Sample Complexity Bounds for Active Ranking from Multi-wise Comparisons

Wenbo Ren
Dept. Computer Science & Engineering
The Ohio State University
ren.453@osu.edu

Jia Liu
Dept. Electrical & Computer Engineering
The Ohio State University
liu.1736@osu.edu

Ness B. Shroff
Dept. Electrical & Computer Engineering and Computer Science & Engineering
The Ohio State University
shroff.11@osu.edu

Abstract

We study the sample complexity (i.e., the number of comparisons needed) bounds for actively ranking a set of n items from multi-wise comparisons. Here, a multi-wise comparison takes m items as input and returns a (noisy) result about the best item (the winner feedback) or the order of these items (the full-ranking feedback). We consider two basic ranking problems: top-k items selection and full ranking. Unlike previous works that study ranking from multi-wise comparisons, in this paper, we do not require any parametric model or assumption and work on the fundamental setting where each comparison returns the correct result with probability 1 or a certain probability larger than $\frac{1}{2}$. This paper helps understand whether and to what degree utilizing multi-wise comparisons can reduce the sample complexity for the ranking problems compared to ranking from pairwise comparisons. Specifically, under the winner feedback setting, one can reduce the sample complexity for top-k selection up to an m factor and that for full ranking up to a $\log m$ factor. Under the full-ranking feedback setting, one can reduce the sample complexity for top-k selection up to an m factor and that for full ranking up to an $m \log m$ factor. We also conduct numerical simulations to confirm our theoretical results.

1 Introduction

1.1 Background and motivation

Ranking from comparisons is a class of fundamental problems that underpin many areas in machine learning, and has found various applications in problems involving crowd-sourcing, social choices, recommendation, and searching. In such ranking problems, there is a hidden ranking among multiple items to be recovered, where items may refer to candidates, products, movies, advertisements, etc. In this paper, we study ranking from multi-wise comparisons. A multi-wise comparison refers to a query on m items about the most preferred one (the winner feedback) or the full ranking (the full-ranking feedback) of these items. These comparisons may be deterministic or non-deterministic (i.e., noisy or they may return incorrect results). The noise comes from the uncertain nature of humans, the lack of information, or the underlying physics. In this paper, we focus on two goals. One is to find the top-k items (ranking or ordering these items are not necessary), and the other is to find the full ranking.

Our focus is on *active ranking* (e.g., [4–7, 10, 17–19]), where "active" means that after each comparison, the learner can adaptively choose the next items to be compared according to past observations

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

and comparison results. Active ranking can be viewed as active learning for ranking problems and the comparisons refer to the samples. The opposite of active ranking is passive ranking, where the learner first obtains a set of comparison results and then recovers a ranking from there. Active ranking can greatly reduce the sample complexity in many scenarios, e.g., if all comparisons return correct results with probability $\frac{2}{3}$, passively ranking n items needs $\Omega(n^2)$ comparisons [3], while active ranking only needs $O(n \log n)$[1] comparisons [9, 18].

Most existing works have focused on ranking from pairwise comparisons. In contrast, we focus on ranking from *multi-wise* (or m-wise) comparisons. The pairwise comparisons can be viewed as multi-wise comparisons with $m = 2$. One motivation is that in many scenarios, multi-wise comparisons are more common. For instance, in video streaming websites or e-shopping apps, customers or users are normally presented more than two options, and the choices made by users can be viewed as multi-wise comparisons that suggest their preferences over these options. Studying ranking from multi-wise comparisons is useful for these types of applications. Besides, ranking from multi-wise comparisons may also reduce the cost of the learning process. In some applications, conducting the comparisons may be expensive. For instance, to find the candidates that are most preferred by the voters, people may need to do a series of surveys. Each survey contains a query about the preference order of a voter. In this application, the cost of finding the voter and asking this voter to fill the survey could far outweigh the cost of filling the survey itself. This means that the cost of conducting a multi-wise comparison is almost the same as that for a pairwise comparison, and how to reduce the number of samples by multi-wise comparisons becomes more interesting. Thus, it is not only interesting but also significant to study whether and to what degree we can reduce the sample complexity for ranking by using multi-wise comparisons.

We focus on a non-parametric model, where each comparison returns the correct result with a certain probability $q > \frac{1}{2}$ and an arbitrary incorrect result otherwise. When $q = 1$, the comparisons are deterministic and always return correct results, and when $q < 1$, we say the comparisons are non-deterministic or noisy. This differs from parametric models[2] that assume that each item holds a value representing the users' preference on this item. In parametric models, the comparisons may provide more information than the setting in this paper[3], and thus, the conclusion drawn under parametric models cannot be directly applied to this paper.

1.2 Problem formulation

Assume that there are n items, indexed by $1, 2, 3, ...n$, that form the item set $[n]$[4]. We further assume that these items have a *unique unknown true* ranking $r_1 \succ r_2 \succ \cdots \succ r_n$, where for any items i and j, notation $i \succ j$ means that item i ranks higher (or is more preferred) than item j. Assume that we can compare at most m items at a time. The comparisons can be either deterministic or non-deterministic. We also assume that the comparisons are independent across time, items, and sets, which is standard in the literature (e.g., [5–7, 10, 11, 13, 17–22]). Here, we note that the independence is based on the assumption that the hidden parameters and ranking are some fixed values.

When the comparisons are deterministic, the comparisons always return the correct results, i.e., the best item under the winner feedback model or the true ranking of the compared items under the full-ranking feedback model. In the deterministic case, our goal is to find the exact top-k items or the true ranking.

When the comparisons are non-deterministic, we assume that they return the correct results with a certain fixed probability $q \in (\frac{1}{2}, 1)$. In this case, it is infeasible to rank the items with 100% confidence, and thus, our goal is to find the ranking with confidence $1 - \delta$ for some $\delta \in (0, \frac{1}{2})$. We focus on the case where $q = \frac{2}{3}$. We note that this does not lose much generality. When $q > \frac{2}{3}$, the

[1] All log in this paper are natural log unless explicitly noted.

[2] For instance, parametric models can be the Bradley-Terry-Luce (BTL) model [2], the Plackett-Luce (PL) model [15], or the multinomial logit (MNL) model [14].

[3] For instance, given three items i, j, k with i being the best and k being the worst, in the setting of this paper with $q = 0.7$, the multi-wise comparison returns the best item i with probability 0.7, and return j or k with some unknown probabilities, which does not provides information about the orders of j and k. In contrast, in a parametric model, the comparison over $\{i, j, k\}$ would return i, j, or k with probability 0.7, 0.2, and 0.1, respectively, which not only provides information about the best one but also information for ordering j and k.

[4] For any positive integer l, we define $[l] := \{1, 2, 3, ..., l\}$.

algorithms and sample complexity bounds in this paper can be directly applied. When $q < \frac{2}{3}$, we can use repeated comparisons to simulate one comparison with correct probability at least $\frac{2}{3}$. To see this, consider a set over which a comparison returns the correct result with probability $\frac{1}{2} + \Delta > \frac{1}{2}$. By comparing this set for $\lceil \frac{1}{2\Delta^2} \log 3 \rceil$ times, the item or permutation that occurs most often is the correct result with probability at least $\frac{2}{3}$ (by Hoeffding Inequality). Thus, we can use the above method to substitute the comparisons in the algorithms designed for the case where $q = \frac{2}{3}$ while only introducing an additional $\frac{1}{\Delta^2}$ factor on the sample complexity.

This $(\frac{1}{2} + \Delta)$ m-wise comparison model can also be justified in many scenarios that use an iterative subroutine to conduct m-wise comparisons over a set in a smaller time-scale. Here, we give a simple example. We use $p_{i,S}$ to denote the probability that item i wins the comparison performed on set S (assuming $|S| \leq m$) and we use i^* to denote the best item of S. If $p_{i^*,S} \geq p_{i,S} + \Delta$ for any item $i \neq i^*$ and any set S, then by repeatedly comparing S for $\Theta(\frac{1}{\Delta^2} \log |S|)$ times, one can find the best item of S with confidence $\frac{2}{3}$. We can use the above procedure as a subroutine in each iteration of our algorithms, and get the algorithms for this case while only introducing an additional $\log m$ factor to the sample complexities. In more general cases, we can use similar tricks but the additional factors may vary.

1.3 Main results

The main results of this paper are summarized in Table 1. We note that due to space limitation, all proofs in this paper are left to the supplementary material.

Table 1: Main results of this paper. All results are established in this paper unless explicitly cited. For results of non-deterministic feedback, δ is the error probability and we assume that all comparisons return correct results with probability $\frac{2}{3}$.

Problem		Top-k Selection	Full Ranking
Winner Feedback Model	Deterministic Feedback	$\Theta(\frac{n}{m} + k)$	$\Theta(\frac{n \log n}{\log m})$ [17]
	Non-Deterministic Feedback	$O((\frac{n}{m} + k) \log \frac{k \log m}{\delta})$ $\Omega(k + \frac{n}{m} \log \frac{k}{\delta})$	$\Theta(n \log \frac{n}{\delta})$
Full-Ranking Feedback Model	Deterministic Feedback	$\Theta(\frac{n}{m})$	$\Theta(\frac{n \log n}{m \log m})$
	Non-Deterministic Feedback	$O(\frac{n}{m} \log \frac{\min\{n/m,k\} \log m}{\delta})$ $\Omega(\frac{n}{m} \log_m \frac{k}{\delta})$	$O(\frac{n}{m} \log \frac{n}{m\delta})$ $\Omega(\frac{n}{m} \log_m \frac{n}{\delta})$

2 Related works

When $m = 2$ and the comparisons are deterministic, the top-k selection problem becomes the classical pairwise k-selection problem, which requires $\Theta(n)$ comparisons [1], and the full ranking problem becomes the classical pairwise sorting problem, which requires $\Theta(n \log n)$ comparisons. Thus, ranking from multi-wise comparisons can also be viewed as extensions of these foundational problems. Surprisingly, these extensions have not been well understood.

For top-k selection from non-deterministic pairwise comparisons, the authors of [9] showed that $\Theta(n \log \frac{k}{\delta})$ comparisons are necessary and sufficient to reach confidence level $1 - \delta$. For probably approximately correct (PAC)[5] top-k selection with error tolerance $\epsilon > 0$, the authors of [5, 7] proved a $\Theta(\frac{n}{\epsilon^2} \log \frac{1}{\delta})$ bound for $k = 1$, and the authors of [16, 18] proved a $\Theta(\frac{n}{\epsilon^2} \log \frac{k}{\delta})$ bound for $k \leq \frac{n}{2}$.

For full ranking from non-deterministic pairwise comparisons, an early work was [9], whose authors proved that when the comparisons of all pairs have the same noise level, then to get the full ranking with confidence $1 - \delta$, $\Theta(n \log \frac{n}{\delta})$ comparisons are necessary and sufficient. The authors of [17]

[5] The PAC setting means that we want to find the ranking approximately with an error tolerance ϵ and a confidence $1 - \delta$, where the ϵ tolerance means that for any two items i and j, item i is viewed to rank higher than item j if item i wins the comparisons over item j with probability $\frac{1}{2} - \epsilon$ or higher.

extend the above results to the case where the comparisons on different pairs have different noise levels, and also obtained the $\Theta(n \log \frac{n}{\delta})$ bounds. The authors of [5–7] showed the $\Theta(\frac{n}{\epsilon^2} \log \frac{n}{\delta})$ order-wise tight upper and lower bounds for PAC full ranking with error tolerance $\epsilon > 0$.

Thus, we can see that for $m = 2$ (i.e., ranking from pairwise comparisons), people have already established optimal sample complexity bounds for top-k selection and full ranking. However, the corresponding multi-wise ranking problems have been relatively under-explored. Most of these works have focused on parametric models such as the PL model and the MNL model. Under the MNL model, the work in [4] showed that under certain cases, one can get an m-reduction from m-wise comparisons over ranking from pairwise comparisons. However, for most cases, using m-wise comparisons does not reduce the sample complexity (ignoring constant factors). For multi-wise full ranking under the MNL model, the works in [16, 17] showed that the lower bound is the same as that using pairwise comparisons, i.e., there is no reduction by using m-wise comparisons. Under the list-wise PL model with certain types of feedback, we may achieve up to m-reduction in the sample complexity according to [19].

In contrast to the above works, we do not assume any parametric model, and instead use a non-parametric model where each comparison returns a correct result with a certain fixed probability q. As noted in Section 1.1, the results drawn from the parametric models cannot be directly applied to this non-parametric model. Multi-wise ranking under this non-parametric setting is even less under-explored in the literature. The most related work to this paper is [17], where the authors showed a $\Theta(\frac{n \log n}{\log m})$ bound for full ranking from the winner feedback, which solves one of the eight cases (top-k selection or full ranking, winner feedback or full-ranking feedback, deterministic feedback or non-deterministic feedback) that is studied in this paper and shown in Table 1. We will study the rest seven cases in this paper.

3 Top-k selection from winner feedback

3.1 Deterministic feedback

Lower bound. We first state an $\Omega(\frac{n}{m} + k)$ lower bound in Theorem 1. When $m = 2$, this lower bound reduces to $\Omega(n)$, the same as that for the pairwise k-selection problem.

Theorem 1. *To find the top-k items from n items by m-wise deterministic winner feedback, any algorithm needs to conduct at least $\Omega(\frac{n}{m} + k)$ comparisons.*

Upper bound When $m = 2$, the problem reduces to the basic pairwise k-selection problem, which requires $\Theta(n)$ comparisons. One algorithm to solve the pairwise k-selection problem is Quick-Select (QS) [12]. With n items, QS randomly chooses a pivot, splits other items into two piles based on whether them are larger or smaller than the pivot, and then recursively calls QS on one of these two piles according to the piles' sizes. The expected number of comparisons required by QS is $O(n)$ (in the worst case it will be $O(n^2)$).

However, if we want to get the $O(\frac{n}{m} + k)$ sample complexity upper bound for multi-wise top-k selection that matches the lower bound stated in Theorem 1, we cannot split the items into two piles because splitting $(n - 1)$ items into two piles requires $\Omega(n)$ comparisons. Instead, we split them into m piles that are formed by $(m - 1)$ randomly chosen pivots. Also, if the algorithm splits all the non-pivot items, then we still have $\Omega(n)$ sample complexity, which is sub-optimal. Our key idea is to stop splitting if we have identified the elements of the first several piles and the number of items in these piles along with the corresponding pivots is no less than k. By analyzing how the items will be split, we will show that we only need to conduct $O(\frac{n}{m} + k)$ comparisons in expectation before terminating splitting. After the splitting, we only need to focus on the pile that contains the k-th item as the piles that rank higher contain only items better than the k-th item and the others contain only items worse then the k-th item. We can show that this pile is of size $O(\frac{n}{m})$ in expectation, and thus, finding the top-k items from it only takes $O(\frac{n}{m})$ comparisons by QS. Therefore, we find the top-k items by using $O(\frac{n}{m} + k)$ comparisons in expectation.

We name the above method as Multi-wise Quick-Select (MQSelect) and describe it in Algorithm 1. The theoretical performance of MQSelect is formally stated in Theorem 2.

Theorem 2. *MQSelect returns the top-k items of S after $O(\frac{n}{m} + k)$ comparisons in expectation.*

Algorithm 1 Multi-wise Quick-Select(S, m, k) (MQSelect).

```
if $|S| \le m$ then
    Compare $S$ for $k$ times; After each comparison remove the winner from $S$ and add it to $Ans$;
    return $Ans$;
end if
Randomly choose $m-1$ items and form a pivot set $V$;
$R_1 \leftarrow S - V$; $R_2, R_3, ..., R_m \leftarrow \emptyset$; $D_0 \leftarrow \emptyset$; $A_i \leftarrow \emptyset$ for $i \in [m-1]$;
for $t = 1, 2, 3, ..., m-1$ do
    Compare $V$ once and denote the winner as $v_t$;
    $V \leftarrow V - \{v_t\}$; $E \leftarrow \emptyset$;
    while $R_t \neq \emptyset$ do
        Choose items from $R_t$ and add them to $E$ until $|E|$ reaches $\min\{m-1, |R_t|\}$;
        Compare $E \cup \{v_t\}$ and denote the winner as $w$;
        if $w = v_t$ then
            $R_{t+1} \leftarrow R_{t+1} + E$; $R_t \leftarrow R_t - E$; $E \leftarrow \emptyset$;
        else
            $A_t \leftarrow A_t + \{w\}$; $R_t \leftarrow R_t - \{w\}$; $E \leftarrow E - \{w\}$;
        end if
    end while
    $D_t \leftarrow D_{t-1} \cup A_t \cup \{v_t\}$;
    if $|D_t| = k$ then
        return $D_t$;
    else if $|D_t| = k+1$ then
        return $D_t - \{v_t\}$;
    else if $|D_t| > k+1$ then
        return $D_{t-1} \cup$ Quick-Select$(A_t, k - |D_{t-1}|)$;
    end if
end for
return $D_{m-1} \cup$ Quick-Select$(R_m, k - |D_{m-1}|)$;
```

According to the lower bound stated in Theorem 1, MQSelect is optimal up to a constant factor. When $m = 2$, the $\Theta(\frac{n}{m} + k)$ bound reduces to $\Theta(n)$, the same as that for pairwise k-selection.

3.2 Non-deterministic feedback

Lower bound. The lower bound for m-wise top-k selection from non-deterministic winner feedback is stated in Proposition 3. Later Theorem 5 will show that this bound is optimal up to a $\log \frac{k \log m}{\delta}$ factor, and when $k \le \frac{n}{m}$, this bound is optimal up to a $\log \log m$ factor.

Proposition 3. *There is an n-sized instance such that to find the top-k items with confidence $1 - \delta$ by using m-wise non-deterministic winner feedback, any algorithm needs at least $\Omega(k + \frac{n}{m} \log \frac{k}{\delta})$ comparisons in expectation.*

Upper bound. We first introduce a simple subroutine Basic Compare (BC) in Algorithm 2. Given δ and input set S, BC compares S for $N_0 = \lceil 18 \log \frac{1}{\delta} \rceil$ times and returns the most often result, which is correct with probability at least $1 - \delta$. This can be shown by using Hoeffding inequality

Algorithm 2 Basic Compare(S, δ) (BC).

```
Compare $S$ for $N_0 = \lceil 18 \log \frac{1}{\delta} \rceil$ times;
return the result that is returned for the most number of times;
```

$$\mathbb{P}\{X \le N_0/2\} \le \exp\{-2N_0(2/3 - 1/2)^2\} \le \delta, \tag{1}$$

where X is the number of times that the correct result is returned.

We now present a relatively simple algorithm called Basic k-Selection (BKS), which will be used later for developing another algorithm. The idea of BKS is that we replace the m-wise comparisons in

MQSelect by the calls of BC with a certain confidence $1-\delta_1$. Intuitively, we can set $\delta_1 = \frac{\delta}{n^2}$ to make BKS return the correct result with confidence $1-\delta$ as MQSelect conducts at most n^2 comparisons. In this case, we will need $O((k+\frac{n}{m})\log\frac{n}{\delta})$ comparison. However, by showing that with probability at least $1-\delta_0$, MQSelect conducts at most $N = O((k+\frac{n}{m})\log\frac{nk\log m}{m\delta_0})$ comparisons, we can set $\delta_1 = \frac{\delta_0}{N}$ and $\delta_0 = \frac{\delta}{3}$ and get a better upper bound $O((\frac{n}{m}+k)\log\frac{nk\log m}{m\delta})$. In fact, if we use BKS with $\delta_1 = \frac{\delta}{n^2}$, then the algorithm we construct in the later subsection will have a worse upper bound. We describe BKS in Algorithm 3. Its theoretical performance is formally stated in Lemma 4.

Algorithm 3 Basic k-Select(S, m, k, δ) (BKS).

```
$\delta_0 \leftarrow \frac{\delta}{3}$; $T_1 \leftarrow 1 + \frac{n}{m-2}\log\frac{2(m-1)(m-2)}{\delta_0}$; $\delta_1 \leftarrow \frac{\delta_0}{6k+5T_1}$ if $m > 2$; $\delta_1 \leftarrow \frac{\delta_0}{|S|^2}$ if $m = 2$;
Run MQSelect on $M$, but using calls of BC with confidence $1-\delta_1$ to replace multi-wise comparisons (except those in the call of QS);
For the call of QS, we replace the pairwise comparisons with calls of BC with confidence $1-\frac{\delta_0}{|A_t|^2}$;
```

Lemma 4. *BKS terminates after $O((\frac{n}{m}+k)\log\frac{nk\log m}{m\delta})$ comparisons in expectation, and with probability at least $1-\delta$, returns the top-k items of S.*

With BKS, we develop an enhanced algorithm for multi-wise top-k selection that can remove the $\log n$ factor in the sample complexity of BKS. At each round t, we split the remaining items into subsets with sizes at most mk. For each set, we use BKS with confidence $1-\frac{\delta_t}{k}$ to find the top-k items and call them *winners*. We can show that with probability at least $1-\delta_t$, the winners of all subsets together contain the top-k items of $[n]$. We keep the winners and remove other items. Repeat the above step on remaining items for multiple rounds until only k winners remain, and these k winners are the top-k items of $[n]$ with probability at least $1-\sum_{t=1}^{\infty}\delta_t$. By setting proper values of δ_t, we can find the top-k items with confidence $1-\delta$ by using $O((\frac{n}{m}+k)\log\frac{k\log m}{\delta})$ comparisons. We name this algorithm Multi-wise Tournament k-Select (MTKS) and describe it in Algorithm 4. Its theoretical performance is formally stated in Theorem 5.

Algorithm 4 Multi-wise Tournament k-Select (S, m, k, δ) (MTKS)

```
if $|S| \le mk$ then
    return BKS$(S, m, k, \delta)$;
end if
Set $t \leftarrow 0$ and $R_1 \leftarrow S$;
repeat
    $t \leftarrow t+1$; $\delta_t \leftarrow \frac{6\delta}{\pi^2 t^2}$;
    Distribute $R_t$ to $\lceil\frac{|R_t|}{mk}\rceil$ disjoint sets $A_1, A_2, ..., A_d$, each with size at most $mk$;
    For $i \in [d]$, let $B_i \leftarrow$BKS$(A_i, m, k, \frac{\delta_t}{k})$;
    $R_{t+1} \leftarrow \bigcup_{i\in[d]} B_i$;
until $|R_{t+1}| \le k$
return $R_{t+1}$;
```

Theorem 5. *MTKS terminates after $O((\frac{n}{m}+k)\log\frac{k\log m}{\delta})$ comparisons in expectation, and with probability at least $1-\delta$, returns the top-k items of S.*

We can see that, by using multi-wise comparisons, MTKS can achieve up to an m-reduction in the sample complexity. According to Proposition 3, when $k \le \frac{n}{m}$, our upper bound is optimal up to a $\log\log m$ factor, which is almost constant for most of the practical cases.

4 Full ranking from winner feedback

4.1 Deterministic feedback

This setting has been studied in [17], where the authors have shown $\Theta(n \log_m n)$ upper and lower bounds for the sample complexity. We restate their results in Proposition 6. By using m-wise comparisons, we can reduce the sample complexity for finding the full ranking by a $\log m$ factor.

Proposition 6 ([17])**.** *To exactly rank n items by using m-wise deterministic comparisons under the winner feedback model, $\Theta(n \log_m n)$ comparisons are necessary and sufficient.*

4.2 Non-deterministic feedback

Unlike full ranking from deterministic comparisons, when the comparisons are non-deterministic, this logarithmic reduction may not exist any more. This result is stated in Theorem 7.

Theorem 7. *There is an n-sized instance such that to get the full ranking with confidence $1 - \delta$ from m-wise winner feedback, any algorithm needs at least $\Omega(n \log \frac{n}{\delta})$ comparisons.*

For $m = 2$, i.e., when using pairwise comparisons, the algorithms in [18] can already find the full ranking of n items with confidence $1 - \delta$ by $O(n \log \frac{n}{\delta})$ comparisons, matching the lower bound in Theorem 7. Thus, we do not propose an algorithm for $m > 2$ and one can directly use the pairwise algorithms for $m > 2$ and obtain optimal sample complexity (ignoring constant factors).

5 Top-k selection from full-ranking feedback

5.1 Deterministic feedback

For the lower bound, since each comparison involves at most m items and all items need to be involved in at least one comparison to get the top-k items, we immediately have the $\Omega(\frac{n}{m})$ lower bound. For the upper bound, we develop a Quick-Select-like algorithm MQSelect-FRF (Multi-wise Quick-Select from Full-Ranking Feedback) under the full-ranking feedback model similar to MQSelect. In the splitting, MQSelect-FRF takes less comparisons since the full-ranking feedback provides more information than the winner feedback, which removes the $O(k)$ term in the upper bound that exists in the sample complexity of MQSelect. Due to space limitation, we only state the bounds in Theorem 8 and the proofs and algorithms are relegated to the supplementary material.

Theorem 8. *To get the top-k items of n items from deterministic m-wise full-ranking feedback, $\Theta(\frac{n}{m})$ comparisons are necessary and sufficient.*

5.2 Non-deterministic feedback

Lower bound. In this part, we provide a lower bound in Proposition 9, which is optimal up to a $\log m \cdot \log \log m$ factor. This lower bound is proved by reducing the pairwise ranking problem to the list-wise ranking problem. In fact, we are not aware whether there is a stronger lower bound, and this problem requires future investigation.

Proposition 9. *There is an n-sized instance such that to find the top-k items by m-wise non-deterministic full-ranking feedback, any algorithm needs $\Omega(\frac{n}{m} \log_m \frac{k}{\delta})$ comparisons.*

Upper bound. We develop the full-ranking algorithm following the similar steps as in MTKS. First, we develop an algorithm named BKS-FRF (BKS from Full-Ranking Feedback) similar to BKS, which replaces the comparisons of the deterministic-comparison algorithm by a call of BC with a certain confidence and has sample complexity $O(\frac{n}{m} \log \frac{n \log m}{m\delta})$. When $k < \frac{n}{m}$, we further use similar steps as in MTKS and develop an algorithm named MTKS-FRF (MTKS from Full-Ranking Feedback) that further reduces the sample complexity to $O(\frac{n}{m} \log \frac{\min\{n/m,k\} \log m}{\delta})$, which is optimal up to a $\log m \cdot \log \log m$ factor. Due to space limitation, we only state the upper bound in Theorem 10 and the algorithms are presented in the supplementary material.

Theorem 10. *MTKS-FRF terminates after $O(\frac{n}{m} \log \frac{\min\{n/m,k\} \log m}{\delta})$ comparisons in expectation, and with probability at least $1 - \delta$, returns the top-k items of S.*

6 Full ranking from full-Ranking feedback

6.1 Deterministic feedback

Lower bound. We first show the lower bound for this setting in Theorem 11. The proof is based on an information-theoretic approach. Specifically, there are $n!$ permutations, which implies that the information entropy of the true ranking is $\log(n!)$. Each m-wise comparison has at most $m!$ possible results. Thus, each m-wise comparisons provides at most $\log(m!)$ information towards the true ranking. By Fano's Inequality [8], to get the true ranking with probability at least $\frac{3}{4}$, the information about the true ranking one needs to obtain is at least $\Omega(\log(n!))$. Since $\log(n!) = \Theta(n \log n)$ and $\log(m!) = \Theta(m \log m)$, the lower bound is at least $\Omega(\frac{n \log n}{m \log m})$.

Theorem 11. *To get the full ranking of n items by using m-wise comparisons under the full-ranking feedback model, any algorithm needs at least $\Omega(\frac{n \log n}{m \log m})$ comparisons in expectation.*

Algorithm 5 Multi-wise Quick-Sort(S, m) (MQSort)

```
if $|S| \leq m$ then
    Compare $S$, obtain its full ranking and return;
end if
Randomly choose $h := \lfloor \frac{m}{2} \rfloor$ items and form pivot set $V$;
Compare $V$ to get its full ranking $v_1 \succ v_2 \succ \cdots \succ v_h$;
Let $v_0$ represent a dummy item that ranks higher than all other items;
$R \leftarrow S - V$; $A_i \leftarrow \emptyset$ for $i = 0, 1, 2, ..., h$;
while $R \neq \emptyset$ do
    Choose $(m - h)$ items from $R$ and form set $E$;
    Compare set $E \cup V$ and get the ranking;
    for item $i$ in $E$ do
        Add item $i$ to set $A_j$ if $j$ items in $V$ ranks higher than $i$ (i.e., if $v_{j-1} \succ i \succ v_j$);
    end for
end while
for $j = 0, 2, 3, ..., h$ do
    Call MQSort($A_j, m$) to get the full ranking of $A_j$;
    Insert the sorted items of $A_j$ into between $v_{j-1}$ and $v_j$;
end for
return the current ranking of $S$;
```

Upper bound. Since it is well-known that full ranking from pairwise comparisons needs $O(n \log n)$ comparisons, developing a full ranking algorithm under the non-deterministic full-ranking feedback model with sample complexity $O(\frac{n}{m} \log n)$ is trivial by viewing an m-wise comparison as $\lfloor \frac{m}{2} \rfloor$ pairwise comparisons. Instead, we propose an algorithm called Multi-wise Quick-Sort (MQSort) with expected sample complexity $O(\frac{n \log n}{m \log m})$, better than the above trivial bound and matches the lower bound stated in Theorem 11. MQSort can be viewed as an extension of the classical Quick-Sort algorithm, but requires more complicated mathematical analysis to prove its sample complexity. We describe MQSort in Algorithm 5 and formally state its theoretical performance in Theorem 12.

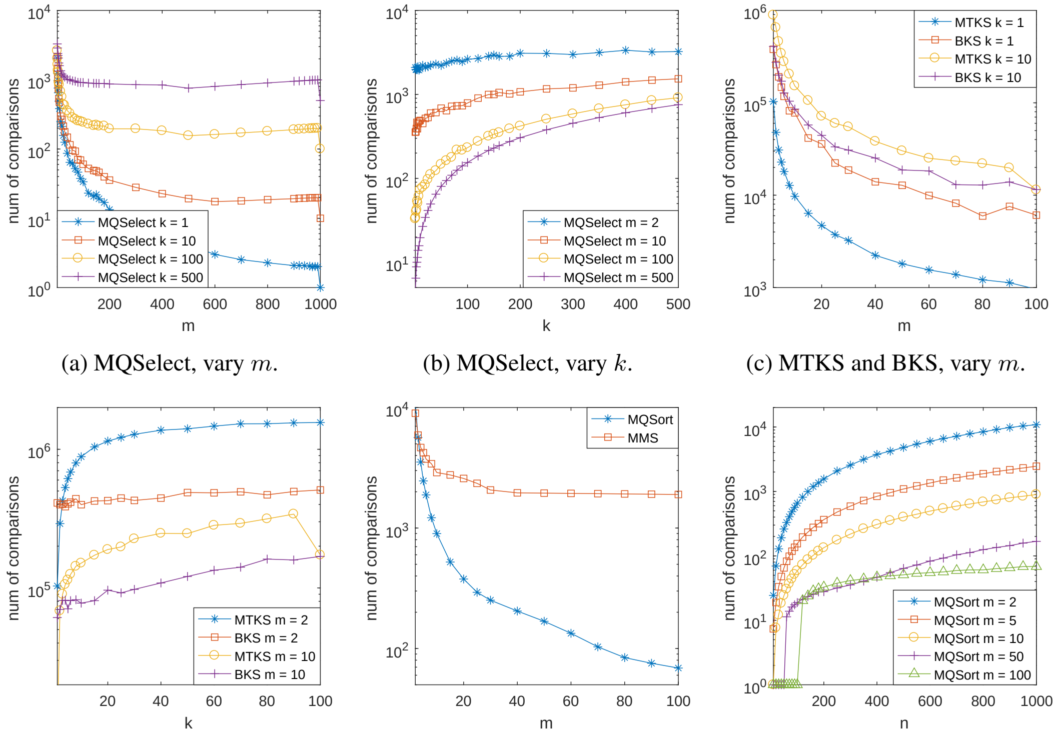
Theorem 12. *MQSort terminates after $O(\frac{n \log n}{m \log m})$ m-wise comparisons in expectation and returns the full ranking of S.*

6.2 Non-deterministic feedback

Lower bound. When the full ranking is obtained, then the top-$\frac{n}{2}$ items can also be obtained for free. Thus, we immediately have the $\Omega(\frac{n}{m} \log_m \frac{n}{\delta})$ full-ranking lower bound in Corollary 13 by invoking Proposition 9. Corollary 13 is optimal up to a $\log m$ factor. Whether a tighter bound exists remains an open problem and requires further investigation.

Corollary 13. *There is an n-sized instance such that to find the full ranking from m-wise full-ranking feedback with confidence $1 - \delta$, any algorithm needs $\Omega(\frac{n}{m} \log_m \frac{n}{\delta})$ comparisons in expectation.*

Next, we modify the algorithm in [9] to achieve an $O(\frac{n}{m} \log \frac{n}{m\delta})$ upper bound, which is stated in Theorem 14. Due to space limitation, we leave the algorithm to the supplementary material.



(a) MQSelect, vary m.

(b) MQSelect, vary k.

(c) MTKS and BKS, vary m.

(d) MTKS and BKS, vary k.

(e) MQSort and MMS, vary m.

(f) MQSort, vary n.

Figure 1: Performance comparisons of all algorithms: In all figures, $n = 1000$ (except (f)), $\delta = 0.01$ (if applicable), and all points are averaged over 100 independent trials with random true rankings.

Theorem 14. *There is an algorithm that finds the full ranking of n items by m-wise full-ranking feedback with confidence $1 - \delta$ and conducts $O(\frac{n}{m} \log \frac{n}{m\delta})$ comparisons in expectation.*

7 Numerical results

In this section, we conduct numerical experiments to verify out theoretical results. The codes can be found in our GitHub repo.[6]

In Figure 1 (a,b), we present the results of MQSelect. In Figure 1 (a), we set $n = 1000$ and $k = \{1, 10, 100, 500\}$, and vary the value of m. In Figure 1 (b), we set $n = 1000$ and $m = \{2, 10, 100, 500\}$, and vary the value of k. First, from the results, we can see that that given the same value of k, the number of comparisons conducted by MQS decreases as m increases, and the decreasing rate is approximately $\frac{1}{m}$ according to Figure 1 (a), consistent with our theory that the sample complexity of MQSelect is $O(\frac{n}{m} + k)$. This indicates that using multi-wise comparisons can significantly reduce the required number of comparisons for top-k ranking, especially for a large value of m. Second, Figure 1 (b) shows that for a given value of m, when k increases, the number of comparisons conducted by MQSelect increases nearly linearly. The larger the m-values, the closer to linear the increasing rates are, which is also consistent with the theory.

In Figure 1 (c,d), we compare the performance of MTKS and BKS. In Figure 1 (c), we set $n = 1000$, $\delta = 0.01$, and $k = \{1, 10\}$, and vary the value of m. In Figure 1 (d), we set $n = 1000$, $\delta = 0.01$, and $m = \{2, 10\}$, and vary the value of k. First, we can see from Figure 1 (c) that when m increases, the number of comparisons conducted by MTKS and BKS both decrease, and the decreasing rate is larger for smaller values of k, which is consistent with the theory that the sample complexities of both algorithms depend on $(\frac{n}{m} + k)$. This also indicates that by using multi-wise comparisons, we can save a significant number of comparisons for top-k selection. Second, from Figure 1 (c), we

[6] https://github.com/WenboRen/Multi-wise-Ranking.git

can see that, when $k = 1$, MTKS uses less comparisons than BKS for almost all values of m, which is consistent with the theory that the sample complexity of MTKS depends on $\log \frac{k \log m}{\delta}$, smaller than the $\log \frac{nk \log m}{m\delta}$ rate of BKS. However, from Figure 1 (c), we can also see that, when k is larger, e.g., $k = 10$, the performance of BKS is slightly better than that of MTKS, which indicates that MTKS has a larger constant factor. Third, according to Figure 1 (d), when k increases, the number of comparisons conducted by both algorithms tend to increase except for a small set of points. When $m = 1$, the increasing rate of the number of conducted comparisons with k is smaller than that when $m = 10$, which is consistent with the theory that the sample complexities of both algorithms depend on $(\frac{n}{m} + k)$. Thus, when m is larger, the complexity of the algorithms are more sensitive with k.

In Figure 1 (e,f), we present the numerical results of MQSort. To show how the full-ranking feedback can help reduce the sample complexity for finding the full ranking, we compare MQSort with Multi-wise Merge Sort (MMS) algorithm proposed in [17], which uses the winner feedback. In theory, to rank n items, MQSort uses $O(\frac{n \log n}{m \log m})$ comparisons and MMS uses $O(\frac{n \log n}{\log m})$ comparisons. In Figure 1 (e), we set $n = 1000$ and vary the value of m. In Figure 1 (f), we set $m = \{2, 5, 10, 50, 100\}$ and vary the value of n. First, from Figure 1 (e,f), we can see that when m increases, the number of comparisons conducted by MQSort decreases, and the decreasing rate is nearly $\frac{1}{m \log m}$, which is consistent with the theory that the sample complexity of MQSort is $O(\frac{n \log n}{m \log m})$. This indicates that by using multi-wise full-ranking feedback with large values of m, we can significantly reduce the sample complexity for finding the full ranking. Second, we can also see that MQSort uses less comparisons than MMS for $m \geq 5$, and the decreasing rate of the complexity is also faster than MMS, consistent with the theory that MQSort has sample complexity $O(\frac{n \log n}{m \log m})$ and MMS has sample complexity $O(\frac{n \log n}{\log m})$. This suggests that compared to winner feedback, using full-ranking feedback is more efficient in terms of sample complexity for finding the full ranking.

8 Conclusion

This paper studied the problems of selecting the top-k items or finding the full ranking of a set of n items by using m-wise comparisons under the winner feedback model or the full-ranking feedback model. The comparisons can be either deterministic or non-deterministic. For all eight combinations of settings (top-k selection or full ranking, winner feedback or full-ranking feedback, deterministic or non-deterministic feedback), we proposed algorithms, derived upper bounds, and/or proved lower bounds. For four settings, we obtained tight upper and lower bounds (up to constant factors). For three settings, we obtained upper and lower bounds where there is only logarithmic gaps between them. The results in this paper showed that by using multi-wise comparisons, one could dramatically reduce the number of comparisons needed for the ranking problems compared to ranking from pairwise comparisons. The numerical results presented also confirmed our theoretical predictions.

Acknowledgments and Disclosure of Funding

This work has been supported in part by NSF grants CAREER CNS-2110259, CNS-2112471, CNS-2102233, CNS-2112471, CNS-2106932, CNS-1955535, CCF-2110252, ECCS-2140277, IIS-2112471, and a Google Faculty Research Award.

References

[1] Blum, M., Floyd, R. W., Pratt, V. R., Rivest, R. L., and Tarjan, R. E. (1973). Time bounds for selection. *J. Comput. Syst. Sci.*, 7(4):448–461.

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

[3] Braverman, M., Mao, J., and Peres, Y. (2019). Sorted top-k in rounds. In *Conference on Learning Theory*, pages 342–382. PMLR.

[4] Chen, X., Li, Y., and Mao, J. (2018). A nearly instance optimal algorithm for top-k ranking under the multinomial logit model. In *Annual ACM-SIAM Symposium on Algorithms*, pages 2504–2522. SIAM.

[5] Falahatgar, M., Hao, Y., Orlitsky, A., Pichapati, V., and Ravindrakumar, V. (2017a). Maxing and ranking with few assumptions. In *Advances in Neural Information Processing Systems*, pages 7060–7070.

[6] Falahatgar, M., Jain, A., Orlitsky, A., Pichapati, V., and Ravindrakumar, V. (2018). The limits of maxing, ranking, and preference learning. In *International Conference on Machine Learning*, pages 1427–1436. PMLR.

[7] Falahatgar, M., Orlitsky, A., Pichapati, V., and Suresh, A. T. (2017b). Maximum selection and ranking under noisy comparisons. In *International Conference on Machine Learning*, pages 1088–1096. JMLR. org.

[8] Fano, R. M. (1961). Transmission of information. *Bell System technical journal*.

[9] Feige, U., Raghavan, P., Peleg, D., and Upfal, E. (1994). Computing with noisy information. *SIAM Journal on Computing*, 23(5):1001–1018.

[10] Heckel, R., Shah, N. B., Ramchandran, K., Wainwright, M. J., et al. (2019). Active ranking from pairwise comparisons and when parametric assumptions do not help. *The Annals of Statistics*, 47(6):3099–3126.

[11] Heckel, R., Simchowitz, M., Ramchandran, K., and Wainwright, M. J. (2018). Approximate ranking from pairwise comparisons. In *International Conference on Artificial Intelligence and Statistics*, pages 1057–1066.

[12] Hoare, C. A. (1961). Algorithm 65: find. *Communications of the ACM*, 4(7):321–322.

[13] Katariya, S., Jain, L., Sengupta, N., Evans, J., and Nowak, R. (2018). Adaptive sampling for coarse ranking. In *International Conference on Artificial Intelligence and Statistics*, pages 1839–1848.

[14] McFadden, D. (1973). Conditional logit analysis of qualitative choice behavior.

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

[16] Ren, W., Liu, J., and Shroff, N. B. (2018). PAC ranking from pairwise and listwise queries: Lower bounds and upper bounds. *arXiv preprint arXiv:1806.02970*.

[17] Ren, W., Liu, J., and Shroff, N. B. (2019). On sample complexity upper and lower bounds for exact ranking from noisy comparisons. In *Advances in Neural Information Processing Systems*, pages 10014–10024.

[18] Ren, W., Liu, J., and Shroff, N. B. (2020). The sample complexity of best-k items selection from pairwise comparisons. In *International Conference on Machine Learning*, pages 8051–8072. PMLR.

[19] Saha, A. and Gopalan, A. (2019a). Active ranking with subset-wise preferences. In *International Conference on Artificial Intelligence and Statistics*, pages 3312–3321.

[20] Saha, A. and Gopalan, A. (2019b). From PAC to instance-optimal sample complexity in the Plackett-Luce model. *arXiv preprint arXiv:1903.00558*.

[21] Shah, N. B. and Wainwright, M. J. (2017). Simple, robust and optimal ranking from pairwise comparisons. *The Journal of Machine Learning Research*, 18(1):7246–7283.

[22] Szörényi, B., Busa-Fekete, R., Paul, A., and Hüllermeier, E. (2015). Online rank elicitation for Plackett-Luce: A dueling bandits approach. In *Advances in Neural Processing Systems*, pages 604–612.

Supplementary Material

A Proofs

A.1 Proof of Theorem 1

Theorem 1. *To find the top-k items from n items by m-wise deterministic winner feedback, any algorithm needs to conduct at least $\Omega(\frac{n}{m}+k)$ comparisons.*

Proof. When $m=2$, i.e., for k-selection from pairwise comparisons, it is well-known that one needs at least $\Omega(n)$ pairwise comparisons to find the top-k items. Any m-wise comparison can be simulated by $(m-1)$ pairwise comparisons. Thus, by using m-wise comparisons, one needs at least $\Omega(\frac{n}{m})$ comparisons for finding the top-k items. Otherwise, we would get a contradiction against the lower bound for top-k items selection from pairwise comparisons.

We then show the $\Omega(k)$ lower bound. We simplify the problem and assume that the learner has already found $r_1, r_2, ..., r_k, r_{k+1}$ and the remaining goal is to identify r_{k+1}. Here, for any i, r_i is the unknown i-th best item. For each m-wise comparison, we can exclude one item, i.e., the winner, from our consideration. Thus, we need k comparisons to exclude all of $r_1, r_2, ..., r_k$, which implies the $\Omega(k)$ sample complexity lower bound. The proof is complete by summing up the above two lower bounds. □

A.2 Proof of Theorem 2

Theorem 2. *MQSelect returns the top-k items of S after $O(\frac{n}{m}+k)$ comparisons in expectation.*

Proof. In the algorithm, we randomly choose $(m-1)$ items as the pivots. These pivots separate the other items into m piles according to the ranks of these items. These piles are denoted as $A_1, A_2, ..., A_m$. Let τ be the first iteration t such that $|D_t| \geq k$, i.e., the algorithm either terminates or enters the call of QS at iteration τ. When the algorithm returns at the last line, we let $\tau = m$. We recall $n = |S|$ is the total number of items.

The key of the proof is to upper bound the size of A_τ, which is stated in Lemma 15. The proof of Lemma 15 is provided in Section A.13 of the supplementary material.

Lemma 15. $\mathbb{E}[|A_\tau|] \leq 8 + \frac{2n}{m}$.

For each iteration t, MQSelect finds one pivot v_t, and thus, the number of comparisons used for finding pivots is at most τ. Let i be an item in A_t. When $t \leq \min\{\tau, m-1\}$, item i wins a comparison that involves pivot v_t and also loses one comparison with each of the pivots $v_1, v_2, ..., v_{t-1}$. When $t = m$, item i loses $m-1$ comparisons, each with a pivot, and wins no comparison with the pivots. For any time that a pivot wins a comparison, $m-1$ items will each lose a comparison if we ignore those comparisons over sets with sizes smaller than m. When taking these sets with smaller sets into consideration, the expected number of items compared in these sets is at least $\frac{m+1}{2}$, which implies at least $\frac{m-1}{2}$ losers in expectation for each comparison. Thus, the number of comparisons required to identify $A_1, A_2, ..., A_\tau$ is at most

$$\sum_{t=1}^{\tau}\Big[|A_t|\cdot\Big(1+\frac{2(t-1)}{m-1}\Big)\Big] \leq 5\sum_{t=1}^{\tau}|A_t|.$$

Therefore, except the call of QS (if it exists), MQSelect conducts at most $\tau+5\sum_{t=1}^{\tau}|A_t|$ comparisons. This implies

$$T(S,m,k) \leq \tau + 5\sum_{t=1}^{\tau}|A_t| + T(A_\tau,k),$$

where $T(A_\tau,k)$ is the number of comparisons QS needs for top-k selection from A_τ. We have $T(A_\tau,k) = O(|A_\tau|)$ and $\tau \leq k$. Also, since MQSelect returns at the first iteration t where $|D_t| \geq k$, we have

$$\sum_{t=1}^{\tau} |A_t| \leq |D_\tau| \leq k + |A_\tau|.$$

Therefore, we have

$$T(S, m, k) \leq k + 5(k + |A_\tau|) + O(|A_\tau|) = O(k + |A_\tau|),$$

which along with Lemma 15 ($\mathbb{E}[|A_\tau|] \leq 8 + \frac{2n}{m}$) completes the proof. □

A.3 Proof of Proposition 3

Proposition 3. *There is an n-sized instance such that to find the top-k items with confidence $1 - \delta$ by using m-wise non-deterministic winner feedback, any algorithm needs at least $\Omega(k + \frac{n}{m} \log \frac{k}{\delta})$ comparisons in expectation.*

Proof. Towards contradiction, we assume that there is an algorithm $\mathcal{A}$ that can find the top-k items of $[n]$ by using $o(\frac{n}{m} \log \frac{k}{\delta})$ m-wise comparisons in expectation. In the proof, we assume that every comparison returns the correct result with probability $\frac{2}{3}$, and will not explicitly state this for simplicity.

We recall that the authors of [9, 17] proved that there is an algorithm that can find the best item among m items with confidence $\frac{2}{3}$ by using $O(m)$ pairwise comparisons. Thus, for any set with size m, we can use $O(m)$ pairwise comparisons with error probability $\frac{1}{3}$ to simulate an m-wise comparison with error probability at most $\frac{1}{3}$. According to our assumption, since algorithm $\mathcal{A}$ can find the top-k items of S with confidence $1 - \delta$ by $o(\frac{n}{m} \log \frac{k}{\delta})$ m-wise comparisons, by simulating the m-wise comparisons by pairwise comparisons, we can construct an algorithm which finds the top-k items of S with confidence $1 - \delta$ by $o(n \log \frac{k}{\delta})$ pairwise comparisons. However, the work in [9] proved that top-k selection with confidence $1 - \delta$ requires at least $\Omega(n \log \frac{k}{\delta})$ pairwise comparisons, leading to a contradiction. Thus, such algorithm $\mathcal{A}$ does not exist, and the $\Omega(\frac{n}{m} \log \frac{k}{\delta})$ lower bound follows.

The $\Omega(k)$ lower bound follows from the lower bound for top-k selection from deterministic comparisons stated in Theorem 1. Combining these two lower bounds, we get the desired lower bound and complete the proof of Proposition 3. □

A.4 Proof of Lemma 4

Lemma 4. *BKS terminates after $O((\frac{n}{m} + k) \log \frac{nk \log m}{m\delta})$ comparisons in expectation, and with probability at least $1 - \delta$, returns the top-k items of S.*

Proof. Similar to the proof of Theorem 1, we let τ be the first iteration t such that $|D_t| \geq k$, i.e., the algorithm either terminates or enters the call of QS at iteration τ. When the algorithm returns at the last line, we let $\tau = m$. We recall $n = |S|$ is the total number of items. To prove the lemma, we need to show that in the execution of BKS, A_τ is of size at most $T_1 = O(\frac{n}{m} \log \frac{m}{\delta})$ with a probability at least $1 - \delta_0$, where $\delta_0 := \frac{\delta}{3}$.

We let $X_1, X_2, ..., X_{m-1}$, L, and R denote the same things as in the proof of Lemma 15 (See Section A.13). We note that $|A_\tau| \leq L + R$. In the proof of Lemma 15, it has been shown in Eq (2) that

$$\mathbb{P}\{L + R = s\} \leq \frac{(m-1)(m-2)(s+1)}{n(n-1)} \cdot \Big(1 - \frac{s-1}{n-2}\Big)^{m-3}.$$

We define $f(s) := \frac{(m-1)(m-2)(s+1)}{n(n-1)} \cdot (1 - \frac{s-1}{n-2})^{m-3}$. When $s \geq \frac{n-m+2}{m-2}$, we have

$$f'(s) = \frac{(m-1)(m-2)}{n(n-1)} \Big[\Big(1 - \frac{s-1}{n-2}\Big)^{m-3} - \frac{(m-3)(s+1)}{n-2}\Big(1 - \frac{s-1}{n-2}\Big)^{m-4}\Big] \leq 0,$$

and thus, for $s \geq \frac{n-m+2}{m-2}$, $f(s)$ is non-increasing.

Now let $s \geq \frac{n-m+2}{m-2}$ be given, and since the maximal of $L+R$ is $n-m+2$, we have

$$\begin{aligned}
\mathbb{P}\{L+R>s\} \leq& (n-m+2-s)\cdot f(s) \\
=& (n-m+2-s)\cdot \frac{(m-1)(m-2)(s+1)}{n(n-1)} \cdot \Big(1-\frac{s-1}{n-2}\Big)^{m-3} \\
\leq& (n-2)\Big(1-\frac{s}{n-2}\Big)\cdot \frac{(m-1)(m-2)(s+1)}{n(n-1)} \cdot \Big(1-\frac{s-1}{n-2}\Big)^{m-3} \\
=& \frac{(n-2)^2(m-1)(m-2)}{n(n-1)} \cdot \frac{s+1}{n-2} \cdot \Big(1-\frac{s-1}{n-2}\Big)^{m-3} \cdot \Big(1-\frac{s}{n-2}\Big) \\
\leq& \frac{(n-2)^2(m-1)(m-2)}{n(n-1)} \cdot \frac{2(s-1)}{n-2} \cdot \Big(1-\frac{s-1}{n-2}\Big)^{m-2} \\
\leq& 2(m-1)(m-2)x(1-x)^{m-2},
\end{aligned}$$

where $x := \frac{s-1}{n-2}$. When $x \geq \frac{1}{m-2}\log\frac{2(m-1)(m-2)}{\delta_0}$, we have

$$\begin{aligned}
\log(2(m-1)(m-2)x(1-x)^{m-2}) =& \log(2(m-1)(m-2)) + \log x + (m-2)\log(1-x) \\
\leq& \log(2(m-1)(m-2)) + 0 - (m-2)x \\
\leq& \log(2(m-1)(m-2)) - \log\frac{2(m-1)(m-2)}{\delta_0} = \log\delta_0,
\end{aligned}$$

and thus, $2(m-1)(m-2)x(1-x)^{m-2} \leq \delta_0$. By $x := \frac{s-1}{n-2}$, we conclude that

$$s \geq 1 + \frac{n}{m-2}\log\frac{2(m-1)(m-2)}{\delta_0} \implies \mathbb{P}\{L+R>s\} \leq \delta_0.$$

Since $|A_\tau| \leq L+R$, we have that

$$|A_\tau| \leq 1 + \frac{n}{m-2}\log\frac{2(m-1)(m-2)}{\delta_0} \text{ with probability at least } 1-\delta_0.$$

We also recall that $T_1 = 1 + \frac{n}{m-2}\log\frac{2(m-1)(m-2)}{\delta_0}$.

Correctness. We let $\mathcal{E}$ be the event that $|A_\tau| \leq T_1$. We have $\mathbb{P}\{\mathcal{E}\} \leq \delta_0$. In the proof of the correctness, we assume that $\mathcal{E}$ happens. Except the call of QS, MQSelect conducts at most $(6k+5|A_\tau|)$ comparisons. Since each comparison is replaced by a call of BC with confidence $1-\frac{\delta_0}{6k+5T_1} \geq 1-\frac{\delta_0}{6k+5|A_\tau|}$, by the union bound, with probability at least $1-\delta_0$, all these calls of BC return correct results. Finally, since the call of QS uses at most $|A_\tau|^2$ comparisons and each comparison is replaced by a call of BC with confidence $1-\frac{\delta_0}{|A_\tau|^2}$, by the union bound, QS returns the correct result with probability at least $1-\delta_0$. Therefore, BKS returns the top-k items of S with probability at least $1-3\delta_0 = 1-\delta$. This proves the correctness.

Sample complexity. By Theorem 2, MQSelect conducts $O(\frac{n}{m}+k)$ comparisons in expectation except the call of QS. In BKS, each comparison is replaced by a call of BC with confidence $1-\frac{\delta_0}{6k+5T_1}$, which conducts $O(\log\frac{6k+5T_1}{\delta_0}) = O(\log\frac{nk\log m}{m\delta_0})$ comparisons. Thus for BKS, Line 2 conducts $O((\frac{n}{m}+k)\log\frac{nk\log m}{m\delta_0})$ comparisons in expectation.

For the call of QS, its expected sample complexity is $\mathbb{E}[O(|A_\tau|)]$. Each comparison of QS is replaced by a call of BC with confidence $1-\frac{\delta_0}{|A_\tau|^2}$, and thus, Line 3 conducts $O(|A_\tau|\log\frac{|A_\tau|}{\delta_0})$ comparisons in expectation. Here, we show Lemma 16 for upper bounding $\mathbb{E}[|A_\tau|\log|A_\tau|]$.

Lemma 16. $\mathbb{E}[|A_\tau|\log|A_\tau|] = O(\frac{n}{m}\log\frac{n}{m})$.

Lemma 16 implies that $O(\mathbb{E}[|A_\tau|\log\frac{|A_\tau|}{\delta_0}]) = O(\frac{n}{m}\log\frac{n}{m\delta})$, and thus, Line 3 conducts at most $O(\frac{n}{m}\log\frac{n}{m\delta})$ comparisons in expectation. By $\delta_0 = \frac{\delta}{3}$, the expected sample complexity of BKS is $O((\frac{n}{m}+k)\log\frac{nk\log m}{m\delta})$. This proves the sample complexity, and the proof of Lemma 4 is complete. □

A.5 Proof of Theorem 5

Theorem 5. *MTKS terminates after $O((\frac{n}{m}+k)\log\frac{k\log m}{\delta})$ comparisons in expectation, and with probability at least $1-\delta$, returns the top-k items of S.*

Proof. **Correctness.** Let t be given, $\delta_t = \frac{6\delta}{\pi^2 t^2}$, and U_t be the set of top-k items of R_t. Since $R_t = \bigcup_{i\in[d]} A_i$ and A_i's are disjoint, for each item u in U_t, there is a set A_{s_u} that contains u. By Lemma 4, for any u in U_t, the call of BKS on set A_{s_u} returns its top-k items with probability at least $1-\frac{6\delta}{\pi^2 t^2 k} = 1-\frac{\delta_t}{k}$, and thus, with probability at least $1-\delta_t$, the calls of BKS on sets $(A_{s_u} : u \in U_t)$ all return their top-k items. Since any u in U_t is one of the top-k items of R_t, and thus, u is also one of the top-k items of A_{s_u}. Therefore, with probability at least $1-\delta_t$, the top-k items of R_t are all added to the set R_{t+1}. By $\sum_{t=1}^{\infty}\delta_t = \frac{6\delta}{\pi^2}\sum_{t=1}^{\infty}\frac{1}{t^2} = \delta$, the correctness of MTKS follows.

Sample complexity. First, we consider the case where $n > mk$. Let τ be the number of rounds the algorithm performs before termination. Since at each time, we divide set R_t into $\lceil\frac{|R_t|}{mk}\rceil$ sets, and for each set, only k items are put to R_{t+1}. Thus, $|R_{t+1}| = k\lceil\frac{|R_t|}{mk}\rceil$, which implies $|R_t| \le \frac{c_1 n}{m^{t-1}}$, where c_1 is some positive constant. For each round t, there are at most $\lceil\frac{|R_t|}{mk}\rceil$ calls of BKS. Each call is on at most mk items. By Lemma 4, each call conducts $O((\frac{mk}{m}+k)\log\frac{mk^2\log m}{m\delta_t}) = O(k\log\frac{k\log m}{\delta_t})$ comparisons in expectation. Thus, the total expected sample complexity of MTKS is upper bounded by

$$\begin{aligned} O\Big(\sum_{t=1}^{\tau}\Big(\frac{n}{m^{t-1}}\cdot\frac{1}{mk}\cdot k\log\frac{k\log m}{\delta_t}\Big)\Big) =& O\Big(\sum_{t=1}^{\tau}\Big(\frac{n}{m^t}\cdot\log\frac{kt\log m}{\delta}\Big)\Big) \\ =& O\Big(\frac{n}{m}\log\frac{k\log m}{\delta}\Big). \end{aligned}$$

Since $n > mk$, i..e, $\frac{n}{m} > k$, the upper bound can also be written as $O((\frac{n}{m}+k)\log\frac{k\log m}{\delta})$.

For the case where $n \le mk$, MTKS directly calls BKS, which yields a sample complexity $O((\frac{n}{m}+k)\log\frac{nk\log m}{m\delta})$. Since $\frac{n}{m} \le k$, the sample complexity reduces to $O((\frac{n}{m}+k)\log\frac{k\log m}{\delta})$. This completes the proof of the sample complexity, and the proof of Theorem 5 is complete. □

A.6 Proof of Theorem 7

Theorem 7. *There is an n-sized instance such that to get the full ranking with confidence $1-\delta$ from m-wise winner feedback, any algorithm needs at least $\Omega(n\log\frac{n}{\delta})$ comparisons.*

Proof. Let $\mathcal{A}$ be an arbitrary algorithm for finding the full ranking by m-wise noisy comparisons under the winner feedback model. We assume that n is even, and the case where n is odd can be proved by ignoring an item. Let $s := \frac{n}{2}$. For a set M and an item i in M, we use $p_{i,M}$ to denote the probability that item i wins the comparison over set M. We recall that the unknown true ranking is $r_1 \succ r_2 \succ \cdots \succ r_n$. Define $\Pi := \{0,1\}^s$, and for each $\pi = (\pi_1, \pi_2, ..., \pi_s)$ in Π, we define the following hypothesis.

Hypothesis $\mathcal{H}_\pi$. The true ranking of $[n]$ is $q_1 \succ q_2 \succ \cdots \succ q_n$ where for any i in $[s]$, $(q_{2i-1}, q_{2i}) = (2i-1, 2i)$ if $\pi_i = 0$ and $(q_{2i-1}, q_{2i}) = (2i, 2i-1)$ otherwise.

We have the following further assumptions. For any value of π and set M, if the best two items of M are not $2i$ and $(2i-1)$ for any i in $[s]$, then for all j in M, $p_{j,M}$ is the same under all values of π; if the best two items of M are $2i$ and $(2i-1)$ for some i in $[s]$, then $(p_{2i-1,M}, p_{2i,M}) = (\frac{2}{3}, \frac{1}{3})$ if $\pi_i = 0$ and $(p_{2i-1,M}, p_{2i,M}) = (\frac{1}{3}, \frac{2}{3})$ if $\pi_i = 1$.

Now, we are interested in the following problem $\mathcal{P}_1$. We note that there is at most one hypothesis $\mathcal{H}_\pi$ holds and we denote it by $\mathcal{H}_{\pi^*}$.

Problem $\mathcal{P}_1$. Assume that there is one π^* in Π such that $\mathcal{H}_{\pi^*}$ is true, and $\mathbb{P}\{\pi^* = \pi\} = \frac{1}{2^s}$ for any π. We want to find the value of π^* with confidence $1-\delta$ by using m-wise comparisons.

First, by using $\mathcal{A}$ to find the true ranking of $[n]$, one can find the value of π_i by checking the whether the $(2i-1)$-th item in the true ranking is $(2i-1)$ or $2i$ for any i. Thus, $\mathcal{A}$ solves the problem $\mathcal{P}_1$.

Second, we show the sample complexity lower bound of $\mathcal{P}_1$. For any set M of which the best two items are not $2i$ or $(2i-1)$, the values of $p_{2i,M}$ and $p_{2i-1,M}$ are the same under any value of π_i, which implies that the comparisons over set M does not contribution any information towards the value of π_i. Thus, only the comparisons over the sets M of which the best two items are $2i$ and $(2i-1)$ can be used to recover the value of π.

Now let $\mathcal{M}_i$ be the collection of sets M_i of which the best two items are $2i$ and $(2i-1)$ for some i in $[s]$. If $\pi_i^* = 0$, then $(p_{2i-1,M_i}, p_{2i,M_i}) = (\frac{2}{3}, \frac{1}{3})$; and if $\pi_i^* = 1$, then $(p_{2i-1,M_i}, p_{2i,M_i}) = (\frac{1}{3}, \frac{2}{3})$. The comparisons over the sets not in $\mathcal{M}_i$ do not provide any information about the value of π_i^*. Therefore, recovering the value of π_i^* is the same as determining whether a Bernoulli distribution is of parameter $\frac{2}{3}$ or $\frac{1}{3}$. According to [17], to determine this parameter with confidence δ_i, at least $\Omega(\log \frac{1}{\delta_i})$ samples are required in expectation, which implies that to recover the value of π_i^* with confidence $1-\delta_i$, at least $\Omega(\log \frac{1}{\delta_i})$ comparisons over sets in $\mathcal{M}_i$ are needed in expectation.

To get the value of π^* with confidence $1-\delta$, we need to find the value of π_i^* with confidence $1-\delta_i$ for any i in $[s]$, where $\prod_{i\in[s]}(1-\delta_i) \geq 1-\delta$. Thus, to solve $\mathcal{P}_1$ with confidence $1-\delta$, the expected number of comparisons required is at least

$$\Omega\Big(\min\Big\{\sum_{i\in[s]} \log \frac{1}{\delta_i} : \prod_{i\in[s]}(1-\delta_i) \geq 1-\delta\Big\}\Big).$$

We note that the set $\{(\delta_1, \delta_1, ..., \delta_s) : \prod_{i\in[s]}(1-\delta_i) \geq 1-\delta\}$ is convex, and the function $\sum_{i\in[s]} \log \frac{1}{\delta_i}$ is convex with respect to $(\delta_1, \delta_1, ..., \delta_s)$. Thus, by Jasen's Inequality, we have

$$\sum_{i\in[s]} \log \frac{1}{\delta_i} \geq s \log \frac{s}{\delta} = \Omega\Big(n \log \frac{n}{\delta}\Big),$$

which implies that the sample complexity lower bound of the problem $\mathcal{P}_1$ is $\Omega(n \log \frac{n}{\delta})$. Since $\mathcal{A}$ solves $\mathcal{P}_1$, the sample complexity of $\mathcal{A}$ is also lower bounded by $\Omega(n \log \frac{n}{\delta})$. Algorithm $\mathcal{A}$ is arbitrary, and this completes the proof of Theorem 7. □

A.7 Proof of Theorem 8

Theorem 8. *To get the top-k items of n items from deterministic m-wise full-ranking feedback, $\Theta(\frac{n}{m})$ comparisons are necessary and sufficient.*

Proof. **Lower bound.** For the lower bound, we can see that to get the top-k items, each item needs to be involved in at least one comparison, and thus, we need at least $\frac{n}{m}$ comparisons. The desired lower bound follows.

Upper bound. The $O(\frac{n}{m})$ sample complexity upper bound can be achieved by the algorithm described in Algorithm 6. We first present this algorithm and then prove the upper bound.

The proof of the upper bound of MQSelect-FRF follows the similar steps as that of Theorem 2. When $|S| \leq m$, the number of comparisons required is 1 and the upper bound follows. In the rest of the proof, we assume $|S| > m$.

From Line 4 to Line 15, MQSelect-FRF conducts at most $(1 + \lceil \frac{|S|-h}{m-h} \rceil) = O(\frac{n}{m})$ comparisons. Similar to MQSelect, MQSelect-FRF randomly chooses h pivots and split the rest of the items to $(h+1)$ piles according to there order relationship with the pivots. Define $\tau := \inf\{t : |D_t| \geq k\}$. If the algorithm returns at the last line, we let $\tau = h+1$. With the same steps as in the proof of Lemma 15, we have $\mathbb{E}|A_\tau| \leq 8 + \frac{2n}{h+1}$. QS conducts $O(|A_\tau|) = O(\frac{n}{m})$ comparisons in expectation. Thus, the total number of comparisons conducted by MQSelect-FRF is $O(\frac{n}{m})$ in expectation. This completes the proof of Theorem 8. □

A.8 Proof of Proposition 9

Proposition 9. *There is an n-sized instance such that to find the top-k items by m-wise non-deterministic full-ranking feedback, any algorithm needs $\Omega(\frac{n}{m} \log_m \frac{k}{\delta})$ comparisons.*

Algorithm 6 Multi-wise Quick-Select from Full-Ranking Feedback(S, m, k) (MQSelect-FRF)

```
if $|S| \le m$ then
    Compare $S$ and return the top-$k$ items according to the returned full ranking;
end if
Randomly choose $h := \lfloor \frac{m}{2} \rfloor$ items and form pivot set $V$;
Compare $V$ to get its full ranking $v_1 \succ v_2 \succ \cdots \succ v_h$;
Define $v_{h+1}$ as an item that ranks lower than any other item; Define $v_0$ as an items that ranks
higher than any other item;
$R \leftarrow S - V$; $A_i \leftarrow \emptyset$ for $i = 1, 2, ..., h+1$;
while $R \neq \emptyset$ do
    Choose $(m-h)$ items from $R$ and form set $E$;
    Compare set $E \cup V$ and get the ranking;
    for item $i$ in $E$ do
        Add item $i$ to set $A_j$ if $v_{j-1} \succ i \succ v_j$;
    end for
    $R \leftarrow R - E$;
end while
$D_0 \leftarrow \emptyset$;
for $t = 1, 2, ..., h$ do
    $D_t \leftarrow D_{t-1} \cup A_t \cup \{v_t\}$;
    if $|D_t| = k$ then
        return $D_t$;
    else if $|D_t| = k+1$ then
        return $D_t - \{v_t\}$;
    else if $|D_t| > k+1$ then
        return $D_{t-1} \cup$ QS$(A_t, k - |D_{t-1}|)$;
    end if
end for
return $D_h \cup$ QS$(A_{h+1}, k - |D_h|)$;
```

Proof. First, we note that by Theorem 12 of [17], to get the full ranking of m items with confidence $\frac{2}{3}$ from non-deterministic pairwise comparisons, $O(m \log m)$ comparisons are sufficient. In other words, we can simulate an m-wise full-ranking-feedback comparison by $O(m \log m)$ pairwise comparisons. We then note that by Theorem 12 of [17], to get the top-k items of $[n]$ with confidence $1-\delta$ from pairwise non-deterministic comparisons, at least $\Omega(n \log \frac{k}{\delta})$ comparisons are needed. Thus, if there is an algorithm $\mathcal{A}$ that can find the top-k items with confidence $1-\delta$ by $o(\frac{n}{m} \log_m \frac{k}{\delta})$ comparisons, then by using pairwise comparisons to simulate m-wise comparisons, we can find the top-k items with confidence $1-\delta$ by $o(n \log \frac{k}{\delta})$ pairwise comparisons. This contradicts the lower bound stated in Theorem 12 of [17]. Therefore, such algorithm $\mathcal{A}$ does not exist and the desired lower bound follows. This completes the proof of Proposition 9. □

A.9 Proof of Theorem 10

Theorem 10. *MTKS-FRF terminates after* $O(\frac{n}{m} \log \frac{\min\{n/m, k\} \log m}{\delta})$ *comparisons in expectation, and with probability at least* $1-\delta$*, returns the top-*k *items of* S.

Proof. We first introduce the subroutine BKS-FRF (Basic k-selection from Full-Ranking Feedback), which is described in Algorithm 7. Its theoretical performance is formally stated in Lemma 17.

Lemma 17. *BKS-FRF terminates after* $O(\frac{n}{m} \log \frac{n \log m}{m\delta})$ *comparisons in expectation, and with probability at least* $1-\delta$*, returns the top-*k *items of* S.

Then, similar to MTKS, we develop a tournament-like algorithm MTKS-FRF (Multi-wise Tournament k-Selection from Full-Ranking Feedback), which is described in Algorithm 8.

Correctness. Let t be given, $\delta_t = \frac{6\delta}{\pi^2 t^2}$, and U_t be the set of top-k items of R_t. Since $R_t = \bigcup_{i \in [d]} A_i$ and A_i's are disjoint, for each item u in U_t, there is a set A_{s_u} that contains u. By Lemma 17,

Algorithm 7 BKS from Full-Ranking Feedback(S, m, k, δ) (BKS-FRF)

```
$h \leftarrow \lfloor \frac{m}{2} \rfloor$; $\delta_0 \leftarrow \frac{\delta}{3}$; $T_2 \leftarrow 1 + \frac{n}{h-1} \log \frac{2h(h-1)}{\delta}$; $\delta_1 \leftarrow \frac{\delta_0}{1+\frac{n-h}{m-h}+T_2}$ if $m > 2$; $\delta_1 \leftarrow \frac{\delta_0}{|S|^2}$ if $m = 2$;
Run MQSelect-FRF on $M$, but using calls of BC with confidence $1 - \delta_1$ to replace multi-wise comparisons (except those in the call of QS);
For the call of QS, we replace the pairwise comparisons with calls of BC with confidence $1 - \frac{\delta_0}{|A_t|^2}$;
```

Algorithm 8 MTKS from Full-Ranking Feedback(S, m, k, δ) (MTKS-FRF)

```
if $|R_t| \leq mk$ then
    return BKS-FRF$(S, m, k, \delta)$;
end if
Set $t \leftarrow 0$ and $R_1 \leftarrow S$;
repeat
    $t \leftarrow t + 1$; $\delta_t \leftarrow \frac{6\delta}{\pi^2 t^2}$;
    Distribute $R_t$ to $\lceil \frac{|R_t|}{mk} \rceil$ disjoint sets $A_1, A_2, ..., A_d$, each with size at most $mk$;
    For $i \in [d]$, let $T_i \leftarrow$BKS-FRF($A_i, m, k, \frac{\delta_t}{k}$);
    $R_{t+1} \leftarrow \bigcup_{i \in [d]} A_i$;
until $|R_{t+1}| = k$
return $R_{t+1}$;
```

for any u in U_t, the call of BKS-FRF on set A_{s_u} returns its top-k items with probability at least $1 - \frac{6\delta}{\pi^2 t^2 k} = 1 - \frac{\delta_t}{k}$, and thus, with probability at least $1 - \delta_t$, the calls of BKS-FRF on sets $(A_{s_u} : u \in U_t)$ all return their top-k items. Since any u in U_t is one of the top-k items of R_t, and thus, u is also one of the top-k items of A_{s_u}. Therefore, with probability at least $1 - \delta_t$, the top-k items of R_t are all added to the set R_{t+1}. By $\sum_{t=1}^{\infty} \delta_t = \frac{6\delta}{\pi^2} \sum_{t=1}^{\infty} \frac{1}{t^2} = \delta$, the correctness of MTKS-FRF follows.

Sample complexity. First, we consider the case where $n > mk$. Let τ be the number of rounds the algorithm performs before termination. Since at each time, we divide set R_t into $\lceil \frac{|R_t|}{mk} \rceil$ sets, and for each set, only k items are put to R_{t+1}. Thus, $|R_{t+1}| = k \lceil \frac{|R_t|}{mk} \rceil$, which implies $|R_t| \leq \frac{c_1 n}{m^{t-1}}$, where c_1 is some positive constant. For each round t, there are at most $\lceil \frac{|R_t|}{mk} \rceil$ calls of BKS-FRF. Each call is on at most mk items, and by Lemma 17, each call conducts $O(\frac{mk}{m} \log \frac{mk \log m}{m\delta_t}) = O(k \log \frac{k \log m}{\delta_t})$ comparisons in expectation. Thus, the total sample complexity of MTKS is upper bounded by

$$
\begin{aligned}
O\Big(\sum_{t=1}^{\tau} \Big(\frac{n}{m^{t-1}} \cdot \frac{1}{mk} \cdot k \log \frac{k \log m}{\delta_t}\Big)\Big) =& O\Big(\sum_{t=1}^{\tau} \Big(\frac{n}{m^t} \cdot \log \frac{kt \log m}{\delta}\Big)\Big) \\
=& O\Big(\frac{n}{m} \log \frac{k \log m}{\delta}\Big).
\end{aligned}
$$

For the case where $n \leq mk$, MTKS directly calls BKS-FRF, which yields a sample complexity $O(\frac{n}{m} \log \frac{n \log m}{m\delta})$. This completes the proof of the sample complexity, and the proof of Theorem 10 is complete. □

A.10 Proof of Theorem 11

Theorem 11. *To get the full ranking of n items by using m-wise comparisons under the full-ranking feedback model, any algorithm needs at least $\Omega(\frac{n \log n}{m \log m})$ comparisons in expectation.*

Proof. Let $\mathcal{A}$ be an arbitrary deterministic algorithm. Let R be the permutation representing the true ranking. Since there are n items, and each permutation has the same probability to be the true ranking, we have $H(R) = \log(n!)$, where $H(\cdot)$ is the information entropy.

Let T be the number of comparisons conducted by $\mathcal{A}$. Let $M_t := \{i_{t,1}, i_{t,2}, ..., i_{t,m}\}$ be the t-th compared set and $\mathcal{M} := (M_1, M_2, ..., M_T)$ be the collection of compared sets. For the t-th comparison on M_t, if the returned true ranking is $(i_{t,s_1}, i_{t,s_2}, ..., i_{t,s_m})$, then we let $X_t = (s_1, s_2, ..., s_m)$, and we use $\mathcal{X} = (X_1, X_2, ..., X_T)$ to denote the collection of comparison results. Define $S_t := (M_t, X_t, \mathbb{1}_{t \le T})$ as the state of the t-th comparison and $U_t := (S_1, S_2, ..., S_t)$ as the history till the t-th comparison.

Since the algorithm is deterministic, M_1 is deterministic, M_t is determined by U_{t-1}, and $\mathbb{1}_{t \le T}$ is determined by (U_{t-1}, M_t, X_t). Thus, we have

$$\begin{aligned}
H(U_t \mid U_{t-1}) =& H(S_t \mid U_{t-1}) \\
\le& H(M_t \mid U_{t-1}) + H(X_t \mid M_t, U_{t-1}) + H(\mathbb{1}_{t \le T} \mid M_t, X_t, U_{t-1}) \\
\le& 0 + H(X_t \mid M_t) + 0 \\
=& \log(|M_t|!) \\
=& \log(m!).
\end{aligned}$$

where $H(\cdot \mid \cdot)$ is the conditional information entropy. Then, we have

$$\begin{aligned}
H(U_t) =& H(U_t \mid U_{t-1}) + H(U_{t-1} \mid U_{t-2}) + \cdots + H(U_1) \\
\le& \log(m!) + \log(m!) + \cdots + \log(m!) \\
=& t \log(m!).
\end{aligned}$$

Since the comparisons are deterministic, when the true ranking R is given, the values of all U_t's are deterministic, i.e., $H(U_t \mid R) = 0$. Therefore, we have

$$\begin{aligned}
H(R \mid U_t) =& H(R) - I(R; U_t) \\
=& H(R) - (H(U_t) - H(U_t \mid R)) \\
\ge& \log(n!) - t \log(m!) + 0,
\end{aligned}$$

where $I(\cdot;\cdot)$ is the mutual information between two random vectors.

We then invoke Fano's Inequality, which is stated in Fact 18.

Fact 18 (Fano's Inequality [8])**.** *To recover the value of X from Y with error probability no more than δ, it must hold that*

$$H(X|Y) \le H(\delta) + \delta \log(N-1),$$

where N is the number of values X can take and $H(\delta) = \delta \log \frac{1}{\delta} + (1-\delta) \log \frac{1}{1-\delta}$.

Thus, to recover the true ranking (i.e., the value of R) with probability at least $\frac{1}{4}$ within t comparisons, by Fano's Inequality, $\log(m!) = \Theta(m \log m)$, and $\log(n!) = \Theta(n \log n)$, it is required that

$$\begin{aligned}
& H(R \mid U_t) \le H(1/4) + \delta \log(n! - 1) \\
\Longrightarrow\ & \log(n!) - t \log(m!) + 0 \le \Theta(1) + \frac{1}{4} \log(n! - 1) \\
\Longrightarrow\ & t \ge \frac{\log(n!) - \frac{1}{4}\log(n!-1) - \Theta(1)}{\log(m!)} \\
\Longrightarrow\ & t = \Omega\Big(\frac{n \log n}{m \log m}\Big).
\end{aligned}$$

Any random algorithm cannot outperform the best deterministic algorithm, and thus, the desired lower bound also holds for random algorithms. This completes the proof of Theorem 11. □

A.11 Proof of Theorem 12

Theorem 12. *MQSort terminates after $O(\frac{n \log n}{m \log m})$ m-wise comparisons in expectation and returns the full ranking of S.*

Proof. Let set S and $n = |S|$ be given. We say that the call of MQSort on S is executed in round 1. In Line 16, it makes multiple recursive calls of MQSort and we say these calls are executed in round 2. Similarly, a call of MQSort executed in round t may also make recursive calls and we say these recursive calls are executed in round $(t+1)$. We observe that during the execution of a call on n' items, splitting the items to sets A_i's takes at most $1 + \lceil \frac{|S-V|}{m-h} \rceil = O(\frac{n'}{m})$ comparisons. Since in each round t, the total number of items processed by all the recursive calls is at most n, the splitting processes in all these recursive calls use at most $O(\frac{n}{m})$ comparisons in total. Thus, the rest of the proof is to prove that the recursive calls on terminate after $O(\log_m n)$ rounds in expectation.

Step 1. Before finding the number of rounds of MQSort, we focus on a similar problem. We say that there are N balls in total where N is large enough. Then at round 1 we randomly put these N balls into M boxes where M is large enough and $M \le N$. In round 2, for each box, we randomly put the balls inside it into M other boxes, i.e., the balls are now randomly distributed to M^2 boxes. For any round t where $t > 2$, we do the same thing, i.e., randomly put the balls of each box into M other boxes. We repeat this process until each box contains at most one ball. Let T be the number of rounds before this process terminates. Our goal is to find an upper bound of $\mathbb{E}[T]$.

Now let $t \ge \log_M N$ be given, i be a random ball, and j be a random box that is introduced to this process in round t. We have that ball i is in box j with probability $\frac{1}{M^t}$. The places of these balls are independent, and thus, we have

$$\mathbb{P}\{T \le t\} = \frac{M^t}{M^t} \cdot \frac{M^t - 1}{M^t} \cdot \frac{M^t - 2}{M^t} \cdots \frac{M^t - N + 1}{M^t} \ge \Big(\frac{M^t - N + 1}{M^t}\Big)^N.$$

Now we let $\delta \in (0,1)$ be given. We define

$$\tau_\delta := 1 + \log_M \frac{N-1}{1-(1-\delta)^{1/N}},$$

we have

$$\mathbb{P}\{T \le \tau_\delta\} \ge \Big(\frac{M^{\tau_\delta} - N + 1}{M^{\tau_\delta}}\Big)^N = \Big(1 - \frac{N-1}{M^{\tau_\delta}}\Big)^N = \Big(1 - \frac{N-1}{M^{\log_M \frac{N-1}{1-(1-\delta)^{1/N}}}}\Big)^N = 1 - \delta.$$

Also, since $1-(1-\delta)^{1/N} \ge 1 - e^{-\delta/N} = \Omega(\frac{\delta}{N})$, we have

$$\tau_\delta = \log_M \Big(\frac{N-1}{1-(1-\delta)^{1/N}}\Big) = O\Big(\log_M \frac{N}{\delta}\Big)$$

Therefore, by $\tau_1 = 1 + \log_M(N-1)$, we conclude that

$$\begin{aligned}
\mathbb{E}[T] =& \sum_{t=0}^{\infty} \mathbb{P}\{T > t\} \\
\le& \sum_{t > \tau_1} \mathbb{P}\{T > t\} + \tau_1 \\
\le& \sum_{s=0}^{\infty} [(\tau_{2^{-s-1}} - \tau_{2^{-s}}) \cdot \mathbb{P}\{t > \tau_{2^{-s}}\}] + O(\log_M N) \\
\le& \sum_{s=0}^{\infty} [\tau_{2^{-s-1}} \cdot \mathbb{P}\{t \ge \tau_{2^{-s}}\}] + O(\log_M N) \\
=& O\Big(\sum_{s=1}^{\infty} \log_M \frac{N}{2^{-s-1}} \cdot 2^{-s}\Big) + O(\log_M N) \\
=& O(\log_M N).
\end{aligned}$$

Step 2. We switch back to MQSort. We define $h = \lfloor \frac{m}{2} \rfloor$ and there are h pivots among n items that separate the set S to $(h+1)$ sets $A_0, A_1, ..., A_h$. For any integer i, we define $[i]_n := i \bmod n$, i.e., the reminder of i divided by n. We introduce a dummy item $v_0 := 0$ that ranks higher than any other item. Let X and Y be two random and distinct items in $\{v_0\} \cup S$ and we assume that the algorithm has finished Line 15, i.e., all the non-pivot items have been added to one of $A_0, A_1, ..., A_h$. We further define $B_j := A_j \cup \{v_j\}$ for any j. In the following analysis, we view pivots $v_1, v_2, ..., v_h$ as random distinct items selected from S. First we have

$$\mathbb{P}\{X \in A_0\} = \mathbb{P}\{\forall i > 0, v_i > X \ge 0\} = \mathbb{P}\{\forall i \ne 0, [v_i - v_0]_n > [X - v_0]_n\}.$$

For any set A_j where $j > 0$, we have

$$\mathbb{P}\{X \in A_j\} = \mathbb{P}\{\forall i \neq j, [v_i - v_j]_n > [X - v_j]_n\}.$$

Therefore, when we change the set from A_i to A_j, actually we are simply rotationally shifting the items with $[v_i - v_j]_n$ steps, which implies that $\mathbb{P}\{X \in A_0\} = \mathbb{P}\{X \in A_j\}$ for any j. Thus, for any j, we have

$$\mathbb{P}\{X \in A_j\} = \frac{1}{h} \text{ and } \mathbb{P}\{X \in A_j | X \neq v_j\} = \frac{1}{h}.$$

Now we assume that X is not a pivot and X is in the set A_j. In this case, we can define $S' = S - \{X\}$ and subtract the indexes of all items $i > X$ by one, and then the probabilities whether Y is in A_0, A_1,..., or A_h remain the same, i.e., for any j and l, we have

$$\mathbb{P}\{Y \in A_j \mid X \in A_l, X \neq v_l\} = \frac{1}{h} \text{ and } \mathbb{P}\{Y \in A_j \mid X \in A_l, X \neq v_l, Y \neq V_j\} = \frac{1}{h}.$$

Repeating the above analysis. we have that with s non-pivot items given, the probabilities whether the $(s+1)$-th item is in some set A_j remain $\frac{1}{h}$ for $s + 1 \leq n - h$.

Step 3. Therefore, with the above findings, we can view the distribution of non-pivot items as the "put balls into boxes" problem, where the non-pivot items are the balls and the sets $A_0, A_1, ..., A_h$ are the boxes. Since after each round, there are h balls removed from the following splitting and the other conditions remain the same, let round T' be the first round when each box has at most one ball, we have

$$\mathbb{E}[T'] = O(\log_{h+1} n) = O(\log_{\lceil \frac{m}{2} \rceil + 1} n) = O(\log_m n).$$

When each box has at most one ball, i.e., all sets $A_0, A_1, ..., A_h$ in all recursive calls of MQSort in round T' have at most one item, the outer-most call will return, and the algorithm will terminate. Thus, there are $O(\log_m n)$ rounds in expectation. Recalling the fact that each round conducts at most at most $O(\frac{n}{m})$ comparisons, the algorithm returns after $O(\frac{n \log n}{m \log m})$ comparisons in expectation. This completes the proof of Theorem 12. □

A.12 Proof of Theorem 14

Theorem 14. *There is an algorithm that finds the full ranking of n items by m-wise full-ranking feedback with confidence $1 - \delta$ and conducts $O(\frac{n}{m} \log \frac{n}{m\delta})$ comparisons in expectation.*

Proof. To get the desired upper bound, we need to invoke the full ranking algorithm in [9], which is denoted by $\mathcal{A}$. When all comparisons return correct results with probability $\frac{2}{3}$ and $m = 2$, $\mathcal{A}$ finds the true ranking of n items with confidence $1 - \delta$ by using $O(n \log \frac{n}{\delta})$ pairwise comparisons. The algorithm is a variant of the insertion sorting, i.e., given a list of n_1 sorted items, it inserts a new item into this list. The algorithm repeats the insertion until all items have been inserted and the full ranking is found. In $\mathcal{A}$, inserting one item i into a list of n_i sorted items with confidence $1 - \delta_i$ uses $O(\log \frac{n_i}{\delta_i}) = O(\log \frac{n}{\delta_i})$ comparisons.

Now, we make the following modifications on $\mathcal{A}$. First, we randomly choose m items and compare these m items for $\Theta(\log \frac{n}{m\delta})$ times to get the ranking of these m items with confidence $1 - \frac{m\delta}{4n}$. We use R to denote this list. At each round, we choose $\frac{m}{2}$ new items (assuming that m is even, and when m is odd we can prove the upper bound similarly) and we want to insert them into the sorted list R. Since an m-wise comparison under the full-ranking feedback model can be viewed as doing $\frac{m}{2}$ pairwise comparisons at the same time, we do the insertion process for these $\frac{m}{2}$ new items simultaneously. During the insertion, we set the confidence at $1 - \frac{m\delta}{4n}$ and the depth of the insertion tree at $\Theta(\log \frac{n}{m\delta})$. After at most $O(\log \frac{n}{m\delta})$ comparisons, if at least $\frac{2}{3}$ proportion of the comparisons return correct results (which happens with probability at least $1 - \frac{m\delta}{4n}$), then all the $\frac{m}{2}$ new items will be inserted to correct places. We then compare these $\frac{m}{2}$ new items for $\Theta(\log \frac{n}{m\delta})$ times to get the ranking of them with confidence at least $1 - \frac{m\delta}{4n}$.

For each round of insertion, we use $O(\log \frac{n}{m\delta})$ comparisons, and each round inserts $\frac{m}{2}$ items. Thus, the insertion takes $O(\frac{n}{m} \log \frac{n}{m\delta})$ comparisons in total. We do $\Theta(\log \frac{n}{m\delta})$ comparisons for ranking the first m items, and the same amount in each round of insertion for ranking the inserted new items, which takes $O(\frac{n}{m}\log\frac{n}{m\delta})$ comparisons in total for inserting all items. Thus, the total sample complexity of this modified algorithm is at most $O(\frac{n}{m}\log\frac{n}{m\delta})$. This proves the sample complexity.

The ranking of the first m item is correct with probability at least $1-\frac{m\delta}{4n}$. For each round of insertion, the ranking of the newly inserted items is correct with probability at least $1-\frac{m\delta}{4n}$, and the insertion is correct with probability at least $1-\frac{m\delta}{4n}$. There are at most $\frac{n-m}{m/2}=\frac{2n}{m}-2$ rounds of insertion. By the union bound, the total error probability of the modified algorithm is $\frac{m\delta}{4n}(1+\frac{4n}{m}-4)\leq\delta$. This proves the correctness and the proof of Theorem 14 is complete. □

A.13 Proof of Lemma 15

Lemma 15. $\mathbb{E}[|A_\tau|]\leq 8+\frac{2n}{m}$.

Proof. The $(m-1)$ pivots are randomly chosen, and we use $X_1, X_2, ..., X_{m-1}$ to denote their positions in the true ranking, i.e., $X_i = j$ means that X_i ranks the j-th largest in the true ranking. For simplicity, we define $X_0 := \infty$ that ranks higher than any item and $X_m := -\infty$ that ranks lower than any item. If $X_{i-1}\succ r_k \succeq X_i$, then we have $\tau = i$ and r_k is in the set A_i or $r_k = X_i$.

We put these n items on a circle, i.e., we assume that item r_1 is on the right of r_n, r_n is on the left of r_1, and they are adjacent. For any integer a, we define $[a]_n := a \mod n$, where $a \mod n$ is the reminder of a divided by n. We further define

$$L := [k - X_{\tau-1}]_n, \text{ and } R := [X_\tau - k]_n.$$

After putting the n items on the a circle, we can view L as the distance from r_k to the closest pivot on the left, and R as the distance from r_k to the closes pivot on the right. We have $|A_\tau|\leq L+R$, and our goal is to bound $\mathbb{E}[L+R]$.

Let $1\leq s\leq n-m+2$ be given. We have

$$\begin{aligned}\mathbb{P}\{L+R=s\} = &\sum_{i,j\in[m-1]:i\neq j}\sum_{r=0}^{s}\Big[\mathbb{P}\{[k-X_i]_n = r\}\mathbb{P}\{[X_j-k]_n = s-r,\}\\ &\qquad\qquad\qquad\cdot\mathbb{P}\{\forall l\neq i,j : s-r < [X_l-k]_n < n-r\}\Big]\\ =&(m-1)(m-2)\cdot\frac{s+1}{n(n-1)}\cdot\frac{\binom{n-s-1}{m-3}}{\binom{n-2}{m-3}}\\ \leq&(m-1)(m-2)\cdot\frac{s+1}{n(n-1)}\cdot\Big(\frac{n-s-1}{n-2}\Big)^{m-3}\\ =&\frac{(m-1)(m-2)(s+1)}{n(n-1)}\cdot\Big(1-\frac{s-1}{n-2}\Big)^{m-3}. \end{aligned}\tag{2}$$

Thus, we have

$$\begin{aligned}\mathbb{E}[L+S]\leq&\sum_{s=1}^{n-m+2}\Big[s\cdot\frac{(m-1)(m-2)(s+1)}{n(n-1)}\cdot\Big(1-\frac{s-1}{n-2}\Big)^{m-3}\Big]\\ \leq&(m-1)(m-2)\sum_{s=1}^{n-m+2}\Big[\Big(\frac{s-1}{n-2}\Big)^2\cdot\Big(1-\frac{s-1}{n-2}\Big)^{m-3}\Big]\end{aligned}$$

Now, we let $x:=\frac{s-1}{n-2}$. Since in $[0,1]$, the function $f(x):=x^2(1-x)^{m-3}$ is non-negative and first increasing then decreasing, we have the following inequality.

$$
\begin{aligned}
\mathbb{E}[L+S] \leq &(m-1)(m-2)\Big[2\sup_{x\in[0,1]} x^2(1-x)^{m-3} + (n-2)\int_0^1 x^2(1-x)^{m-3}\,\mathrm{d}x\Big] \\
=&(m-1)(m-2)\Big[2x^2(1-x)^{m-3}|_{x=\frac{2}{m}} \\
&\quad + (n-2)\cdot \frac{(x-1)(1-x)^{m-3}[(m^2-3m)x^2+2(m-2)x+2]}{m(m-1)(m-2)}\Big|_0^1\Big] \\
=&(m-1)(m-2)\Big[\frac{8}{m^2}\Big(1-\frac{2}{m}\Big)^3 + (n-2)\cdot\frac{2}{m(m-1)(m-2)}\Big] \\
\leq&(m-1)(m-2)\Big[\frac{8}{(m-1)(m-2)}\cdot 1^2 + n\cdot\frac{0-(-2)}{m(m-1)(m-2)}\Big] \\
=&8+\frac{2n}{m}.
\end{aligned}
$$

Therefore, we have $\mathbb{E}[|A_\tau|] \leq \mathbb{E}[L+R] \leq 8+\frac{2n}{m}$. This completes the proof of Lemma 15. □

A.14 Proof of Lemma 16

Lemma 16. $\mathbb{E}[|A_\tau|\log|A_\tau|] = O(\frac{n}{m}\log\frac{n}{m})$.

Proof. We let $X_1, X_2, ..., X_{m-1}$, L, and R denote the same things as in the proof of Lemma 15 (See Section A.13). We note that $|A_\tau| \leq L+R$. Let $X = |A_\tau|$.

According to Eq (2), we have

$$
\mathbb{P}\{L+R=s\} \leq \frac{(m-1)(m-2)(s+1)}{n(n-1)}\cdot\Big(1-\frac{s-1}{n-2}\Big)^{m-3},
$$

We further use x to denote $\frac{s-1}{n-2}$. The above result, by $X \leq L+R$, implies

$$
\begin{aligned}
E[X\log X] \leq &\frac{(m-1)(m-2)}{n(n-1)}\sum_{s=1}^{n-m+2}\Big[s(s+1)\Big(1-\frac{s-1}{n-2}\Big)^{m-3}\log s\Big] \\
\leq&\frac{(m-1)(m-2)}{n(n-1)}\sum_{s=1}^{n-m+2}\Big[2(s-1)^2\Big(1-\frac{s-1}{n-2}\Big)^{m-3}(1+\log(s-1))\Big] \\
\leq&\frac{(m-1)(m-2)}{n(n-1)}\sum_{s=1}^{n-2}\Big[2(n-2)^2x^2(1-x)^{m-3}(1+\log((n-2)x))\Big].
\end{aligned}
$$

Here, we define $f(x) := x^2(1-x)^{m-3}\log((n-2)x)$. We have

$$
f'(x) = x(1-x)^{m-4}[(2-(m-1)x)\log((n-2)x)+1-x].
$$

When $\frac{1}{n-2} < x \leq 1$, $f'(x)$ is first positive and then negative, though the actual transiting point is unknown. We also have $f(0) = f(\frac{1}{n-2}) = 0$. Thus, we have

$$
\sum_{s=1}^{n-m+2} f(x) \leq (n-2)\int_0^1 f(x)\,\mathrm{d}x - 2\min_{0\leq x\leq\frac{1}{n-2}} f(x) + 2\max_{\frac{1}{n-2}\leq x\leq 1} f(x).
$$

First, we have

$$
\begin{aligned}
\int_0^1 f(x)\,\mathrm{d}x =&\frac{2\log(n-2)}{m^3-3m^2+2m} + \frac{3-2\sum_{i=1}^m\frac{1}{i}}{m^3-3m^2+2m} \\
\leq&\frac{2\log n - 2\log m + O(1)}{m^3-3m^2+2m} \\
=&O\Big(\frac{\log\frac{n}{m}}{m^3}\Big).
\end{aligned}
$$

Second, when $0 \leq x \leq \frac{1}{n-2}$, we have

$$f(x) \geq x^2 \log((n-2)x),$$

which takes the minimum when

$$(x^2 \log((n-2)x))' = 2x \log((n-2)x) + x = 0,$$

i.e., when $x = \frac{e^{-1/2}}{n-2}$. Thus, we have

$$\min_{0 \leq x \leq \frac{1}{n-2}} f(x) \geq -\frac{1}{2}\Big(\frac{e^{-1/2}}{n-2}\Big)^2 = -\Theta(n^{-2}).$$

Third, when $x \geq \frac{1}{n-2}$, we have $f(x) \leq x^2(1-x)^{m-3} \log n$. $x^2(1-x)^{m-3}$ is maximal when $(x^2(1-x)^{m-3})' = 0$, i.e., $x = \frac{2}{m-1}$. Thus,

$$\max_{\frac{1}{n-2} \leq x \leq 1} f(x) \leq \Big(\frac{2}{m-1}\Big)^2 \Big(1 - \frac{2}{m-1}\Big)^{m-3} \log n = O\Big(\frac{\log n}{m^2}\Big).$$

Therefore, we have

$$\begin{aligned}\sum_{s=1}^{n-m+2} f(x) =&(n-2) \cdot O\Big(\frac{\log n}{m^3}\Big) + O\Big(\frac{1}{n^2}\Big) + 2 \cdot O\Big(\frac{\log n}{m^2}\Big)\\ =&O\Big(\frac{n \log \frac{n}{m}}{m^3}\Big).\end{aligned}$$

Similarly, since for $0 \leq x \leq 1$, $x^2(1-x)^{m-3}$ first increases and then decreases, we have

$$\begin{aligned}\sum_{s=1}^{n-m+2} x^2(1-x)^{m-3} \leq&(n-2) \int_0^1 x^2(1-x)^{m-3} \,\mathrm{d}x + 2 \max_{0 \leq x \leq 1} x^2(1-x)^{m-3}\\ =&\frac{2(n-2)}{m^3 - 3m^2 + 2m} + 2\Big(\frac{2}{m-1}\Big)^2 \Big(1 - \frac{2}{m-1}\Big)^{m-3}\\ =&O\Big(\frac{n}{m^3}\Big).\end{aligned}$$

With the above results, we have

$$\begin{aligned}\mathbb{E}[X \log X] =&\frac{(m-1)(m-2)}{n(n-1)} \cdot 2(n-2)^2 \Big(O\Big(\frac{n \log \frac{n}{m}}{m^3}\Big) + O\Big(\frac{n}{m^3}\Big)\Big)\\ =&O\Big(\frac{n}{m} \log \frac{n}{m}\Big).\end{aligned}$$

This completes the proof of Lemma 16. □

A.15 Proof of Lemma 17

Lemma 17. *BKS-FRF terminates after $O(\frac{n}{m} \log \frac{n \log m}{m\delta})$ comparisons in expectation, and with probability at least $1-\delta$, returns the top-k items of S.*

Proof. If QS is called, then we use A_τ to denote the set A_t inputed to QS. To prove the lemma, we need to show that in the execution of MQSelect-FRF, A_τ is of size at most $T_1 = O(\frac{n}{m} \log \frac{m}{\delta})$ with a probability at least $1 - \delta_0$, where $\delta_0 := \frac{\delta}{3}$.

We let $X_1, X_2, ..., X_{m-1}$, L, and R denote the same things as in the proof of Lemma 15. By similar steps as in the proof of Lemma 4, we have that with probability at least $1 - \delta_0$, $|A_\tau| \leq T_2$. Here we note that the only that is changed is the number of pivots (i.e., from $(m-1)$ to h), and we only need to change m to $(h+1)$ in the formula of T_2.

Correctness. We let $\mathcal{E}$ be the event that $|A_\tau| \leq T_2$. We have $\mathbb{P}\{\mathcal{E}\} \leq \delta_0$. In the proof of the correctness, we assume that $\mathcal{E}$ happens. Except the call of QS, MQSelect conducts at most

$(1 + \frac{n-h}{m-h})$ comparisons, where one comparison is for ranking the pivots and $\frac{n-h}{m-h}$ comparisons are for splitting the non-pivot items. Since each comparison is replaced by a call of BC with confidence $1 - \frac{\delta_0}{1+\frac{n-h}{m-h}+T_2}$, by the union bound, with probability at least $1 - \delta_0$, all these calls of BC return correct results. Finally, since the call of QS uses at most $|A_\tau|^2$ comparisons and each comparison is replaced by a call of BC with confidence $1 - \frac{\delta_0}{|A_\tau|^2}$, by the union bound, QS returns the correct result with probability at least $1 - \delta_0$. Therefore, BKS-FRF returns the top-k items of S with probability at least $1 - 3\delta_0 = 1 - \delta$. This proves the correctness.

Sample complexity. By Theorem 2, MQSelect conducts $O(\frac{n}{m})$ comparisons in expectation except the call of QS. In BKS-FRF, each comparison is replaced by a call of BC with confidence $1 - \frac{\delta_0}{1+\frac{n-h}{m-h}+T_2}$, which conducts $O(\log \frac{1+\frac{n-h}{m-h}+T_2}{\delta_0}) = O(\log \frac{n \log m}{m\delta_0})$ comparisons. Thus for BKS-FRF, Line 2 conducts $O(\frac{n}{m} \log \frac{n \log m}{m\delta_0})$ comparisons in expectation. For the call of QS, its expected sample complexity is $\mathbb{E}[O(|A_\tau|)]$. Each comparison of QS is replaced by a call of BC with confidence $1 - \frac{\delta_0}{|A_\tau|^2}$, and thus, by similar steps as the proof of Lemma 16 (we only need to change m to $h + 1$ in the new steps), Line 3 conducts $O(|A_\tau| \log \frac{|A_\tau|}{\delta_0}) = O(\frac{n}{m} \log \frac{n}{m\delta_0})$ comparisons in expectation. Therefore, by $\delta_0 = \frac{\delta}{3}$, the expected sample complexity of BKS -FRF is $O(\frac{n}{m} \log \frac{n \log m}{m\delta})$. This proves the sample complexity, and the proof of Lemma 17 is complete. □