_* - \widehat{\boldsymbol{\theta}}_n\right)\right)^2 = \max_{i \in [L]} \mathbf{Tr}\left(\mathbf{A}_i^\top \left(\boldsymbol{\theta}_* - \widehat{\boldsymbol{\theta}}_n\right)\left(\boldsymbol{\theta}_* - \widehat{\boldsymbol{\theta}}_n\right)^\top \mathbf{A}_i\right) \,, \tag{4}$$

where $\mathbf{A}_i$ is a matrix representing list i and $\mathbf{a} \in \mathbf{A}_i$ is a column in it. In the absolute feedback model, the columns of $\mathbf{A}_i$ are feature vectors of items in list i (Section 4.1). In the ranking feedback model, the columns of $\mathbf{A}_i$ are the differences of the feature vectors of items in list i (Section 5.1). Therefore, $\mathbf{A}_i$ is human-feedback model specific. As we show later, the algebraic form of $\mathbf{A}_i$ is dictated by the covariance of $\widehat{\boldsymbol{\theta}}_n$ in the corresponding feedback model.

We prove in Sections 4 and 5 that the learner can minimize the maximum prediction error in (4) and the ranking loss in (3) by sampling from a fixed distribution $\pi_* \in \triangle^L$. That is, the probability of selecting list i at round t is $\mathbb{P}(I_t = i) = \pi_*(i)$. The distribution π_* is a minimizer of

$$g(\pi) = \max_{i \in [L]} \mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_\pi^{-1} \mathbf{A}_i)\,, \tag{5}$$

where $\mathbf{V}_\pi = \sum_{i=1}^L \pi(i) \mathbf{A}_i \mathbf{A}_i^\top$ is a design matrix. The *optimal design* aims to find the distribution π_*. Since it does not depend on the received feedback, our algorithms are not adaptive.

The problem of finding π_* that minimizes (5) is called the *G-optimal design* [51]. The minimum of (5) and the support of π_* are characterized by the Kiefer-Wolfowitz theorem [46, 51]. The original theorem is for least-squares regression, where $\mathbf{A}_i$ are feature vectors. We generalize it to ranked lists, where $\mathbf{A}_i$ are matrices of feature vectors representing list i. This generalization allows us to go from a design over feature vectors to a design over lists represented by matrices.

Theorem 1 (Matrix Kiefer-Wolfowitz)**.** *Consider any L matrices $\mathbf{A}_i \in \mathbb{R}^{d \times M}$ for $i \in [L]$, whose column space spans $\mathbb{R}^d$. Let $\mathbf{V}_\pi$ be the design matrix in* (5)*. Then the following are equivalent:*

(a) π_* is a minimizer of $g(\pi)$ defined in (5)*.*

(b) π_* is a maximizer of $f(\pi) = \log\det(\mathbf{V}_\pi)$.

(c) $g(\pi_*) = d$.

Furthermore, there exists a minimizer π_* of $g(\pi)$ such that $|\mathrm{Supp}(\pi_*)| \leq d(d+1)/2$.

Proof. We generalize the proof of the Kiefer-Wolfowitz theorem in Lattimore and Szepesvári [51]. The key observation is that even if $\mathbf{A}_i$ is a matrix and not a vector, the design matrix $\mathbf{V}_\pi$ is positive definite. Using this, we establish the following two key facts used in the original proof. First, we show that f is concave in π and that $(\nabla f(\pi))_i = \mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_\pi^{-1} \mathbf{A}_i)$ is its gradient with respect to $\pi(i)$. Next we show that $\sum_{i=1}^L \pi(i) \mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_\pi^{-1} \mathbf{A}_i) = d$. The complete proof is in Appendix B. □

From the equivalence in Theorem 1, it follows that the learner should solve the optimal design

$$\pi_* = \max_{\pi \in \triangle^L} f(\pi) = \max_{\pi \in \triangle^L} \log\det(\mathbf{V}_\pi) \tag{6}$$

and sample according to π_* to minimize the maximum prediction error in (4). Note that the optimal design over lists in (6) is different from the one over features [51]. As an example, suppose that we have 4 feature vectors $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$, and two lists $\mathbf{A}_1 = (\mathbf{x}_1, \mathbf{x}_2), \mathbf{A}_2 = (\mathbf{x}_3, \mathbf{x}_4)$. The list design in (6) is over 2 variables (lists) while the feature-vector design would be over 4 variables (feature vectors). The list design can also be viewed as a constrained feature-vector design, where the pairs $(\mathbf{x}_1, \mathbf{x}_2)$ and $(\mathbf{x}_3, \mathbf{x}_4)$ are observed together with the same probability.

The optimization problem in (6) is convex and thus easy to solve. When the number of lists is large, the Frank-Wolfe algorithm [67, 72, 37] can be applied, which solves convex optimization problems with linear constraints as a sequence of linear programs. We use CVXPY [23] to compute the optimal design and report the computation time for various numbers of lists L in Table 1 (Appendix E). For $L = 100$, the computation takes 4 seconds; and for $L = 400$, it takes 20. Therefore, it is fast. In the following sections, we use Theorem 1 to bound the maximum prediction error and ranking loss for both absolute and ranking feedback.

4 Learning with Absolute Feedback

This section is organized as follows. First, we present our algorithm, which logs absolute feedback using a policy computed by an optimal design and learns a model from it. Then we analyze it, by bounding its maximum prediction error in (4) and the expected ranking loss in (3).

4.1 Algorithm Dope

Now we present our algorithm for absolute feedback called **D-op**timal d**e**sign (Dope). The algorithm has four main parts. First, we solve the optimal design problem in (6) to get a data logging policy π_*. The matrix for list i is $\mathbf{A}_i = [\mathbf{x}_{i,k}]_{k \in [K]} \in \mathbb{R}^{d \times K}$, where $\mathbf{x}_{i,k}$ is the feature vector of item k in list i. This algebraic form arises from the covariance matrix of the estimator in (8). Specifically, note that $\sum_{k=1}^K \mathbf{x}_{i,k}\mathbf{x}_{i,k}^\top = \mathbf{Tr}(\mathbf{A}_i \mathbf{A}_i^\top)$. Second, the policy π_* is used to collect human feedback for n rounds. At round $t \in [n]$, the learner samples a list $I_t \sim \pi_*$ and observes $y_{t,k}$ for $k \in [K]$, as defined in (1). Third, we estimate the unknown model parameter as

$$\widehat{\boldsymbol{\theta}}_n = \bar{\boldsymbol{\Sigma}}_n^{-1} \sum_{t=1}^{n} \sum_{k=1}^{K} \mathbf{x}_{I_t,k} y_{t,k} \,. \tag{7}$$

The normalized and unnormalized covariance matrices corresponding to the estimate are

$$\boldsymbol{\Sigma}_n = \frac{1}{n} \sum_{t=1}^{n} \sum_{k=1}^{K} \mathbf{x}_{I_t,k} \mathbf{x}_{I_t,k}^\top \,, \quad \bar{\boldsymbol{\Sigma}}_n = n \boldsymbol{\Sigma}_n \,, \tag{8}$$

respectively. Finally, the learner orders the items in every list $i \in [L]$ according to their estimated mean rewards $\mathbf{x}_{i,k}^\top \widehat{\boldsymbol{\theta}}_n$ in descending order to obtain the permutation $\widehat{\sigma}_{n,i}$.

When the noise is sub-Gaussian, our MLE is the same as *ordinary least-squares (OLS)*. Therefore, the optimal design under absolute feedback logs data for a least-squares problem by minimizing the covariance of the OLS [51, 38]. We present the full pseudo-code in Algorithm 1 in Appendix F.

4.2 Maximum Prediction Error Under Absolute Feedback

In this section, we bound the prediction error of Dope under absolute feedback. We start with a lemma that uses the optimal design π_* to bound $\max_{i \in [L]} \sum_{\mathbf{a} \in \mathbf{A}_i} \|\mathbf{a}\|^2_{\bar{\boldsymbol{\Sigma}}_n^{-1}}$.

Lemma 2. *Let π_* be the optimal design in (6). Fix budget n and let each allocation $n\pi_*(i)$ be an integer. Then $\max_{i \in [L]} \sum_{\mathbf{a} \in \mathbf{A}_i} \|\mathbf{a}\|^2_{\bar{\boldsymbol{\Sigma}}_n^{-1}} = d/n$.*

This lemma is proved in Appendix C.1. With this result in hand, the maximum prediction error can be bounded as follows.

Theorem 3 (Maximum prediction error)**.** *Fix $\delta \in [0, 1)$. Then, with probability at least $1 - \delta$, the maximum prediction error after n rounds is bounded as*

$$\max_{i \in [L]} \mathbf{Tr}\left(\mathbf{A}_i^\top \left(\boldsymbol{\theta}_* - \widehat{\boldsymbol{\theta}}_n\right)\left(\boldsymbol{\theta}_* - \widehat{\boldsymbol{\theta}}_n\right)^\top \mathbf{A}_i\right) = \widetilde{O}\left(\frac{d^2 \log(1/\delta)}{n}\right).$$

This claim is proved in Appendix C.2. Theorem 3 shows that the maximum prediction error under absolute feedback is $O(1/n)$, when the learner selects lists using the optimal design π_*. In Lemma 2 and Theorem 3, we assume that the number of times that each list is chosen is an integer. If the allocations were not integers, rounding errors would arise. Efficient rounding procedures have been established for such cases [72, 27, 42]. This would only introduce a constant multiplicative factor $1 + \beta$ in our bounds [51] (notes in Chapter 21) for some $\beta > 0$. For simplicity, we omit this factor in our derivation. Finally, since we assume integer allocations, the covariance matrix is invertible, as in Zhu et al. [111].

4.3 Ranking Loss Under Absolute Feedback

In this section, we bound the expected ranking loss under absolute feedback. Recall from Section 2 that the original order of items in each list is optimal. With this in mind, the *gap* between the mean rewards of items j and k in list i is $\Delta_{i,j,k} = (\mathbf{x}_{i,j} - \mathbf{x}_{i,k})^\top \boldsymbol{\theta}_*$, for any $i \in [L]$ and $(j, k) \in \Pi_2(K)$.

Theorem 4 (Ranking loss)**.** *The expected ranking loss under absolute feedback is bounded as*

$$\mathbb{E}\left[\mathrm{R}_n\right] \le \sum_{i=1}^{L} \sum_{j=1}^{K} \sum_{k=j+1}^{K} 2 \exp\left(-\frac{n \Delta_{i,j,k}^2}{2d}\right).$$

Proof. From the definition of the ranking loss, we have

$$\mathbb{E}\left[\mathrm{R}_n\right] = \sum_{i=1}^{L}\sum_{j=1}^{K}\sum_{k=j+1}^{K} \mathbb{E}\left[\mathbb{I}\{\widehat{\sigma}_{n,i}(j) > \widehat{\sigma}_{n,i}(k)\}\right] = \sum_{i=1}^{L}\sum_{j=1}^{K}\sum_{k=j+1}^{K} \mathbb{P}\left(\mathbf{x}_{i,k}^\top \widehat{\boldsymbol{\theta}}_n > \mathbf{x}_{i,j}^\top \widehat{\boldsymbol{\theta}}_n\right),$$

where $\mathbb{P}(\mathbf{x}_{i,k}^\top \widehat{\boldsymbol{\theta}}_n > \mathbf{x}_{i,j}^\top \widehat{\boldsymbol{\theta}}_n)$ is the probability of predicting a sub-optimal item k above item j in list i. We bound these probabilities from above by the sum of probabilities $\mathbb{P}(\mathbf{x}_{i,j}^\top(\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*) < -\Delta_{i,j,k})$ Finally, we bound these from above by $\exp\left(-\frac{n\Delta_{i,j,k}^2}{2d}\right)$ using a concentration inequality for sub-Gaussian random variables derived in Lemma 2. The full proof is in Appendix C.3. □

Each term in Theorem 4 can be bounded from above by $\exp\left(-n\Delta_{\min}^2/(2d)\right)$, where n is the sample size, d is the number of features, and $\Delta_{\min}$ is the minimum gap. Therefore, the bound decreases exponentially with budget n and gaps, and increases with d. This dependence is similar to existing sample complexity bounds for fixed-budget best-arm identification in linear models. Specifically, it is the same as in Theorem 1 of Azizi et al. [8]. Yang and Tan [101] derived a similar upper bound and a matching lower bound. The gaps $\Delta_{i,j,k}$ in Theorem 4 reflect the hardness of sorting list i, which depends on the differences of the mean rewards of items j and k in list i. Although we do not derive a matching lower bound to confirm this dependence, it is expected and not surprising.

5 Learning with Ranking Feedback

This section is organized similarly to Section 4. First, we present our algorithm, which logs ranking feedback using a policy computed by an optimal design and learns a model from it. Then we analyze it, by bounding its maximum prediction error in (4) and the expected ranking loss in (3).

5.1 Algorithm Dope

We present Dope for ranking feedback next. The algorithm is similar to Dope in Section 4 and has four main parts. First, we solve the optimal design problem in (6) to get a data logging policy π_*. The matrix for list i is $\mathbf{A}_i = [\mathbf{z}_{i,j,k}]_{(j,k)\in\Pi_2(K)} \in \mathbb{R}^{d\times K(K-1)/2}$, where $\mathbf{z}_{i,j,k} = \mathbf{x}_{i,j} - \mathbf{z}_{i,k}$ is the difference between feature vectors of items j and k in list i. The algebraic form of $\mathbf{A}_i$ arises from the covariance matrix of the estimator in (11). In particular, $\sum_{j=1}^{K}\sum_{k=j+1}^{K} \mathbf{z}_{i,j,k}\mathbf{z}_{i,j,k}^\top = \mathbf{Tr}(\mathbf{A}_i\mathbf{A}_i^\top)$. Second, the policy π_* is used to collect human feedback for n rounds. At round $t \in [n]$, the learner samples a list $I_t \sim \pi_*$ and observes σ_t drawn from the PL model in (2). Third, we compute the MLE of the unknown model parameter $\boldsymbol{\theta}_*$ as

$$\widehat{\boldsymbol{\theta}}_n \in \arg\min_{\boldsymbol{\theta}} \ell_n(\boldsymbol{\theta}), \quad \ell_n(\boldsymbol{\theta}) = -\frac{1}{n}\sum_{t=1}^{n}\sum_{k=1}^{K} \log\left(\frac{\exp\left(\mathbf{x}_{I_t,\sigma_t(k)}^\top \boldsymbol{\theta}\right)}{\sum_{j=k}^{K}\exp\left(\mathbf{x}_{I_t,\sigma_t(j)}^\top \boldsymbol{\theta}\right)}\right). \tag{9}$$

Finally, the learner orders the items in every list $i \in [L]$ according to their estimated mean rewards $\mathbf{x}_{i,k}^\top \widehat{\boldsymbol{\theta}}_n$ in descending order to obtain the permutation $\widehat{\sigma}_{n,i}$. We compute the MLE using *iteratively reweighted least squares (IRLS)* [93], a well-known second-order technique for *generalized linear models (GLMs)*. We present the full pseudo-code in Algorithm 2 in Appendix F.

For $K = 2$, (9) becomes logistic regression and so does the optimal design. The D-optimal design in generalized linear models was studied before. For instance, Azizi et al. [8] applied it to fixed-budget best-arm identification.

5.2 Maximum Prediction Error Under Ranking Feedback

In this section, we bound the prediction error of Dope under ranking feedback. Before we proceed, we make the following assumption, which we borrow from Zhu et al. [111].

Assumption 1 (Identifiability of $\boldsymbol{\theta}_*$). *We assume that the true parameter satisfies $\boldsymbol{\theta}_* \in \boldsymbol{\Theta}$, where*

$$\boldsymbol{\Theta} = \{\boldsymbol{\theta} \in \mathbb{R}^d : \boldsymbol{\theta}^\top \mathbf{1}_d = 0, \|\boldsymbol{\theta}\|_2 \leq 1\}. \tag{10}$$

We also assume that $\max_{i\in[L],k\in[K]} \|\mathbf{x}_{i,k}\| \leq 1$.

This assumption is common in the linear bandit literature [1, 51] and has been recently used in the context of K-wise ranking feedback in Zhu et al. [111].

Assumption 2. *We assume that* $\kappa = \inf_{\{\mathbf{x}:\|\mathbf{x}\|\leq 1, \boldsymbol{\theta}:\|\boldsymbol{\theta}-\boldsymbol{\theta}_*\|\leq 1\}} \exp\left(\mathbf{x}^\top \boldsymbol{\theta}\right) > 0.$

This amounts to assuming a lower bound κ on the derivative of the mean function. This is needed since learning in GLMs is hard when the slope of the mean function is low.

As before, and similarly to Theorem 4, we first bound the maximum prediction error by decomposing it into two parts, where one part captures the efficiency of the optimal design and the other part captures the uncertainty in the MLE $\widehat{\boldsymbol{\theta}}_n$. To measure the uncertainty of the MLE in our analysis, we define the normalized and unnormalized covariance matrices corresponding to the estimate,

$$\boldsymbol{\Sigma}_n = \frac{2}{K(K-1)n} \sum_{t=1}^{n} \sum_{j=1}^{K} \sum_{k=j+1}^{K} \mathbf{z}_{I_t,j,k} \mathbf{z}_{I_t,j,k}^\top \,, \quad \bar{\boldsymbol{\Sigma}}_n = \frac{K(K-1)n}{2} \boldsymbol{\Sigma}_n \,, \tag{11}$$

respectively. We bound the maximum prediction error next.

Theorem 5 (Maximum prediction error)**.** *Fix* $\delta \in [0, 1)$. *Then, with probability at least* $1 - \delta$, *the maximum prediction error after* n *rounds is bounded as*

$$\max_{i \in [L]} \mathbf{Tr}\left(\mathbf{A}_i^\top \left(\boldsymbol{\theta}_* - \widehat{\boldsymbol{\theta}}_n\right)\left(\boldsymbol{\theta}_* - \widehat{\boldsymbol{\theta}}_n\right)^\top \mathbf{A}_i\right) \leq \widetilde{O}\left(\frac{K^6 d^2 \log(1/\delta)}{n}\right).$$

This theorem is proved in Appendix D.1. We build on a self-normalizing bound of Zhu et al. [111], $\|\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*\|_{\boldsymbol{\Sigma}_n}^2 \leq O\left(K^4\left(\frac{d + \log(1/\delta)}{n}\right)\right)$, which may not be tight in K. If the bound can be improved by a multiplicative $c > 0$ in the future, we would get a multiplicative c improvement in Theorem 5. We remind the reader again that the sampling allocation for each list may not be an integer. In such cases, a separate rounding procedure [72] is needed, which would introduce an additional factor of $1 + \beta$ for some $\beta > 0$ in our bound. For simplicity, we omit this factor in our derivations.

5.3 Ranking Loss Under Ranking Feedback

In this section, we bound the expected ranking loss under ranking feedback. Similarly to Section 4.3, we define the *gap* between the mean rewards of items j and k in list i as $\Delta_{i,j,k} = (\mathbf{x}_{i,j} - \mathbf{x}_{i,k})^\top \boldsymbol{\theta}_*$.

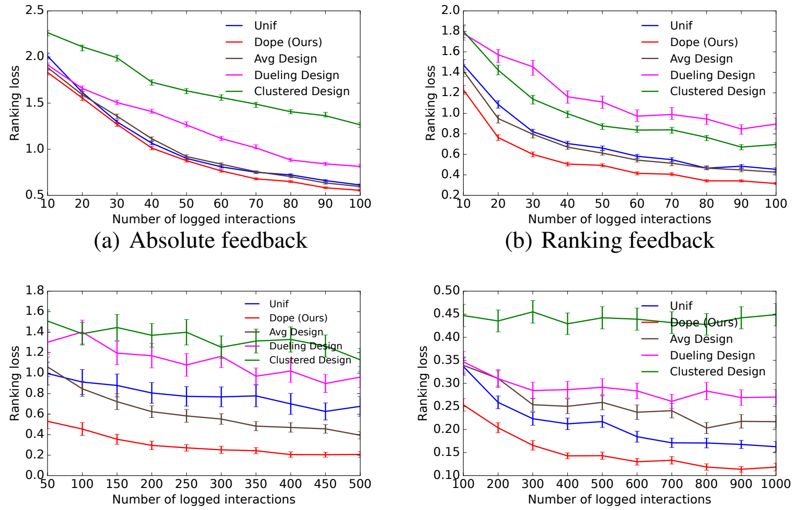
Theorem 6 (Ranking loss)**.** *The expected ranking loss under ranking feedback is bounded as*

$$\mathbb{E}\left[\mathrm{R}_n\right] \leq \sum_{i=1}^{L} \sum_{j=1}^{K} \sum_{k=j+1}^{K} 2 \exp\left(-\frac{n\kappa^2 \Delta_{i,j,k}^2}{2d}\right).$$

Proof. The proof is similar to Theorem 4. At the end of round n, we bound the probability that a sub-optimal item k is ranked above item j. This bound has two parts. First, for any $(j, k) \in \Pi_2(K)$, we show that $\mathbb{P}(\mathbf{x}_{i,j}^\top \widehat{\boldsymbol{\theta}}_n < \mathbf{x}_{i,k}^\top \widehat{\boldsymbol{\theta}}_n) = \mathbb{P}(\mathbf{z}_{i,j,k}^\top (\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*) < -\Delta_{i,j,k})$, where we introduce feature vector differences $\mathbf{z}_{i,j,k}$. Then we bound the above quantity by $2\exp\left(-\frac{n\kappa^2 \Delta_{i,j,k}^2}{2d}\right)$, using Lemma 10 in Appendix D.2 and Lemma 2. The full proof is in Appendix D.2. $\square$

The bound in Theorem 6 is similar to that in Theorem 4. The only difference is in the factor of κ, which appears in generalized linear bandit analyses [59, 62, 8]. This dependence arises since learning in GLMs is hard when the slope of the mean function is low.

Each term in Theorem 6 can be bounded by $\exp\left(-n\kappa^2 \Delta_{\min}^2/(2d)\right)$, where n is the sample size, d is the number of features, κ is defined in Assumption 2, and $\Delta_{\min}$ is the minimum gap. Therefore, similarly to Theorem 4, the bound decreases exponentially with budget n, κ, and gaps; and increases with d. This dependence is similar to existing sample complexity bounds for fixed-budget best-arm identification in GLMs. Specifically, it is similar to the result of Theorem 2 of Azizi et al. [8] which studies fixed budget BAI setting in linear and GLM bandits for absolute feedback.



(a) Absolute feedback (b) Ranking feedback
(c) Nectar dataset (d) Anthropic dataset

Figure 1: Ranking loss of all compared methods plotted as a function of the number of rounds. The confidence bars are one standard error of the estimate.

6 Experiments

The goal of our experiments is to evaluate Dope empirically and compare it to several baselines. All compared methods estimate $\widehat{\boldsymbol{\theta}}_t$ using (7) and (9), depending on the feedback. Then they rank items in the lists based on their estimated mean rewards $\mathbf{x}_{i,k}^\top \widehat{\boldsymbol{\theta}}_t$. The performance of methods is evaluated by the ranking loss in (3) divided by L. All experiments are averaged over 100 independent runs.

We compare the following algorithms:

(1) Dope (**D-o**ptimal design): This is our proposed approach. We solve the optimal design problem in (6) and then sample lists I_t according to π_*.

(2) Unif (uniform sampling): This approach chooses lists I_t uniformly at random from $[L]$. While simple, it is known to be competitive in real-world problems where feature vectors may cover the feature space close to uniformly [5, 103, 4, 77].

(3) Avg-Design: The exploration policy is an optimal design over feature vectors. The feature vector of list i is the mean of the feature vectors of all items in it, $\bar{\mathbf{x}}_i = \frac{1}{K} \sum_{k=1}^K \mathbf{x}_{i,k}$. After the design is computed, we sample lists I_t according to it. The rest is the same as in Dope. This baseline shows that our list representation with multiple feature vectors can outperform more naive choices.

(4) Clustered-Design: This approach uses the same representation as Avg-Design. The difference is that we cluster the list feature vectors using k-medoids. Then we sample lists I_t uniformly at random from the cluster centers. The rest is the same as in Avg-Design. This baseline shows that we can outperform other notions of diversity, such as obtained by clustering. We tuned k and report the best results. It is $k = 10$ for the Nectar dataset and $k = 6$ otherwise.

(5) Dueling-Design: We turn L lists into $\binom{K}{2} L$ lists of length 2, one for each pair of items in the original lists. Then we apply Dope for $K = 2$. Specifically, we get pairwise feedback on the lists and learn a BTL model, which is a special case of the PL model in (9) for $K = 2$. The evaluation is the same as in the other methods. This baseline shows that pairwise feedback gathers less information than K-way feedback.

Pure exploration algorithms are often compared to cumulative regret baselines [15, 7]. Since our problem is a form of learning to rank, this suggests that we could compare to *online learning to rank (OLTR)* baselines [73, 47, 115]. We do not compare to them for the following reason. The problem of an optimal design over lists is to design a distribution over queries. All OLTR algorithms solve a different problem, return a ranked list of items conditioned on a query chosen by the environment. Since they do not choose queries, they cannot solve our problem.

Synthetic experiment 1 (absolute feedback): We have $L = 400$ questions and represent them by random vectors $\mathbf{q}_i \in [0,1]^6$. Each question has $K = 4$ answers. For each question, we generate K random answers $\mathbf{a}_{i,k} \in [1,2]^6$. Both the question and answer vectors are normalized to unit length. For each question-answer pair (i,k), the feature vector is $\mathbf{x}_{i,k} = \mathrm{vec}(\mathbf{q}_i \mathbf{a}_{i,k}^\top)$ and has length $d = 36$. The outer product captures cross-interaction terms of the question and answer representations. A similar technique is used for feature preprocessing of the Yahoo! Front Page Today Module User Click Log Dataset by [57, 58, 112, 9]. We randomly sample $\boldsymbol{\theta}_* \in [0,1]^d$. The absolute feedback is generated as in (1). Our results are reported in Figure 1(a). We observe that the ranking loss of Dope decreases the fastest among all methods, with Unif and Avg-Design being close second.

Synthetic experiment 2 (ranking feedback): This experiment is similar to the first experiment, except that the feedback is generated by the PL model in (2). Our results are reported in Figure 1(b) and we observe again that the ranking loss of Dope decreases the fastest. The two closest baselines are Unif and Avg-Design. Their lowest ranking loss ($n = 100$) is attained by Dope at $n = 60$, which is nearly a two-fold reduction in sample size.

Real-world experiment 3 (Nectar dataset): We take $L = 2000$ questions from the Nectar dataset [110]. Each question has $K = 5$ answers generated by gpt-4, gpt-4-0613, gpt-3.5-turbo, gpt-3.5-turbo-instruct, and anthropic. We first obtain 768-dimensional Instructor embeddings [84] of both questions and answers. Then we project them to $\mathbb{R}^{10}$ using a random projection matrix. Let $\mathbf{q}_i$ and $\mathbf{a}_{i,k}$ be the projected embeddings of question i and answer k to it. For each question-answer pair (i,k), the feature vector is $\mathbf{x}_{i,k} = \mathrm{vec}(\mathbf{q}_i \mathbf{a}_{i,k}^\top)$ and has length $d = 100$. We estimate $\boldsymbol{\theta}_* \in \mathbb{R}^d$ from the original ranking feedback in the dataset using the MLE in (9). During simulation, the ranking feedback is generated by the PL model in (2). Our results are reported in Figure 1(c). We observe that the ranking loss of Dope is consistently the lowest. The closest baseline is Avg-Design. Its lowest ranking loss ($n = 500$) is attained by Dope at $n = 150$, which is more than a three-fold reduction in sample size.

Real-world experiment 4 (Anthropic dataset): We take $L = 2000$ questions, each with $K = 2$ answers, from the Anthropic dataset [10]. We again obtain 768-dimensional Instructor embeddings of all questions and answers. Then we project them to $\mathbb{R}^6$ using a random projection matrix. For each question-answer pair (i,k), the feature vector is $\mathbf{x}_{i,k} = \mathrm{vec}(\mathbf{q}_i \mathbf{a}_{i,k}^\top)$ and has length $d = 36$. We estimate $\boldsymbol{\theta}_* \in \mathbb{R}^d$ from the original ranking feedback in the dataset using the MLE in (9). During simulation, the ranking feedback is generated by the PL model in (2). Our results are reported in Figure 1(d). We observe again that the ranking loss of Dope is the lowest. The closest baseline is Unif. Its lowest ranking loss ($n = 1000$) is attained by Dope at $n = 300$, which is more than a three-fold reduction in sample size.

7 Conclusions

We study the problem of optimal human feedback collection for learning preference models. The problem is formalized as learning to rank K answers to L questions under a fixed budget n on the number of asked questions. To our knowledge, this is the first paper on fixed-budget pure exploration for ranked lists based on optimal design. We consider two settings: absolute and ranking feedback. The absolute setting is motivated by how human annotators assign relevance scores in search [34, 63]. The ranking feedback is motivated by learning preference models in RLHF [44, 74, 41, 18, 102, 82, 20]. We solve both settings in a unified way. The key idea in our general solution is extending optimal designs [46, 51], which can be used to compute optimal information gathering policies, to K-way questions. After the human feedback is collected, we learn a model of human preferences using an existing MLE. This approach is statistically efficient, computationally efficient, and can be analyzed. Specifically, in both absolute and ranking feedback models, we bound the estimation errors of our algorithms and the resulting ranking loss. We experiment with several synthetic and real-world datasets to show that our approach is practical.

In the future, we want to extend our work in several directions. For instance, while we proved upper bounds on the ranking loss with absolute and ranking feedback, and discussed them in detail, we did not prove matching lower bounds. We intend to prove them. We also want to extend our approach to the fixed-confidence setting, with both absolute and ranking feedback. Finally, we want to apply our approach to learning a reward model for fine-tuning an LLM.

References

[1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. *Advances in neural information processing systems*, 24, 2011.

[2] Pieter Abbeel and Andrew Y Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1, 2004.

[3] Riad Akrour, Marc Schoenauer, and Michèle Sebag. April: Active preference learning-based reinforcement learning. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II 23*, pages 116–131. Springer, 2012.

[4] Jordan Ash, Surbhi Goel, Akshay Krishnamurthy, and Sham Kakade. Gone fishing: Neural active learning with fisher embeddings. *Advances in Neural Information Processing Systems*, 34, 2021.

[5] Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. *arXiv preprint arXiv:1906.03671*, 2019.

[6] Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Exploration–exploitation tradeoff using variance estimates in multi-armed bandits. *Theoretical Computer Science*, 410(19): 1876–1902, 2009.

[7] Jean-Yves Audibert, Sébastien Bubeck, and Rémi Munos. Best arm identification in multi-armed bandits. In *COLT*, pages 41–53, 2010.

[8] Mohammad Javad Azizi, Branislav Kveton, and Mohammad Ghavamzadeh. Fixed-budget best-arm identification in structured bandits. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence*, 2022.

[9] Jackie Baek and Vivek Farias. Ts-ucb: Improving on thompson sampling with little to no additional computation. In *International Conference on Artificial Intelligence and Statistics*, pages 11132–11148. PMLR, 2023.

[10] 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.

[11] Viktor Bengs, Róbert Busa-Fekete, Adil El Mesaoudi-Paul, and Eyke Hüllermeier. Preference-based online learning with dueling bandits: A survey. *The Journal of Machine Learning Research*, 22(1):278–385, 2021.

[12] Erdem Bıyık, Nicolas Huynh, Mykel J Kochenderfer, and Dorsa Sadigh. Active preference-based gaussian process regression for reward learning. *arXiv preprint arXiv:2005.02575*, 2020.

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

[14] 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*, pages 783–792. PMLR, 2019.

[15] Sebastien Bubeck, Remi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *Proceedings of the 20th International Conference on Algorithmic Learning Theory*, pages 23–37, 2009.

[16] Christopher Burges. From RankNet to LambdaRank to LambdaMART: An overview. Technical Report MSR-TR-2010-82, Microsoft Research, 2010.

[17] Róbert Busa-Fekete, Balázs Szörényi, Paul Weng, Weiwei Cheng, and Eyke Hüllermeier. Preference-based reinforcement learning: evolutionary direct policy search using a preference-based racing algorithm. *Machine learning*, 97:327–351, 2014.

[18] Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*, 2023.

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

[20] Zhuotong Chen, Yifei Ma, Branislav Kveton, and Anoop Deoras. Active learning with crowd sourcing improves information retrieval. In *ICML 2023 Workshop on Interactive Learning with Implicit Human Feedback*, 2023.

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

[22] Aleksandr Chuklin, Ilya Markov, and Maarten De Rijke. *Click models for web search*. Springer Nature, 2022.

[23] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[24] Miroslav Dudik, Dumitru Erhan, John Langford, and Lihong Li. Doubly robust policy evaluation and optimization. *Statistical Science*, 29(4):485–511, 2014.

[25] Beyza Ermis, Patrick Ernst, Yannik Stein, and Giovanni Zappella. Learning to rank in the position based model with bandit feedback. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2405–2412, 2020.

[26] Valerii Vadimovich Fedorov. *Theory of optimal experiments*. Elsevier, 2013.

[27] Tanner Fiez, Lalit Jain, Kevin G Jamieson, and Lillian Ratliff. Sequential experimental design for transductive linear bandits. *Advances in neural information processing systems*, 32, 2019.

[28] Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on Robot Learning*, pages 158–168. PMLR, 2022.

[29] Johannes Fürnkranz and Eyke Hüllermeier. Pairwise preference learning and ranking. In *European conference on machine learning*, pages 145–156. Springer, 2003.

[30] Alyssa Glass. Explaining preference learning, 2006.

[31] Adam Gleave and Geoffrey Irving. Uncertainty estimation for language reward models. *arXiv preprint arXiv:2203.07472*, 2022.

[32] Joey Hejna and Dorsa Sadigh. Inverse preference learning: Preference-based rl without a reward function. *arXiv preprint arXiv:2305.15363*, 2023.

[33] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29, 2016.

[34] Katja Hofmann, Shimon Whiteson, and Maarten de Rijke. Fidelity, soundness, and efficiency of interleaved comparison methods. *ACM Transactions on Information Systems*, 31(4):1–43, 2013.

[35] Neil Houlsby, Ferenc Huszár, Zoubin Ghahramani, and Máté Lengyel. Bayesian active learning for classification and preference learning. *arXiv preprint arXiv:1112.5745*, 2011.

[36] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Computing Surveys (CSUR)*, 50(2):1–35, 2017.

[37] Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *International conference on machine learning*, pages 427–435. PMLR, 2013.

[38] Kevin Jamieson and Lalit Jain. Interactive machine learning. 2022.

[39] Xiang Ji, Huazheng Wang, Minshuo Chen, Tuo Zhao, and Mengdi Wang. Provable benefits of policy learning from human preferences in contextual bandit problems. *arXiv preprint arXiv:2307.12975*, 2023.

[40] Ying Jin, Zhuoran Yang, and Zhaoran Wang. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pages 5084–5096. PMLR, 2021.

[41] Yachen Kang, Diyuan Shi, Jinxin Liu, Li He, and Donglin Wang. Beyond reward: Offline preference-guided policy optimization, 2023.

[42] Julian Katz-Samuels, Lalit Jain, Kevin G Jamieson, et al. An empirical process approach to the union bound: Practical algorithms for combinatorial and linear bandits. *Advances in Neural Information Processing Systems*, 33:10371–10382, 2020.

[43] Julian Katz-Samuels, Jifan Zhang, Lalit Jain, and Kevin Jamieson. Improved algorithms for agnostic pool-based active classification. In *International Conference on Machine Learning*, pages 5334–5344. PMLR, 2021.

[44] Timo Kaufmann, Paul Weng, Viktor Bengs, and Eyke Hüllermeier. A survey of reinforcement learning from human feedback, 2024.

[45] Maurice George Kendall. Rank correlation methods. 1948.

[46] Jack Kiefer and Jacob Wolfowitz. The equivalence of two extremum problems. *Canadian Journal of Mathematics*, 12:363–366, 1960.

[47] Branislav Kveton, Csaba Szepesvari, Zheng Wen, and Azin Ashkan. Cascading bandits: Learning to rank in the cascade model. In *Proceedings of the 32nd International Conference on Machine Learning*, 2015.

[48] Branislav Kveton, Csaba Szepesvari, Zheng Wen, and Azin Ashkan. Cascading bandits: Learning to rank in the cascade model. In *International conference on machine learning*, pages 767–776. PMLR, 2015.

[49] Paul Lagrée, Claire Vernade, and Olivier Cappe. Multiple-play bandits in the position-based model. *Advances in Neural Information Processing Systems*, 29, 2016.

[50] Anusha Lalitha Lalitha, Kousha Kalantari, Yifei Ma, Anoop Deoras, and Branislav Kveton. Fixed-budget best-arm identification with heterogeneous reward variances. In *Uncertainty in Artificial Intelligence*, pages 1164–1173. PMLR, 2023.

[51] Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

[52] Tor Lattimore, Branislav Kveton, Shuai Li, and Csaba Szepesvari. TopRank: A practical algorithm for online stochastic ranking. In *Advances in Neural Information Processing Systems 31*, pages 3949–3958, 2018.

[53] Kimin Lee, Laura Smith, Anca Dragan, and Pieter Abbeel. B-pref: Benchmarking preference-based reinforcement learning. *arXiv preprint arXiv:2111.03026*, 2021.

[54] Tyler Lekang and Andrew Lamperski. Simple algorithms for dueling bandits. *arXiv preprint arXiv:1906.07611*, 2019.

[55] John R Lepird, Michael P Owen, and Mykel J Kochenderfer. Bayesian preference elicitation for multiobjective engineering design optimization. *Journal of Aerospace Information Systems*, 12(10):634–645, 2015.

[56] Gene Li, Cong Ma, and Nati Srebro. Pessimism for offline linear contextual bandits using
$backslash$
ell
$_p$ confidence sets. *Advances in Neural Information Processing Systems*, 35:20974–20987, 2022.

[57] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 661–670, 2010.

[58] Lihong Li, Wei Chu, John Langford, and Xuanhui Wang. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In *Proceedings of the fourth ACM international conference on Web search and data mining*, pages 297–306, 2011.

[59] Lihong Li, Yu Lu, and Dengyong Zhou. Provably optimal algorithms for generalized linear contextual bandits. In *International Conference on Machine Learning*, pages 2071–2080. PMLR, 2017.

[60] Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

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

[62] Blake Mason, Kwang-Sung Jun, and Lalit Jain. An experimental design approach for regret minimization in logistic bandits. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7736–7743, 2022.

[63] MS MARCO. MS MARCO Dataset. https://microsoft.github.io/msmarco/, 2016.

[64] Subhojyoti Mukherjee, Ardhendu S Tripathy, and Robert Nowak. Chernoff sampling for active testing and extension to active regression. In *International Conference on Artificial Intelligence and Statistics*, pages 7384–7432. PMLR, 2022.

[65] Gergely Neu and Csaba Szepesvári. Training parsers by inverse reinforcement learning. *Machine learning*, 77:303–337, 2009.

[66] Andrew Y Ng, Stuart Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, volume 1, page 2, 2000.

[67] Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

[68] Ellen Novoseller, Yibing Wei, Yanan Sui, Yisong Yue, and Joel Burdick. Dueling posterior sampling for preference-based reinforcement learning. In *Conference on Uncertainty in Artificial Intelligence*, pages 1029–1038. PMLR, 2020.

[69] Kweku A Opoku-Agyemang. Randomized controlled trials via reinforcement learning from human feedback. 2023.

[70] 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.

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

[72] Friedrich Pukelsheim. *Optimal design of experiments*. SIAM, 2006.

[73] Filip Radlinski, Robert Kleinberg, and Thorsten Joachims. Learning diverse rankings with multi-armed bandits. In *Proceedings of the 25th International Conference on Machine Learning*, pages 784–791, 2008.

[74] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=HPuSIXJaa9.

[75] Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. In *IJCAI*, volume 7, pages 2586–2591, 2007.

[76] Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. *Advances in Neural Information Processing Systems*, 34:11702–11716, 2021.

[77] Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM Computing Surveys (CSUR)*, 54 (9):1–40, 2021.

[78] Dorsa Sadigh, Anca Dragan, Shankar Sastry, and Sanjit Seshia. *Active preference-based learning of reward functions*. 2017.

[79] Aadirupa Saha and Pierre Gaillard. Versatile dueling bandits: Best-of-both-world analyses for online learning from preferences. *arXiv preprint arXiv:2202.06694*, 2022.

[80] Aadirupa Saha, Aldo Pacchiano, and Jonathan Lee. Dueling rl: Reinforcement learning with trajectory preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 6263–6289. PMLR, 2023.

[81] Ayush Sekhari, Karthik Sridharan, Wen Sun, and Runzhe Wu. Contextual bandits and imitation learning with preference-based active queries. *Advances in Neural Information Processing Systems*, 36, 2024.

[82] Tianhao Shen, Renren Jin, Yufei Huang, Chuang Liu, Weilong Dong, Zishan Guo, Xinwei Wu, Yan Liu, and Deyi Xiong. Large language model alignment: A survey. *arXiv preprint arXiv:2309.15025*, 2023.

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

[84] Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen-tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. One embedder, any task: Instruction-finetuned text embeddings. 2022. URL https://arxiv.org/abs/2212.09741.

[85] Yanan Sui, Masrour Zoghi, Katja Hofmann, and Yisong Yue. Advancements in dueling bandits. In *IJCAI*, pages 5502–5510, 2018.

[86] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[87] Adith Swaminathan and Thorsten Joachims. Counterfactual risk minimization: Learning from logged bandit feedback. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 814–823, 2015.

[88] Balázs Szörényi, Róbert Busa-Fekete, Adil Paul, and Eyke Hüllermeier. Online rank elicitation for plackett-luce: A dueling bandits approach. *Advances in neural information processing systems*, 28, 2015.

[89] EM Voorhees. Proceedings of the 8th text retrieval conference. *TREC-8 Question Answering Track Report*, pages 77–82, 1999.

[90] Yining Wang, Liwei Wang, Yuanzhi Li, Di He, and Tie-Yan Liu. A theoretical analysis of ndcg type ranking measures. In *Conference on learning theory*, pages 25–54. PMLR, 2013.

[91] Yuanhao Wang, Qinghua Liu, and Chi Jin. Is rlhf more difficult than standard rl? a theoretical perspective. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[92] Christian Wirth, Riad Akrour, Gerhard Neumann, Johannes Fürnkranz, et al. A survey of preference-based reinforcement learning methods. *Journal of Machine Learning Research*, 18 (136):1–46, 2017.

[93] R. Wolke and H. Schwetlick. Iteratively reweighted least squares: Algorithms, convergence analysis, and numerical comparisons. *SIAM Journal on Scientific and Statistical Computing*, 9 (5):907–921, 1988.

[94] Runzhe Wu and Wen Sun. Making rl with preference-based feedback efficient via randomization. *arXiv preprint arXiv:2310.14554*, 2023.

[95] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th international conference on Machine learning*, pages 1192–1199, 2008.

[96] Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent pessimism for offline reinforcement learning. *Advances in neural information processing systems*, 34:6683–6694, 2021.

[97] Tengyang Xie, Nan Jiang, Huan Wang, Caiming Xiong, and Yu Bai. Policy finetuning: Bridging sample-efficient offline and online reinforcement learning. *Advances in neural information processing systems*, 34:27395–27407, 2021.

[98] Wei Xiong, Hanze Dong, Chenlu Ye, Han Zhong, Nan Jiang, and Tong Zhang. Gibbs sampling from human feedback: A provable kl-constrained framework for rlhf. *arXiv preprint arXiv:2312.11456*, 2023.

[99] Tengyu Xu, Yue Wang, Shaofeng Zou, and Yingbin Liang. Provably efficient offline reinforcement learning with trajectory-wise reward. *arXiv preprint arXiv:2206.06426*, 2022.

[100] Yichong Xu, Ruosong Wang, Lin Yang, Aarti Singh, and Artur Dubrawski. Preference-based reinforcement learning with finite-time guarantees. *Advances in Neural Information Processing Systems*, 33:18784–18794, 2020.

[101] Junwen Yang and Vincent YF Tan. Minimax optimal fixed-budget best arm identification in linear bandits. *arXiv preprint arXiv:2105.13017*, 2021.

[102] Kevin Yang, Dan Klein, Asli Celikyilmaz, Nanyun Peng, and Yuandong Tian. Rlcd: Reinforcement learning from contrast distillation for language model alignment. *arXiv preprint arXiv:2307.12950*, 2023.

[103] Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. Cold-start active learning through self-supervised language modeling. *arXiv preprint arXiv:2010.09535*, 2020.

[104] 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.

[105] Andrea Zanette. When is realizability sufficient for off-policy reinforcement learning? In *International Conference on Machine Learning*, pages 40637–40668. PMLR, 2023.

[106] Andrea Zanette, Martin J Wainwright, and Emma Brunskill. Provable benefits of actor-critic methods for offline reinforcement learning. *Advances in neural information processing systems*, 34:13626–13640, 2021.

[107] Wenhao Zhan, Masatoshi Uehara, Nathan Kallus, Jason D Lee, and Wen Sun. Provable offline preference-based reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2023.

[108] Zixin Zhong, Wang Chi Chueng, and Vincent YF Tan. Thompson sampling algorithms for cascading bandits. *The Journal of Machine Learning Research*, 22(1):9915–9980, 2021.

[109] Tianchen Zhou, Jia Liu, Yang Jiao, Chaosheng Dong, Yetian Chen, Yan Gao, and Yi Sun. Bandit learning to rank with position-based click models: Personalized and equal treatments. *arXiv preprint arXiv:2311.04528*, 2023.

[110] Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, and Jiantao Jiao. Starling-7b: Improving llm helpfulness & harmlessness with rlaif, November 2023.

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

[112] Yinglun Zhu, Dongruo Zhou, Ruoxi Jiang, Quanquan Gu, Rebecca Willett, and Robert Nowak. Pure exploration in kernel and neural bandits. *Advances in neural information processing systems*, 34:11618–11630, 2021.

[113] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, Anind K Dey, et al. Maximum entropy inverse reinforcement learning. In *Aaai*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

[114] Masrour Zoghi, Tomas Tunys, Mohammad Ghavamzadeh, Branislav Kveton, Csaba Szepesvari, and Zheng Wen. Online learning to rank in stochastic click models. In *International conference on machine learning*, pages 4199–4208. PMLR, 2017.

[115] Shi Zong, Hao Ni, Kenny Sung, Nan Rosemary Ke, Zheng Wen, and Branislav Kveton. Cascading bandits for large-scale recommendation problems. *arXiv preprint arXiv:1603.05359*, 2016.

A Related Works

In this section, we discuss related works. Our setting is similar to the preference learning framework. In preference learning framework the goal is to learn the preferences of one or more agents from the observations [29, 30, 35]. The preference learning literature mainly consists of two types of feedback: pairwise preferences and ranking feedback. In pairwise preference, the observed data is a preference between pairs of objects while in ranking feedback the learner observes the absolute ranking of the items. Both of these have been studied in the online and bandit communities [48, 114, 25, 11]. The preference based bandits have several similarities with the dueling bandit settings [104, 54, 79, 80]. In dueling bandits at every round t the learner selects a pair of items and observes an instantaneous comparison over them where the outcome does not depend on the items previously selected. So in dueling bandits the goal of the learner is to select the item winning with the highest probability. In contrast in our setting, the learner observes the absolute feedback for K items or a ranking over these K items.

In the ranking setting the learner observed a ranking over the items and the goal of the learner is to find the K items with the highest reward [111]. These rankings can be generated from an underlying human preference model like Placket-Luce (PL) [71, 61] or BTL model [13]. However, in this work, we focus on the fixed budget pure exploration setting whereas all the previous works focus on simple regret setting under the ranking feedback [48, 114]. Previous works in online learning to rank have focused on several types of click feedback models, like position based model [49, 25, 109] or cascading model [47, 115, 108]. In contrast in our setting, we do not assume any such underlying click model, but assume that there is an underlying human ranking model (PL or BTL).

Similar preference based learning has been studied in Reinforcement Learning (RL) as well [92, 68, 100, 32]. This is termed Preference Based Reinforcement Learning (PBRL). The key difference between RL and PBRL settings is that in PBRL the learner has to learn the underlying human preference through the rewards observed which can be non-numerical [21, 53, 19]. All of these works focus on regret minimization or finding the optimal policy in RL setting. However, in our setting, we only consider the stateless bandit framework, and we focus pure exploration setting. The PBRL setting has also been studied under general function approximation when the reward is parameterized by a neural network [95, 60, 17, 14, 70, 83].

Our work is also closely related to Inverse Reinforcement Learning (IRL) and Offline Reinforcement Learning. These frameworks also allow the agent to take into account human preferences into it decision-making process. In IRL and imitation learning the agent only observes the expert's interaction history and aims to predict the expert's preference [66, 2, 75, 113, 65, 33, 36, 28]. In the offline RL setting the agent directly observes the past history of interactions. Note that these actions can be sub-optimal and there can be issues of data coverage and distribution shifts. Therefore in recent years pessimism under offline RL has gained traction [40, 76, 97, 106, 96, 56, 99, 105, 107, 81]. In contrast to these works, we study offline K-wise preference ranking under PL and BTL models for pure exploration setting. We do not use any pessimism but use optimal design [72, 26] to ensure diversity among the data collected.

B Proof of Matrix Kiefer-Wolfowitz

We follow the proof technique of Lattimore and Szepesvári [51]. Observe that $\mathbf{V}_\pi \in \mathbb{R}^{d\times d}$ is a square matrix. Using Jacobi formula we have

$$\nabla f(\pi)_{\pi(i)} = \frac{\mathbf{Tr}\left(\mathrm{adj}(\mathbf{V}_\pi)\mathbf{A}_i\mathbf{A}_i^\top\right)}{\det(\mathbf{V}_\pi)} \tag{12}$$

$$= \frac{\mathbf{Tr}(\mathbf{A}_i^\top \mathrm{adj}(\mathbf{V}_\pi)\mathbf{A}_i)}{\det(\mathbf{V}_\pi)} \tag{13}$$

$$\overset{(a)}{=} \mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_\pi^{-1}\mathbf{A}_i), \tag{14}$$

where the last equality follows from the fact that for a square matrix $\mathbf{V}_\pi$, its adjoint matrix $\mathrm{adj}(\mathbf{V}_\pi)$ is the transpose of its cofactor matrix and hence, the inverse is $\mathbf{V}_\pi^{-1} = \frac{1}{\det(\mathbf{V}_\pi)}\mathrm{adj}(V_\pi)^\top$. Then, we have

$$\sum_{i\in[L]} \pi(i)\mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_\pi^{-1}\mathbf{A}_i) = \mathbf{Tr}\left(\sum_{i\in[L]} \pi(i)\mathbf{A}_i\mathbf{A}_i^\top \mathbf{V}_\pi^{-1}\right) \tag{15}$$

$$= \mathbf{Tr}\left(\sum_{i\in[L]} \pi(i)\mathbf{A}_i\mathbf{A}_i^\top \left(\sum_{i\in[L]} \pi(i)\mathbf{A}_i\mathbf{A}_i^\top\right)^{-1}\right) \tag{16}$$

$$= \mathbf{Tr}\left(\mathbf{I}_d\right) = d. \tag{17}$$

The above equation implies $g(\pi) \geq d$ for all π.
$(b) \Rightarrow (a)$: Suppose that π_* is a maximizer of $f(\pi)$. By the first-order optimality criterion, for any π distribution,

$$\begin{aligned}
0 &\geq \langle \nabla f\left(\pi_*\right), \pi - \pi_* \rangle \\
&\geq \left(\sum_{i\in[L]} \pi(i)\mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_{\pi_*}^{-1}\mathbf{A}_i) - \sum_{i\in[L]} \pi_*(i)\mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_{\pi_*}^{-1}\mathbf{A}_i)\right) \\
&\geq \left(\sum_{i\in[L]} \pi(i)\mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_{\pi_*}^{-1}\mathbf{A}_i) - d\right).
\end{aligned}$$

For an arbitrary π, choosing π to be the Dirac at i proves that $\mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_{\pi_*}^{-1}\mathbf{A}_i) \leq d$ for all $i \in [L]$. Since $g(\pi) \geq d$ for all π by (17), it follows that π_* is a minimizer of g and that $g(\pi_*) = d$.

$(c) \implies (b)$: Suppose that $g\left(\pi_*\right) = d$. Then, for any π,

$$\langle \nabla f\left(\pi_*\right), \pi - \pi_* \rangle = \sum_{i\in[L]]} \pi(i)\mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_{\pi_*}^{-1}\mathbf{A}_i) - d \leq 0. \tag{18}$$

And it follows that π_* is a maximizer of $f(\pi)$ by the first-order optimality conditions and the concavity of $f(\pi)$.

$(a) \implies (c)$: Follows from the previous two steps as we proved that π_* is a minimizer of $g(\pi)$ and π_* is also a maximizer of $f(\pi)$.

To prove the second part of the theorem, let π_* be a minimizer of g, which by the previous part is a maximizer of f. Let $\mathcal{M} = \mathrm{Supp}\left(\pi_*\right)$, and suppose that $|\mathcal{M}| > d(d+1)/2$. Since the dimension of the subspace of $d \times d$ symmetric matrices is $d(d+1)/2$, there must be a non-zero function $v : \mathcal{S} \to \mathbb{R}^L$ with $\mathrm{Supp}(v) \subseteq \mathcal{M}$ such that

$$\sum_{i\in\mathcal{M}} v(i)\mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_\pi^{-1}\mathbf{A}_i) = 0. \tag{19}$$

where, $v(i)$ is the probability assigned to the $\mathbf{a}$ under the function (distribution) v. Notice that for any $j \in \mathcal{M}$, the first-order optimality conditions ensure that $\mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_\pi^{-1}\mathbf{A}_i) = d$. Hence we can show that

$$d\sum_{i\in\mathcal{M}} v(i) = \sum_{i\in\mathcal{M}} v(i)\mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_\pi^{-1}\mathbf{A}_i) = 0,$$

where the last equality follows from (19). Let $\pi(t) = \pi_* + tv$ and let $\tau = \max\left\{t > 0 : \pi(t) \in \Delta^{|\mathcal{S}|}\right\}$, which exists since $v \neq 0$ and $\sum_{i\in\mathcal{M}} v(i) = \mathbf{0}$ and $\mathrm{Supp}(v) \subseteq \mathcal{M}$. By (19), $\mathbf{V}_{\pi(t)} = \mathbf{V}_{\pi_*}$, and hence $f(\pi(\tau)) = f\left(\pi_*\right)$, which means that $\pi(\tau)$ also maximises f. The claim follows by checking that $|\mathrm{Supp}(\pi(t))| < |\mathrm{Supp}\left(\pi_*\right)|$ and then using induction. The claim of the theorem follows.

C Learning with Absolute Feedback

C.1 Proof of Lemma 2

We start by noting that for all $i \in [L]$,

$$\sum_{\mathbf{a}\in\mathbf{A}_i} \|\mathbf{a}\|^2_{\overline{\mathbf{\Sigma}}_n^{-1}} = \mathbf{Tr}(\mathbf{A}_i^\top \overline{\mathbf{\Sigma}}_n^{-1}\mathbf{A}_i).$$

Since we assume that $n\pi_*(i)$ is an integer, the covariance matrix $\mathbf{\Sigma}_n$ is invertible. This is because the optimal design π_* outputs a set of lists that spans $\mathbb{R}^d$ and avoids degenerate solutions. Then, we can rewrite the above as

$$\sum_{\mathbf{a} \in \mathbf{A}_i} \|\mathbf{a}\|^2_{\overline{\mathbf{\Sigma}}_n^{-1}} = \mathbf{Tr}(\mathbf{A}_i^\top \overline{\mathbf{\Sigma}}_n^{-1} \mathbf{A}_i) = \mathbf{Tr}\left(\mathbf{A}_i^\top \left(\sum_{t=1}^{n} \sum_{k=1}^{K} \mathbf{x}_{I_t,k} \mathbf{x}_{I_t,k}^\top\right)^{-1} \mathbf{A}_i\right)$$
$$\stackrel{(a)}{=} \frac{1}{n} \mathbf{Tr}\left(\mathbf{A}_i^\top \left(\sum_{i=1}^{L} \pi_*(i) \sum_{k=1}^{K} \mathbf{x}_{i,k} \mathbf{x}_{i,k}^\top\right)^{-1} \mathbf{A}_i\right) = \frac{1}{n} \mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_{\pi_*}^{-1} \mathbf{A}_i),$$

where (a) follows from the fact that given a fixed design π_* and budget n, list i is seen exactly $n\pi_*(i)$ times. Now we apply Theorem 1, use our definition of $g(\pi_*)$, and get that $g(\pi_*) = d$. Thus

$$\max_{i \in [L]} \mathbf{Tr}(\mathbf{A}_i^\top \overline{\mathbf{\Sigma}}_n^{-1} \mathbf{A}_i) = \frac{d}{n}.$$

The claim of the lemma follows.

C.2 Proof of Theorem 3

We start with

$$\max_{i \in [L]} \mathbf{Tr}\left(\mathbf{A}_i^\top \left(\boldsymbol{\theta}_* - \widehat{\boldsymbol{\theta}}_n\right)\left(\boldsymbol{\theta}_* - \widehat{\boldsymbol{\theta}}_n\right)^\top \mathbf{A}_i\right) = \max_{i \in [L]} \sum_{\mathbf{a} \in \mathbf{A}_i} \left(\mathbf{a}^\top \left(\boldsymbol{\theta}_* - \widehat{\boldsymbol{\theta}}_n\right)\right)^2$$
$$= \max_{i \in [L]} \sum_{\mathbf{a} \in \mathbf{A}_i} \left(\mathbf{a}^\top \overline{\mathbf{\Sigma}}_n^{-1/2} \overline{\mathbf{\Sigma}}_n^{1/2} \left(\boldsymbol{\theta}_* - \widehat{\boldsymbol{\theta}}_n\right)\right)^2 \stackrel{(a)}{\leq} \max_{i \in [L]} \sum_{\mathbf{a} \in \mathbf{A}_i} \|\mathbf{a}\|^2_{\overline{\mathbf{\Sigma}}_n^{-1}} \|\boldsymbol{\theta}_* - \widehat{\boldsymbol{\theta}}_n\|^2_{\overline{\mathbf{\Sigma}}_n}$$
$$\stackrel{(b)}{=} \underbrace{\max_{i \in [L]} \sum_{\mathbf{a} \in \mathbf{A}_i} \|\mathbf{a}\|^2_{\overline{\mathbf{\Sigma}}_n^{-1}}}_{\textbf{Part I}} \underbrace{n \|\boldsymbol{\theta}_* - \widehat{\boldsymbol{\theta}}_n\|^2_{\mathbf{\Sigma}_n}}_{\textbf{Part II}}, \tag{20}$$

where (a) follows from the Cauchy-Schwarz inequality and (b) follows from the definition of $\overline{\mathbf{\Sigma}}_n$.

Part I captures the efficiency of the data collection process and depends on the optimal design. The quantity represents the maximum possible sum of errors across all items in any list. These errors represent the uncertainty in our estimates of the average reward for each item in any list under the empirical covariance matrix $\mathbf{\Sigma}_n$. By Lemma 2, it is

$$\max_{i \in [L]} \sum_{\mathbf{a} \in \mathbf{A}_i} \|\mathbf{a}\|^2_{\overline{\mathbf{\Sigma}}_n^{-1}} = \mathbf{Tr}(\mathbf{A}_i^\top \mathbf{V}_{\pi_*}^{-1} \mathbf{A}_i) = \frac{d}{n}.$$

Part II measures the quality of the MLE $\widehat{\boldsymbol{\theta}}_n$, and depends on the squared distance of $\widehat{\boldsymbol{\theta}}_n$ from the true parameter $\boldsymbol{\theta}_*$, under the empirical covariance matrix $\mathbf{\Sigma}_n$. For Part II, we use Lemma 7 and get

$$\left\|\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*\right\|^2_{\mathbf{\Sigma}_n} \leq 4\left(\frac{2d\log(6) + \log(1/\delta)}{n}\right)$$

The main claim follows from combining the upper bounds on Parts I and II in (20). $\square$

The supporting lemma is proved below.

Lemma 7. *Under the absolute feedback model, for any $\lambda > 0$, with probability at least $1 - \delta$,*

$$\left\|\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*\right\|^2_{\mathbf{\Sigma}_n} \leq 4\left(\frac{2d\log 6 + \log(1/\delta)}{n}\right).$$

Here $\mathbf{\Sigma}_n = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} \mathbf{x}_{i,k}\mathbf{x}_{i,k}^\top$.

Proof. First we define some additional notation. Recall that each $\mathbf{x}_{i,k}$ is column vector in $\mathbb{R}^d$. We define the feature vector associated with I_t as $\mathbf{X}_t \in \mathbb{R}^{K \times d}$ as $[\mathbf{x}_{I_t,1}, \mathbf{x}_{I_t,2}, \ldots, \mathbf{x}_{I_t,K}]^\top$ and define the feature matrix after n observations $\mathbf{X} \in \mathbb{R}^{Kn \times d}$ as $[\mathbf{X}_1^\top, \mathbf{X}_2^\top, \ldots, \mathbf{X}_n^\top]^\top$. Similarly, we define the observation vector $\mathbf{Y}_t \in \mathbb{R}^K$ at round t as $[y_{t,1}, y_{t,2}, \ldots, y_{t,K}]^\top$ and define the observation vector after n observations $\mathbf{Y} \in \mathbb{R}^{Kn}$ as $[\mathbf{Y}_1^\top, \mathbf{Y}_2^\top, \ldots, \mathbf{Y}_t^\top]^\top$.

Under the sub-Gaussian noise we can show that our MLE is the same as the Ordinary Least-Squares (OLS) estimate such that $\widehat{\boldsymbol{\theta}}_n = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y}$. Then we can show that

$$\begin{aligned}
&\widehat{\boldsymbol{\theta}}_n = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{Y} = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top(\mathbf{X}\boldsymbol{\theta}_* + \eta_n) = \boldsymbol{\theta}_* + (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\eta_n \\
\implies &\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_* = (\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\eta_n.
\end{aligned}$$

Since, the noise is independent sub-Gaussian noise, it follows then for any $\mathbf{a} \in \mathbb{R}^d$

$$\begin{aligned}
\mathbf{a}^\top(\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*) &\sim \mathcal{SG}(0, \mathbf{a}^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\eta_n\eta_n^\top\mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{a}) \\
&\sim \mathcal{SG}(0, \mathbf{a}^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{a}) \\
&\sim \mathcal{SG}(0, \|\mathbf{a}\|^2_{(\mathbf{X}^\top\mathbf{X})^{-1}})
\end{aligned}$$

Therefore we have that using sub-Gaussian concentration inequality that

$$\mathbb{P}\left(\mathbf{a}^\top(\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*) \geq \sqrt{2(\|\mathbf{a}\|^2_{(\mathbf{X}^\top\mathbf{X})^{-1}})\log(1/\delta)}\right) \leq \delta. \tag{21}$$

Also, note that $(\mathbf{X}^\top\mathbf{X}) = n\boldsymbol{\Sigma}_n$. Follows Chapter 20 of Lattimore and Szepesvári [51] We now use a covering argument. Let there exists a set $\mathcal{C}_\epsilon \subset \mathbb{R}^d$ with $|\mathcal{C}_\epsilon| \leq (3/\varepsilon)^d$ such that for all $\mathbf{x} \in S^{d-1}$ there exists a $\mathbf{y} \in \mathcal{C}_\varepsilon$ with $\|\mathbf{x} - \mathbf{y}\|_2 \leq \epsilon$. Now define event

$$\xi = \left\{\text{exists } x \in \mathcal{C}_\varepsilon : \langle \boldsymbol{\Sigma}_n^{1/2}\mathbf{x}, \widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*\rangle \geq \sqrt{\frac{2}{n}\log\left(\frac{|\mathcal{C}_\epsilon|}{\delta}\right)}\right\}$$

We now want to show that $\mathbb{P}(\xi) \leq \delta$. We can show this as follows:

$$\begin{aligned}
\|\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*\|_{\boldsymbol{\Sigma}_n} &= \frac{1}{\sqrt{n}} \max_{\mathbf{x}\in S^{d-1}} \langle \boldsymbol{\Sigma}_n^{1/2}\mathbf{x}, \widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*\rangle \\
&= \frac{1}{\sqrt{n}} \max_{\mathbf{x}\in S^{d-1}} \min_{\mathbf{y}\in\mathcal{C}_\epsilon} \left[\langle \boldsymbol{\Sigma}_n^{1/2}\mathbf{x} - \mathbf{y}, \widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*\rangle + \langle \boldsymbol{\Sigma}_n^{1/2}\mathbf{y}, \widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*\rangle\right] \\
&< \frac{1}{\sqrt{n}} \max_{\mathbf{x}\in S^{d-1}} \min_{\mathbf{y}\in\mathcal{C}_\epsilon} \left[\|\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*\|_{\boldsymbol{\Sigma}_n}\|\mathbf{x} - \mathbf{y}\|_2^2 + \sqrt{2\log\frac{|\mathcal{C}_\epsilon|}{\delta}}\right] \\
&\leq \frac{\epsilon}{\sqrt{n}}\|\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*\|_{\boldsymbol{\Sigma}_n} + \frac{1}{\sqrt{n}}\sqrt{2\log\frac{|\mathcal{C}_\epsilon|}{\delta}}
\end{aligned}$$

Rearranging the above yields

$$\|\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*\|_{\boldsymbol{\Sigma}_n} \leq \frac{1}{\sqrt{n}} \cdot \frac{1}{1-\epsilon}\sqrt{2\log\frac{|\mathcal{C}_\epsilon|}{\delta}}$$

Setting $\epsilon = \frac{1}{2}$ we get that

$$\mathbb{P}\left(\|\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*\|_{\boldsymbol{\Sigma}_n} \geq 2\sqrt{\frac{2d}{n}\log(6) + \frac{1}{n}\log\frac{1}{\delta}}\right) \leq \delta.$$

The claim of the lemma follows. □

C.3 Proof of Theorem 4

From the definition of ranking loss we have

$$\mathbb{E}\left[\mathrm{R}_n\right] = \sum_{i=1}^{L}\sum_{j=1}^{K}\sum_{k=j+1}^{K} \mathbb{E}\left[\mathbb{I}\{\widehat{\sigma}_{n,i}(j) > \widehat{\sigma}_{n,i}(k)\}\right] = \sum_{i=1}^{L}\sum_{j=1}^{K}\sum_{k=j+1}^{K} \mathbb{P}\left(\mathbf{x}_{i,j}^\top\widehat{\boldsymbol{\theta}}_n < \mathbf{x}_{i,k}^\top\widehat{\boldsymbol{\theta}}_n\right).$$

Hence, our first step is to bound the prediction error where item k is predicted above item j under absolute feedback $\mathbb{P}\left(\mathbf{x}_{i,j}^\top \widehat{\boldsymbol{\theta}}_n < \mathbf{x}_{i,k}^\top \widehat{\boldsymbol{\theta}}_n\right)$ for all list $i \in [L]$ and $(j,k) \in \Pi_2(K)$. Our proof closely follows the proof technique of Lemma 2 in Yang and Tan [101]. Recall $\Delta_{i,j,k} = (\mathbf{x}_{i,j} - \mathbf{x}_{i,k})^\top \boldsymbol{\theta}_*$. At the end of round n, we bound the prediction error as follows

$$
\begin{aligned}
\mathbb{P}\left(\mathbf{x}_{i,j}^\top \widehat{\boldsymbol{\theta}}_n < \mathbf{x}_{i,k}^\top \widehat{\boldsymbol{\theta}}_n\right) &= \mathbb{P}\left(\mathbf{x}_{i,j}^\top \widehat{\boldsymbol{\theta}}_n - \mathbf{x}_{i,k}^\top \widehat{\boldsymbol{\theta}}_n - \Delta_{i,j,k} < -\Delta_{i,j,k}\right) \\
&= \mathbb{P}\left((\mathbf{x}_{i,j} - \mathbf{x}_{i,k})^\top \widehat{\boldsymbol{\theta}}_n - (\mathbf{x}_{i,j} - \mathbf{x}_{i,k})^\top \boldsymbol{\theta}_* < -\Delta_{i,j,k}\right) \\
&= \mathbb{P}\left((\mathbf{x}_{i,j} - \mathbf{x}_{i,k})^\top (\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*) < -\Delta_{i,j,k}\right) \\
&\leq \mathbb{P}\left(\mathbf{x}_{i,j}^\top (\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*) < -\Delta_{i,j,k}\right) + \mathbb{P}\left(\mathbf{x}_{i,k}^\top (\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*) > \Delta_{i,j,k}\right) \\
&\overset{(a)}{\leq} \exp\left(-\frac{\Delta_{i,j,k}^2}{2\|\mathbf{x}_{i,j}\|^2_{\overline{\boldsymbol{\Sigma}}_n^{-1}}}\right) + \exp\left(-\frac{\Delta_{i,j,k}^2}{2\|\mathbf{x}_{i,k}\|^2_{\overline{\boldsymbol{\Sigma}}_n^{-1}}}\right) \\
&\overset{(b)}{\leq} \exp\left(-\frac{n\Delta_{i,j,k}^2}{2d}\right) + \exp\left(-\frac{n\Delta_{i,j,k}^2}{2d}\right) \\
&= 2\exp\left(-\frac{n\Delta_{i,j,k}^2}{2d}\right),
\end{aligned}
$$

where, (a) follows from Lemma 8, and (b) follows from Lemma 2. Finally, the total probability of error for the fixed budget setting under absolute feedback is given by

$$
\sum_{i=1}^{L}\sum_{j=1}^{K}\sum_{k=j+1}^{K} \mathbb{P}\left(\mathbf{x}_{i,j}^\top \widehat{\boldsymbol{\theta}}_n < \mathbf{x}_{i,k}^\top \widehat{\boldsymbol{\theta}}_n\right) \leq \sum_{i=1}^{L}\sum_{j=1}^{K}\sum_{k=j+1}^{K} 2\exp\left(-\frac{n\Delta_{i,j,k}^2}{2d}\right).
$$

The claim of the proposition follows. □

The supporting lemma is proved below.

Lemma 8. *For an arbitrary constant Δ and $\mathbf{x} \in \mathbb{R}^d$ we can show that*

$$
\mathbb{P}\left(\mathbf{x}^\top \left(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*\right) > \Delta\right) \leq \exp\left(-\frac{\Delta^2}{2\|\mathbf{x}\|^2_{\overline{\boldsymbol{\Sigma}}_n^{-1}}}\right)
$$

where, $\overline{\boldsymbol{\Sigma}}_n = \sum_{i=1}^n \sum_{k=1}^K \mathbf{x}_{i,k}\mathbf{x}_{i,k}^\top$.

Proof. The proof of the lemma is from Section 2.2 in Jamieson and Jain [38]. Under the sub-Gaussian noise assumption, we can show that for any vector $\mathbf{x} \in \mathbb{R}^d$ the following holds

$$
\mathbf{x}^\top \left(\widehat{\boldsymbol{\theta}} - \boldsymbol{\theta}_*\right) = \underbrace{\mathbf{x}^\top \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top}_{\mathbf{w}} \eta = \mathbf{w}^\top \eta.
$$

Then for an arbitrary constant Δ and $\mathbf{x} \in \mathbb{R}^d$, we can show that

$$
\begin{aligned}
\mathbb{P}\left(\mathbf{x}^\top\left(\widehat{\boldsymbol{\theta}}-\boldsymbol{\theta}_*\right)>\Delta\right) &= \mathbb{P}\left(\mathbf{w}^\top\eta>\Delta\right) \\
&\overset{(a)}{\leq} \exp(-\lambda\Delta)\mathbb{E}\left[\exp\left(\lambda\mathbf{w}^\top\eta\right)\right], \quad \text{let } \lambda>0 \\
&= \exp(-\lambda\Delta)\mathbb{E}\left[\exp\left(\lambda\sum_{s=1}^{t}\mathbf{w}_s\eta_s\right)\right] \\
&\overset{(b)}{=} \exp(-\lambda\Delta)\prod_{s=1}^{t}\mathbb{E}\left[\exp\left(\lambda\mathbf{w}_s\eta_s\right)\right] \\
&\overset{(c)}{\leq} \exp(-\lambda\Delta)\prod_{s=1}^{t}\exp\left(\lambda^2\mathbf{w}_s^2/2\right) \\
&= \exp(-\lambda\Delta)\exp\left(\frac{\lambda^2}{2}\|\mathbf{w}\|_2^2\right) \\
&\overset{(d)}{\leq} \exp\left(-\frac{\Delta^2}{2\|\mathbf{w}\|_2^2}\right) \\
&\overset{(e)}{=} \exp\left(-\frac{\Delta^2}{2\mathbf{x}^\top\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{x}}\right) = \exp\left(-\frac{\Delta^2}{2\|\mathbf{x}\|^2_{\mathbf{\Sigma}_n^{-1}}}\right)
\end{aligned}
$$

where, (a) follows from Chernoff Bound, (b) follows from independence of $\mathbf{w}_s\eta_s$, (c) follows sub-Gaussian assumption, (d) follows by setting $\lambda = \frac{\Delta}{\|\mathbf{w}\|_2^2}$, and (e) follows from the equality

$$
\|\mathbf{w}\|_2^2 = \mathbf{x}^\top\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top\mathbf{X}\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{x} = \mathbf{x}^\top\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{x}.
$$

The claim of the lemma follows. □

D Learning with Ranking Feedback

D.1 Proof of Theorem 5

Since $\overline{\mathbf{\Sigma}}_n$ is invertible, following similar steps to Theorem 3 yields

$$
\max_{i\in[L]}\mathbf{Tr}\left(\mathbf{A}_i^\top\left(\boldsymbol{\theta}_*-\widehat{\boldsymbol{\theta}}_n\right)\left(\boldsymbol{\theta}_*-\widehat{\boldsymbol{\theta}}_n\right)^\top\mathbf{A}_i\right) \leq \underbrace{\max_{i\in[L]}\sum_{\mathbf{a}\in\mathbf{A}_i}\|\mathbf{a}\|^2_{\overline{\mathbf{\Sigma}}_n^{-1}}}_{\textbf{Part I}}\underbrace{\frac{K(K-1)n}{2}\|\boldsymbol{\theta}_*-\widehat{\boldsymbol{\theta}}_n\|^2_{\mathbf{\Sigma}_n}}_{\textbf{Part II}}. \quad (22)
$$

Part I captures the efficiency of the optimal design and Part II captures the uncertainty in the MLE $\widehat{\boldsymbol{\theta}}_n$.

Now we bound the individual quantities in Parts I and II. First, we use Lemma 2 to bound Part I. Then we bound Part II using Lemma 9. We use the proof technique of Theorem 4.1 in Zhu et al. [111] to prove Lemma 9. Finally, the proof follows by combining Lemma 2 and Lemma 9. At the end, we get

$$
\begin{aligned}
\max_{i\in[L]}\sum_{\mathbf{a}\in\mathbf{A}_i}\left(\mathbf{a}^\top\left(\boldsymbol{\theta}_*-\widehat{\boldsymbol{\theta}}_n\right)\right)^2 &\leq \frac{K(K-1)n}{2}\underbrace{\max_{i\in[L]}\sum_{\mathbf{a}\in\mathbf{A}_i}\|\mathbf{a}\|^2_{\overline{\mathbf{\Sigma}}_n^{-1}}}_{\textbf{Part I}}\underbrace{\|\boldsymbol{\theta}_*-\widehat{\boldsymbol{\theta}}_n\|^2_{\mathbf{\Sigma}_n}}_{\textbf{Part II}} \\
&\leq \left(\frac{K(K-1)n}{2}\right)\frac{d}{n}C\left(\frac{K^4(d+\log(1/\delta))}{n}\right) = \widetilde{O}\left(\frac{K^6d^2\log(1/\delta)}{n}\right),
\end{aligned}
$$

for some constant $C>0$. This concludes the proof. □

The supporting lemma is proved below.

Lemma 9. *Fix $\delta\in(0,1)$. Under the ranking feedback model, for any $\lambda>0$ and a constant $C>0$, with probability at least $1-\delta$,*

$$
\left\|\widehat{\boldsymbol{\theta}}_n-\boldsymbol{\theta}^*\right\|^2_{\mathbf{\Sigma}_n} \leq C\cdot\left(\frac{K^4(d+\log(1/\delta))}{n}\right).
$$

Proof. **Step 1 (Strong Convexity of loss):** We first prove the strong convexity of the loss $\ell_{\mathcal{D}_n}(\boldsymbol{\theta})$ with respect to the semi-norm $\|\cdot\|_{\boldsymbol{\Sigma}_n}$ at $\boldsymbol{\theta}_*$ meaning that there is some constant $\gamma > 0$ such that

$$\ell_n(\boldsymbol{\theta}_* + \Delta) - \ell_n(\boldsymbol{\theta}_*) - \langle \nabla \ell_n(\boldsymbol{\theta}_*), \Delta \rangle \geq \gamma \|\Delta\|^2_{\boldsymbol{\Sigma}_n}$$

for all pertubations of Δ such that $\boldsymbol{\theta}_* + \Delta \in \boldsymbol{\Theta}$. First, the Hessian of the negative log-likelihood can be written as

$$\nabla^2 \ell_n(\boldsymbol{\theta}) = \frac{1}{n}\sum_{t=1}^{n}\sum_{j=1}^{K}\sum_{k=j}^{K}\sum_{k'=j}^{K} \frac{\exp\left(\boldsymbol{\theta}^\top \mathbf{x}_{I_t,\sigma_t(k)} + \boldsymbol{\theta}^\top \mathbf{x}_{I_t,\sigma_t(k')}\right)}{2\left(\sum_{k'=j}^{K-1}\exp\left(\boldsymbol{\theta}^\top \mathbf{x}_{I_t,\sigma_t(k')}\right)\right)^2} \cdot \mathbf{z}_{I_t,\sigma_t(k),\sigma_t(k')}\mathbf{z}^\top_{I_t,\sigma_t(k),\sigma_t(k')}.$$

Since $\exp(\boldsymbol{\theta}^\top \mathbf{x}) \in [\mathrm{e}^{-1}, \mathrm{e}]$ for any $\mathbf{x}$, we know that the coefficients satisfy

$$\frac{\exp\left(\boldsymbol{\theta}^\top \mathbf{x}_{I_t,\sigma_t(k)} + \boldsymbol{\theta}^\top \mathbf{x}_{I_t,\sigma_t(k')}\right)}{\left(\sum_{k'=j}^{K-1}\exp\left(\boldsymbol{\theta}^\top \mathbf{x}_{I_t,\sigma_t(k')}\right)\right)^2} \geq \frac{\mathrm{e}^{-4}}{2(K-j)^2}. \tag{23}$$

This implies that for any arbitrary vector $\mathbf{v} \in \mathbb{R}^d$ we have that

$$\begin{aligned}
\mathbf{v}^\top \nabla^2 \ell_n(\boldsymbol{\theta}) \mathbf{v} &\geq \frac{1}{n}\mathbf{v}^\top \left(\sum_{t=1}^{n}\sum_{j=1}^{K}\frac{1}{(K-j)^2}\sum_{k=j}^{K}\sum_{k'=j}^{K}\mathbf{z}_{I_t,\sigma_t(k),\sigma_t(k')}\mathbf{z}^\top_{I_t,\sigma_t(k),\sigma_t(k')}\right)\mathbf{v} \\
&= \mathbf{v}^\top \left(\boldsymbol{\Sigma}_n + \sum_{t=1}^{n}\sum_{j=0}^{K}\frac{1}{n(K-j)^2}\sum_{k=j}^{K}\sum_{k'=j}^{K}\mathbf{z}_{I_t,\sigma_t(k),\sigma_t(k')}\mathbf{z}^\top_{I_t,\sigma_t(k),\sigma_t(k')} - \boldsymbol{\Sigma}_n\right)\mathbf{v} \\
&\overset{(a)}{\geq} \mathbf{v}^\top \boldsymbol{\Sigma}_n \mathbf{v} \\
&= \|\mathbf{v}\|^2_{\boldsymbol{\Sigma}_n},
\end{aligned}$$

where (a) follows by noting

$$\sum_{t=1}^{n}\sum_{j=1}^{K}\frac{1}{n(K-j)^2}\sum_{k=j}^{K}\sum_{k'=j}^{K-1}\mathbf{z}_{I_t,\sigma_t(k),\sigma_t(k')}\mathbf{z}^\top_{I_t,\sigma_t(k),\sigma_t(k')} - \boldsymbol{\Sigma}_n \tag{24}$$

$$= \sum_{t=1}^{n}\sum_{j=1}^{K}\frac{1}{n(K-j)^2}\sum_{k=j}^{K}\sum_{k'=j}^{K}\mathbf{z}_{I_t,\sigma_t(k),\sigma_t(k')}\mathbf{z}^\top_{I_t,\sigma_t(k),\sigma_t(k')} - \frac{2}{K(K-1)n}\sum_{t=1}^{n}\sum_{j=1}^{K}\sum_{k=j+1}^{K}\mathbf{z}_{I_t,j,k}\mathbf{z}^\top_{I_t,j,k} \tag{25}$$

$$= \sum_{t=1}^{n}\frac{1}{n}\left(\sum_{j=1}^{K}\frac{1}{(K-j)^2}\sum_{k=j}^{K}\sum_{k'=j}^{K}\mathbf{z}_{I_t,\sigma_t(k),\sigma_t(k')}\mathbf{z}^\top_{I_t,\sigma_t(k),\sigma_t(k')} - \frac{2}{K(K-1)}\sum_{j=1}^{K}\sum_{k=j+1}^{K}\mathbf{z}_{I_t,j,k}\mathbf{z}^\top_{I_t,j,k}\right), \tag{26}$$

is positive semi-definite matrix.

Hence, we have that $\ell_n(\boldsymbol{\theta})$ is strongly convex at $\boldsymbol{\theta}_*$ with respect to the norm $\|\cdot\|_{\boldsymbol{\Sigma}_n}$. Therefore, we have

$$\begin{aligned}
\gamma\|\Delta\|^2_{\boldsymbol{\Sigma}_n} &\leq \ell_n(\boldsymbol{\theta}_* + \Delta) - \ell_n(\boldsymbol{\theta}_*) - \langle \nabla \ell_n(\boldsymbol{\theta}_*), \Delta \rangle \\
&\overset{(a)}{\leq} -\langle \nabla \ell_n(\boldsymbol{\theta}_*), \Delta \rangle \\
&\leq \|\nabla \ell_n(\boldsymbol{\theta}_*)\|_{\boldsymbol{\Sigma}_n^{-1}} \|\Delta\|_{\boldsymbol{\Sigma}_n}
\end{aligned}$$

where, (a) follows as $\ell_n\left(\widehat{\boldsymbol{\theta}}_n\right) \leq \ell_n(\boldsymbol{\theta}_*)$, and the last inequality follows from $|\langle \nabla \ell_n(\boldsymbol{\theta}_*), \Delta \rangle| \leq \|\nabla \ell_n(\boldsymbol{\theta}_*)\|_{\boldsymbol{\Sigma}_n^{-1}} \|\Delta\|_{\boldsymbol{\Sigma}_n}$.

Step 2 (Upper bound the gradient of loss $\ell_n(\boldsymbol{\theta}_*)$): First recall that the gradient of loss is as follows

$$\begin{aligned}
\nabla \ell_n(\boldsymbol{\theta}_*) &= -\frac{1}{n}\sum_{t=1}^{n}\sum_{j=1}^{K}\sum_{k=j}^{K}\frac{\exp\left(\boldsymbol{\theta}_*^\top \mathbf{x}_{I_t,\sigma_t(k)}\right)}{\sum_{k'=j}^{K}\exp\left(\boldsymbol{\theta}_*, \mathbf{x}_{I_t,\sigma_t(k')}\right)}\mathbf{z}_{I_t,\sigma_t(j),\sigma_t(k)} \\
&\overset{(a)}{=} \frac{1}{n}\sum_{t=1}^{n}\sum_{j=1}^{K}\sum_{k=j}^{K} V_{I_t,j,k}\mathbf{z}_{I_t,j,k} = \frac{1}{n}\mathbf{X}^T\mathbf{V},
\end{aligned}$$

where, in (a) we have define

$$V_{I_t,j,k} = \begin{cases} \frac{\exp\left(\boldsymbol{\theta}_*^\top \mathbf{x}_{I_t,k}\right)}{\sum_{k'=\sigma_t^{-1}(j)}^{K-1} \exp\left(\boldsymbol{\theta}_*^\top \mathbf{x}_{I_t,\sigma_t(k')}\right)}, & \text{if } \sigma_t^{-1}(j) < \sigma_t^{-1}(k) \\ -\frac{\exp\left(\boldsymbol{\theta}_*^\top \mathbf{x}_{I_t,j}\right)}{\sum_{k'=\sigma_t^{-1}(k)}^{K-1} \exp\left(\boldsymbol{\theta}_*^\top, \mathbf{x}_{I_t,\sigma_t(k')}\right)}, & \text{otherwise.} \end{cases}$$

In (b) we define matrix $\mathbf{X} \in \mathbb{R}^{(nK(K-1)/2)\times d}$ has the differencing vector $\mathbf{z}_{I_t,jk}$ as its $\left(tK(K-1)/2 + k + \sum_{l=K-j+1}^{K} l\right)^{th}$ row and $\mathbf{V} \in \mathbb{R}^{nK(K-1)/2}$ is the concatenated vector of $\{\{V_{I_t,j,k}\}_{0\leq j<k\leq K-1}\}_{t=1}^{n}$. With this notation, we have

$$\begin{aligned} \|\nabla \ell_n(\boldsymbol{\theta}_*)\|_{\boldsymbol{\Sigma}_n^{-1}}^2 &= \frac{1}{n^2}\mathbf{V}^\top \mathbf{X}\boldsymbol{\Sigma}_n^{-1}\mathbf{X}^\top \mathbf{V} \\ &\overset{(a)}{\leq} \frac{K^2}{n}\|\mathbf{V}\|_2^2 \\ &\overset{(b)}{\leq} CK^4 \cdot (d + \log(1/\delta)), \end{aligned}$$

where (a) follows as $\frac{K^2}{n} I \succeq \frac{1}{n^2}\mathbf{X}\boldsymbol{\Sigma}_n^{-1}\mathbf{X}^\top$ and the (b) follows with probability $1-\delta$ as the vector $\mathbf{V}$ is sub-Gaussian with parameter K (follows from Zhu et al. [111]) and Bernstein's inequality for sub-Gaussian random variables in quadratic form.

Step 3 (Combining everything): Combining everything we have

$$\begin{aligned} \gamma\|\Delta\|_{\boldsymbol{\Sigma}_n}^2 &\leq \|\nabla \ell_n(\boldsymbol{\theta}_*)\|_{\boldsymbol{\Sigma}_n^{-1}} \|\Delta\|_{\boldsymbol{\Sigma}_n} \\ &\leq \sqrt{C \cdot \frac{K^4(d+\log(1/\delta))}{n}}\|\Delta\|_{\boldsymbol{\Sigma}_n}, \end{aligned}$$

for some finite constant $C > 0$. It follows then that

$$\left\|\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*\right\|_{\boldsymbol{\Sigma}_n}^2 \leq C \cdot \left(\frac{K^4(d+\log(1/\delta))}{n}\right).$$

The claim of the lemma follows. □

D.2 Proof of Theorem 6

Following the same steps as that of Theorem 4, at the end of round n, we bound the prediction error for $(j,k) \in \Pi_2(K)$ for list $i \in [L]$ under ranking feedback as follows

$$\begin{aligned} \mathbb{P}\left(\mathbf{x}_{i,j}^\top \widehat{\boldsymbol{\theta}}_n < \mathbf{x}_{i,k}^\top \widehat{\boldsymbol{\theta}}_n\right) &= \mathbb{P}\left(\mathbf{x}_{i,j}^\top \widehat{\boldsymbol{\theta}}_n - \mathbf{x}_{i,k}^\top \widehat{\boldsymbol{\theta}}_n - \Delta_{i,j,k} < -\Delta_{i,j,k}\right) \\ &= \mathbb{P}\left((\mathbf{x}_{i,j} - \mathbf{x}_{i,k})^\top \widehat{\boldsymbol{\theta}}_n - (\mathbf{x}_{i,j} - \mathbf{x}_{i,k})^\top \boldsymbol{\theta}_* < -\Delta_{i,j,k}\right) \\ &= \mathbb{P}\left((\mathbf{x}_{i,j} - \mathbf{x}_{i,k})^\top (\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*) < -\Delta_{i,j,k}\right) \\ &= \mathbb{P}\left(\mathbf{z}_{i,j,k}^\top (\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*) < -\Delta_{i,j,k}\right) \\ &\overset{(a)}{\leq} \exp\left(-\frac{\kappa^2\Delta_{i,j,k}^2}{2\|\mathbf{z}_{i,j,k}\|_{\boldsymbol{\Sigma}_n^{-1}}^2}\right) \\ &\overset{(b)}{\leq} \exp\left(-\frac{n\kappa^2\Delta_{i,j,k}^2}{2d}\right) \end{aligned}$$

where, (a) follows from Lemma 10, and (b) follows from Lemma 2. Finally, the total probability of error for the fixed budget setting under absolute feedback is given by

$$\sum_{i=1}^{L}\sum_{j=1}^{K}\sum_{k=j+1}^{K} \mathbb{P}\left(\mathbf{x}_{i,k}^\top \widehat{\boldsymbol{\theta}}_n > \mathbf{x}_{i,j}^\top \widehat{\boldsymbol{\theta}}_n\right) \leq \sum_{i=1}^{L}\sum_{j=1}^{K}\sum_{k=j+1}^{K} 2\exp\left(-\frac{n\kappa^2\Delta_{i,j,k}^2}{2d}\right).$$

The claim of the proposition follows. □

The supporting lemma is proved below.

Lemma 10. ***(Restatement of Theorem 1 from Li et al. [59])*** *Let $\delta > 0$ be given Furthermore, assume that*

$$\lambda_{\min}\left(\overline{\boldsymbol{\Sigma}}_n\right) \geq \frac{512\gamma^2}{\kappa^4}\left(d^2 + \log\frac{1}{\delta}\right).$$

where, $\kappa = \inf_{\{\mathbf{x}:\|\mathbf{x}\|\leq 1, \boldsymbol{\theta}:\|\boldsymbol{\theta}-\boldsymbol{\theta}_*\|\leq 1\}} \dot{\exp}\left(\mathbf{x}^\top\boldsymbol{\theta}\right) > 0$, *and* $\dot{\exp}(\cdot)$ *denotes the first derivative of the* $\exp(\cdot)$ *function (see Assumption 2). Then, with probability at least $1-3\delta$, the maximum likelihood estimator for a generalized least square model satisfies, for any $\mathbf{x} \in \mathbb{R}^d$, that*

$$\left|\mathbf{x}^\top\left(\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*\right)\right| \leq \frac{3}{\kappa}\sqrt{\log(1/\delta)}\|\mathbf{x}\|_{\overline{\boldsymbol{\Sigma}}_n^{-1}}.$$

Setting, $\delta = \exp\left(-\frac{9\kappa^2\Delta^2}{\|\mathbf{x}\|_{\overline{\boldsymbol{\Sigma}}_n^{-1}}}\right)$ *for some arbitrary constant $\Delta > 0$ we get that*

$$\mathbb{P}\left(\mathbf{x}^\top\left(\widehat{\boldsymbol{\theta}}_n - \boldsymbol{\theta}_*\right) \geq \Delta\right) \leq \exp\left(-\frac{9\kappa^2\Delta^2}{\|\mathbf{x}\|_{\overline{\boldsymbol{\Sigma}}_n^{-1}}}\right).$$

for $\lambda_{\min}\left(\overline{\boldsymbol{\Sigma}}_n\right) \geq \frac{512}{\kappa^4}\left(d^2 + \frac{\|\mathbf{x}\|_{\overline{\boldsymbol{\Sigma}}_n^{-1}}}{9\kappa^2\Delta^2}\right)$.

E Computation Time for Different Sized Lists

In the following table, we give the computation time for solving the optimization in (6). We use the same setting as in experiment 2 in Section 6 and increase the list size.

List (L)	Computation Time (seconds)
100	4.71
200	8.31
300	15.63
400	21
500	26.6
600	35
700	41.25
800	49.72

Table 1: Computation time Table

From the Table 1 we see that using the package cvxpy results in fast computation of the optimal design in (6).

F Dope Algorithms

In this section we present the full pseudo-code of our algorithm Dope both for the absolute and ranking feedback. First in Algorithm 1 we present the Dope for the absolute feedback setting. The algorithmic details are given in Section 4.1. Then in Algorithm 2 we present the Dope for the ranking feedback setting under the PL model. The algorithmic details are given in Section 5.1.

Algorithm 1 Dope under absolute feedback

Input: Feature vectors $\mathbf{x}_{i,k}$ for all items $k \in [K]$ and for all lists $i \in [L]$, budget n
for $i = 1, \ldots, L$ **do**
  $\mathbf{A}_i \leftarrow [\mathbf{x}_{i,1}, \mathbf{x}_{i,2}, \ldots, \mathbf{x}_{i,1K}]$
end for
$\mathbf{V}_\pi \leftarrow \sum_{i=1}^{L} \pi(i) \mathbf{A}_i \mathbf{A}_i^\top$
$\pi_* \leftarrow \max_{\pi \in \triangle^L} \log\det(\mathbf{V}_\pi)$
for $t = 1, \ldots, n$ **do**
  $I_t \sim \pi_*$
  Observe $y_{I_t,k}$ for all $k \in [K]$
end for
$\mathbf{\Sigma}_n \leftarrow \frac{1}{n} \sum_{t=1}^{n} \sum_{k=1}^{K} \mathbf{x}_{I_t,k} \mathbf{x}_{I_t,k}^\top$ (Covariance matrix)
$\widehat{\boldsymbol{\theta}}_n \leftarrow \mathbf{\Sigma}_n^{-1} \sum_{t=1}^{n} \sum_{k=1}^{K} \mathbf{x}_{I_t,k} y_{t,k}$ (MLE under absolute feedback)
for $i = 1, \ldots, L$ **do**
  $R_i \leftarrow [\mathbf{x}_{i,1}^\top \widehat{\boldsymbol{\theta}}_n, \mathbf{x}_{i,2}^\top \widehat{\boldsymbol{\theta}}_n, \ldots, \mathbf{x}_{i,K}^\top \widehat{\boldsymbol{\theta}}_n]$ (estimated mean rewards)
  Sort R_i in descending order
  for $k = 1, \ldots, K$ **do**
    $\sigma_{n,i}(k) \leftarrow$ item in k-th position in R_i
  end for
end for
Output: Permutation $\sigma_{n,i}(k)$ for all $i \in [L]$

Algorithm 2 Dope under ranking feedback

Input: Feature vectors $\mathbf{x}_{i,k}$ for all items $k \in [K]$ and for all lists $i \in [L]$, budget n
for $i = 1, \ldots, L$ **do**
  for $j = 1 \ldots, K$ **do**
    for $k = j+1 \ldots, K$ **do**
      $\mathbf{z}_{i,j,k} \leftarrow \mathbf{x}_{i,j} - \mathbf{x}_{i,k}$
    end for
  end for
  $\mathbf{A}_i \leftarrow [\mathbf{z}_{i,j,k}]_{(j,k) \in \Pi_2(K)}$
end for
$\mathbf{V}_\pi \leftarrow \sum_{i=1}^{L} \pi(i) \mathbf{A}_i \mathbf{A}_i^\top$
$\pi_* \leftarrow \max_{\pi \in \triangle^L} \log\det(\mathbf{V}_\pi)$
for $t = 1, \ldots, n$ **do**
  $I_t \sim \pi_*$
  Observe σ_{I_t}
end for
$\widehat{\boldsymbol{\theta}}_n \leftarrow \arg\min_{\boldsymbol{\theta}} -\frac{1}{n} \sum_{t=1}^{n} \sum_{k=1}^{K} \log\left(\frac{\exp\left(\mathbf{x}_{I_t,\sigma_t(k)}^\top \boldsymbol{\theta}\right)}{\sum_{k'=k}^{K-1} \exp\left(\mathbf{x}_{I_t,\sigma_t(k')}^\top \boldsymbol{\theta}\right)}\right)$ (MLE under ranking feedback)
for $i = 1, \ldots, L$ **do**
  $R_i \leftarrow [\mathbf{x}_{i,1}^\top \widehat{\boldsymbol{\theta}}_n, \mathbf{x}_{i,2}^\top \widehat{\boldsymbol{\theta}}_n, \ldots, \mathbf{x}_{i,K}^\top \widehat{\boldsymbol{\theta}}_n]$ (estimated mean rewards)
  Sort R_i in descending order
  for $k = 1, \ldots, K$ **do**
    $\sigma_{n,i}(k) \leftarrow$ item in k-th position in R_i
  end for
end for
Output: Permutation $\sigma_{n,i}$ for all $i \in [L]$

G Table of Notations

Notations	Definition
K	Total number of items in a list
d	Dimension of the feature
L	Total number of list
$\triangle^L$	Probability simplex over L items
π_*	Optimal design over L lists
$\sigma_t : [K] \to [K]$	permutation provided by the human labeler at round t
$\mathbf{x}_{i,k}$	Feature of the k-th item in the list i
$\boldsymbol{\theta}_*$	Hidden parameter for the feedback model
$\mathbb{P}\left(\sigma_t\right) = \prod_{k=1}^{K} \frac{\exp\left(\mathbf{x}_{I_t,\sigma_t(k)}^\top \boldsymbol{\theta}_*\right)}{\sum_{i=k}^{K} \exp\left(\mathbf{x}_{I_t,\sigma_t(i)}^\top \boldsymbol{\theta}_*\right)}$	Plackett-Luce model
$y_{t,k} = \mathbf{x}_{I_t,k}^\top \boldsymbol{\theta}_* + \eta_{t,k},$	Absolute feedback model
n	Total horizon
$\Pi_2(K) = \{(j,k) : j < k; j,k \in [K]\}$	set of all ordered pairs where the first coordinate is strictly less than the second one.
$\kappa = \inf_{\{\mathbf{x}: \\|\mathbf{x}\\| \leq 1, \boldsymbol{\theta}: \\|\boldsymbol{\theta} - \boldsymbol{\theta}_*\\| \leq 1\}} \exp\left(\mathbf{x}^\top \boldsymbol{\theta}\right) > 0$	Lower bound of gradient
$\mathbf{V}_\pi = \sum_{i=1}^{L} \pi(i) \mathbf{A}_i \mathbf{A}_i^\top$	Design matrix
$\boldsymbol{\Sigma}_n = \frac{1}{n} \sum_{t=1}^{n} \sum_{k=1}^{K} \mathbf{x}_{I_t,k} \mathbf{x}_{I_t,k}^\top$	Covariance matrix for absolute feedback
$\boldsymbol{\Sigma}_n = \frac{2}{K(K-1)n} \sum_{t=1}^{n} \sum_{j=1}^{K} \sum_{k=j+1}^{K} \mathbf{z}_{I_t,j,k} \mathbf{z}_{I_t,j,k}^\top$	Covariance matrix for ranking feedback
$\overline{\boldsymbol{\Sigma}}_n = \dfrac{K(K-1)n}{2} \boldsymbol{\Sigma}_n$	Un-normalized covariance matrix for ranking feedback
$\Delta_{i,j,k} = (\mathbf{x}_{i,j} - \mathbf{x}_{i,k})^\top \boldsymbol{\theta}_*$	Gap between the mean rewards of items j and k such that $(j,k) \in \Pi_2(K)$ in list i

Table 2: Table of Notations