PAC Top-k Identification under SST in Limited Rounds

Arpit Agarwal
Columbia University

Sanjeev Khanna
University of Pennsylvania

Prathamesh Patil
University of Pennsylvania

Abstract

We consider the problem of finding top-k items from a set of n items using actively chosen pairwise comparisons. This problem has been widely studied in machine learning and has widespread applications in recommendation systems, sports, social choice etc. Motivated by applications where there can be a substantial delay between requesting comparisons and receiving feedback, we consider an active/adaptive learning setting where the algorithm uses limited rounds of parallel interaction with the feedback generating oracle.

We study this problem under the strong stochastic transitivity (SST) noise model which is a widely studied ranking model and captures many applications. A special case of this model is the noisy comparison model for which it was recently shown that $O(n \log k)$ comparisons and $\log n$ rounds of adaptivity are sufficient to find the set of top-k items (Cohen-Addad et al., 2020; Braverman et al., 2019). Under the more general SST model, it is known that $O(n)$ comparisons and $O(n)$ rounds are sufficient to find a PAC top-1 item (Falahatgar et al., 2017a,b), however, not much seems to be known for general k, even given unbounded rounds of adaptivity. We first show that (nk) comparisons are necessary for PAC top-k identification under SST even with unbounded adaptivity, establishing that this problem is strictly harder under SST than it is for the noisy comparison model. Our main contribution is to show that the 2-round query complexity for this problem is $(n^{4=3} + nk)$, and to show that just 3 rounds are sufficient to obtain a nearly optimal query complexity of (nk). We further show that our 3-round result can be improved by a $\log(n)$ factor using $2 \log n + 4$ rounds.

1 INTRODUCTION

The problem of finding the k best items amongst n totally ordered items using pairwise comparisons is a fundamental problem in ranking/sorting and has wide-ranging applications in a variety of domains including recommendation systems, sports, social choice, crowdsourcing etc. (Radlinski et al., 2008; Radlinski and Joachims, 2007; Baltrunas et al., 2010; Chen et al., 2013; Yue and Joachims, 2011; Soufiani et al., 2013). Due to the high cost of procuring comparison data, the natural objective is to minimize the number of pairwise queries required for finding top-k items.

However, in many practical applications such as recommendation systems, crowdsourcing etc., there can be a substantial delay between requesting comparisons and receiving feedback, thereby making it more efficient for an algorithm to query in parallel. Motivated by such applications, we consider an active/adaptive setting where the algorithm interacts with a feedback generating oracle in rounds, with each round consisting of comparisons for multiple pairs of items in parallel. Hence, the goal in our problem setting is to find the k best items while minimizing the number of pairwise queries with limited rounds of parallel interaction.

This problem is well-studied in both theoretical computer science and machine learning. The classical Selection algorithm finds top-k items using $O(n)$ noiseless comparisons in $O(\log n)$ rounds of interaction; Braverman et al. (2016) improved the number of rounds to 4 while having the same query complexity. The noisy comparison model (Feige et al., 1994), where there is a (fixed, constant) parameter $2\ [0; \frac{1}{2})$ such that the true outcome of a comparison is flipped with probability , has also been well-studied. Braverman et al. (2016) show that one can find the top-k set w.h.p. using $O(n \log n)$ noisy comparisons in 4 rounds, and Cohen-Addad et al. (2020) further improve the query complexity to $(n \log k)$ in $\log n$ rounds[1].

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151. Copyright 2022 by the author(s).

[1] The function $\log n$ is the number of times the logarithm

Table 1: Overview of Results

Adaptive Rounds	Upper Bound	Lower Bound
1	$O(n^2 \log n)$ [Trivial]	
2	$O(\max\{nk; n^{4=3}\} \log^2 n)$ [Theorem 2]	(n^2) [Braverman et al. (2019)]
3 2 log	$O(nk \log^2 n)$ [Theorem 3]	$(\max\{nk; n^{4=3}\})$ [Theorem 1+Alon and Azar (1988)]
n + 4	$O((nk + k^2 \log^2 k) \log k)$ [Corollary 1]	

(nk) [Theorem 1]

In this paper, we consider a more general noise model for pairwise comparisons, known as the strong stochastic sransitivity (SST) model. This model has roots in social science and psychology (Fishburn, 1973) and several empirical studies (Tversky, 1972; Ballinger and Wilcox, 1997) have indicated it to be effective at mod-eling real-world human decision-making, making it an active area of research in the machine learning commu-nity (Shah et al., 2016; Falahatgar et al., 2017a, 2018, 2017b). Given a set of $[n]$ items, the SST model is parameterized by a preference matrix P 2 $[0;1]^{nn}$ where P_{ij} is the probability of item i beating item j in a pairwise comparison. This model further as-sumes an underlying strict ordering over the items, and posits that for any items $h; i; j$ 2 $[n]$ ordered such that h i j, then P_{hj} $\max\{P_{hi}; P_{ij}\}$ [2]. This implies that the matrix P is consistent with the underlying ordering in the sense that P_{ij} $\frac{1}{2}$ if i j.

Since this model allows P_{ij} for any pair $(i;j)$ to be arbitrarily close to $\frac{1}{2}$, one would require an arbitrarily large number of comparisons to differentiate between such pairs $(i;j)$, making the separation of the exact top-k items from other items inefficient. We overcome this difficulty by adopting the probably approximately correct (PAC) paradigm, which has been commonly used in ranking literature (Busa-Fekete et al., 2014; Falahatgar et al., 2017a,b; Ren et al., 2020). Under this paradigm one can return an "approximately optimal" set of k items with high probability.

Definition 1 ((; k)-optimality and PAC top-k selection). For a set $[n]$, given $k <$ n, and 2 $(0;1]$, a subset S $[n]$ is said to be an $(;k)$-optimal subset[3] of $[n]$ if $jSj =$ k and for any items i 2 $S; j$ 2 $[n] n S$, $P_{ij} > 1=2$. Given 2 $(0;1]$, the $(;)$-PAC top-k identification[4] problem is to identify an $(;k)$-optimal subset of items w.p. 1 .

There have been several results showing that even under this more general SST model, PAC top-1 identification is possible using $O(n)$ comparisons and $O(n)$ rounds of interaction (Falahatgar et al., 2017a,b). Recently, Ren et al. (2020) showed that PAC top-k selection is possible using $(n \log k)$ comparisons and $(\log n)$ rounds if the model satisfies a stochastic triangle inequality (STI) condition in addition to SST [5]. However, the query com-plexity of PAC top-k selection under SST for general k without the STI assumption is not known, even when allowed unbounded adaptivity. Therefore, we seek to understand the following fundamental question –

Assuming the SST noise model, what is the query complexity of PAC top-k selection, and how many rounds of interaction/adaptivity are sufficient to achieve this?

1.1 Summary of Key Contributions (Table 1)

1. Lower Bound: We show (nk) comparisons are necessary for PAC top-k selection under SST even given unbounded adaptivity. One can observe a sharp contrast between the mild log k dependence un-der the noisy comparison model (as well as SST+STI model), and the linear dependence on k in the above bound under SST.

2. 2-Round Algorithm: We design a 2-round algorithm with a query complexity of $\tilde{O}(nk+n^{4=3})$, which is the best complexity achievable in 2 rounds (up to polylog factors), and shows that the additional cost incurred by limiting to 2 rounds is $\tilde{O}(n^{4=3})$.

3. 3-Round Algorithm: We design a 3-round algorithm with query complexity of $\tilde{O}(nk)$ which is optimal up to polylog factors. This shows, perhaps surprisingly, that even under this more general SST model, we need only a constant number of rounds for achieving optimal query complexity.

4. $(2 \log n + 4)$-round Algorithm: We show that it is possible to further improve the performance of the 3-round algorithm by log factors using more number of rounds. In particular, we design a $(2 \log n + 4)$ round algorithm that has a nearly optimal query complexity of $O(nk \log k)$ for any k $n= \log^2 n$.

function must be iteratively applied to n before the result is less than or equal to 1. This is a very slowly growing function and is less than 6 for most practical values of n.

[2]Note that the noisy comparison model discussed above satisfies this condition under SST, and hence, SST is a strictly more general model than noisy comparison model

[3]Note that the exact top-k set under the noisy comparison model is also a (; k)-optimal subset.

[4]For ease of exposition, we suppress dependence on ; .

[5]Note that the algorithms in Falahatgar et al. (2017a,b); Ren et al. (2020) were not optimized for rounds of interaction and turned out to be highly adaptive in nature.

1.2 Overview of Challenges and Key Ideas

We discuss the key technical challenges and algorithmic ideas used to overcome these challenges.

Lower bound for unbounded number of rounds. In order to understand the idea behind our lower bound one has to first observe that under the SST model, the probability of observing the correct preference relation between a pair of items upon comparing them can be item-dependent and arbitrary, unlike the noisy comparison model where the probability of observing the correct outcome for any pair is a fixed constant, say 2=3. For instance in our setting, for any given tolerance , it is possible to have a triple h i j, with $P_{hi}; P_{ij}$ = 1=2 + n ^{2}, and P_{hj} = 1=2 + , essentially implying that it is impossible to efficiently infer the sub-optimality of item j without explicitly comparing the pair (h; j). This idea forms the basis of our lower bound, which for any given tolerance , consists of an instance with two distinguished items 1; n with P_{1n} = 1=2 + , and for all other items i 2 [2; : : : ; n 1], P_{1i} = P_{in} = 1=2 + n ^{2}. For any k, the only invalid solution for this instance is a set which contains item n but excludes item 1. However, identifying these distinguished elements 1; n is only possible by comparing the specific pair (1; n). Now it is easy to see that any algorithm that succeeds with a sufficiently high probability must perform
(nk) comparisons. Intuitively, if the algorithm plans to return a set S of k items, it needs to have compared most pairs i 2 S; j 2 [n] n S. Otherwise, there is a possibility that the suboptimal item n was included, but item 1 was excluded, and the algorithm failed to detect this situation since the pair (1; n) was amongst the pairs that the algorithm did not compare. The following informal theorem describes our lower bound result, which is formally stated in Section 2.

Theorem 1 (Informal). *Given any set of items [n] with pairwise preferences satisfying SST, and any k n=2, any (;)-PAC top-k identification algorithm for a sufficiently small must perform (nk) comparisons, even when allowed unbounded adaptivity.*

In the above bound, we focus on establishing a sample complexity as a function of n and k. Though our lower bound construction does not reflect it, prior work (cf. Falahatgar et al. (2017a) and references therein) has established that a (worst-case) multiplicative dependence on the precision 1=^{2} is necessary – supposing we perform fewer comparisons per pair, then the observed relation can be erroneous with constant probability even for pairs that are -separated in pairwise preferences. We now discuss the key ideas behind our 2 and 3-round algorithms which are our main contributions.

The 2-round algorithm. At a high level, both our 2 and 3-round algorithms are based on the idea of pivoting – Suppose we aim for a query complexity of $\tilde{O}$(n) comparisons across 2 rounds, then we select anchor items, and compare them to all items (up to a desired precision) in parallel in the first round. The idea is that supposing we chose these anchors uniformly at random, then they should be roughly equally spaced in the true ordering – n= apart in expectation. Now suppose we could correctly determine the relative position of every item with respect to every anchor, then we can partition our items into chunks of size roughly n=. We can then process these anchors in their sorted order, adding entire chunks to our final solution without comparison until we reach the first chunk where adding it entirely would cause the solution size to exceed k. We can then focus our attention to just this chunk in the second round, where we compare all pairs of items in this chunk to the desired precision in parallel and add just the top items into our solution to meet our cardinality requirement. The query complexity would then be e O(n + (n=)^2), which after optimizing for would give us a sample complexity of $\tilde{\Theta}$(n$^{4=3}$) for = n$^{1=3}$.

This is indeed the idea used in the 2-round algorithm for top-k selection in the noisy comparison model (with some optimizations to save log factors), where it is possible to identify the relative position of every item with respect to every anchor by performing sufficiently many comparisons. However, this idea alone fails under the SST model where this neat partitioning of items into chunks of size O(n=) is no longer viable due to the arbitrary and item-dependent nature of the paired preference probabilities. To demonstrate this, consider the following instance: given any k, let > 0 be some arbitrarily small constant, with the set of items [n] being partitioned as follows: G_0 is the set of true top-k items, G_1 is a set of k^1 n$^{2=3}$ "equivalent" items, G_2 is a set of n$^{2=3}$ "not equivalent but indistinguishable" items, and G_3 are the rest of the "not equivalent but distinguishable" items. The groups are ordered as G_0 G_1 G_2 G_3, with an arbitrary internal ordering within each set. Let b = n ^{2} be a negligible bias term, then the pairwise preference probabilities[6] are:

$$P = \begin{pmatrix} \frac{1}{2} & \frac{1}{2}+b & \frac{1}{2}+ & \frac{1}{2}+ \\ & \frac{1}{2} & \frac{1}{2}+b & \frac{1}{2}+ \\ & & \frac{1}{2} & \frac{1}{2}+ \\ & & & \frac{1}{2} \end{pmatrix} \begin{matrix} G_0 \\ G_1 \\ G_2 \\ G_3 \end{matrix}$$

(columns: G_0, G_1, G_2, G_3)

Observe that any k items from G_0 [G_1 is an (; k)-

[6] Within each group, we can assume a negligible bias of b in the preference probabilities to ensure the strict ordering assumption required by SST. However, we ignore it here for ease of exposition.

optimal subset. However, if we sample just $O(n^{1=3})$ anchors, we almost certainly get no anchors from G_0 for $k < n^{2=3}$. We almost certainly also get no anchors from G_2 and just $O(k^1)$ anchors from set G_1. The rest of the anchors will be from set G_3. Now observe what happens when we compare all items to the anchors: since anchors from group G_1 are unable to differentiate between items from groups $G_0; G_1; G_2$, we end up with a set of $O(k^1 \; n^{2=3})$ items whose position can not be determined relative to the first $O(k^1)$ anchors. While we can construct a partial solution of size $O(k^1)$ using the anchors alone (since no item will beat any of these pivots from group G_1 with a margin larger than), finding an $(; k \quad k^1)$-optimal subset from the remaining chunk of size $(k^1 \; n^{2=3})$ in just the one remaining round is a non-trivial challenge. We cannot simply take an arbitrary set of (k) items from this large chunk due to the existence of the set G_2, and the naive recipe of comparing all pairs of items in this chunk would require $(k^2 \; {}^2 n^{4=3})$ comparisons, which is larger than our desired query complexity of $O(n^{4=3} + nk)$ for any polynomially large k. There-fore, the pivoting idea which worked well for the noisy comparison model, fails here.

Despite this apparent difficulty, we show that a pair of new ideas combined with pivoting actually gives us a 2-round algorithm with a query complexity of $\Theta(n^{4=3} + nk)$ (optimal up to log factors), and the above hard instance proves helpful in developing intuition.

The first key idea addresses the problem of the margins in pairwise probabilities being arbitrarily small: we first compare all items to the anchors to a precision slightly smaller than the allowed tolerance , say =4, and for every anchor, construct a winner set of items that beat the anchor with a large observed margin, say larger than 3=4. The idea here is that the winner set of an anchor can only contain items that precede the anchor in the true sorted order, and necessarily contains every item that beats the anchor with a true margin larger than , giving these items priority over the anchor. The items that do not make it into the winner set can be treated as equivalent if not worse than the anchor. We then process the anchors in their sorted order, breaking ties by preferring anchors with smaller winner sets when the ordering is unclear, adding their entire winner sets into the partial solution stopping at the first anchor whose winner set, if added to the partial solution, would cause its size to exceed k. If this final winner set is sufficiently small in size to perform all pairwise comparisons, we do so and pick the best items from this final winner set to meet the cardinality requirement of k. However, as demonstrated by the hard instance outlined above, this by itself is still not sufficient to control the sizes of these winner sets, with the last winner set being possibly as large as $(kn^{2=3})$ in size. This brings us to the second key idea, which is the way we handle the second round of queries.

Observe in the above instance, we ran into difficulty because the final winner set we were left with was too large to perform all pairwise comparisons. Moreover, although all items in this set had an essentially identical profile when compared against all anchors, there existed a problematic set of "not equivalent but indistinguishable" items (G_2) hidden in this chunk which prevented us from picking arbitrary items to meet our cardinality requirement of k. However, observe that these suboptimal items are very small in number, and we can in fact argue that this must always be the case. Specifically, there can be at most $O(n^{2=3})$ such items because otherwise, we would have sampled an anchor from this set allowing us to isolate the "true" top-k set exactly. Since we broke ties by preferring anchors with smaller winner sets, this would have guaranteed the final winner set to have size at most O(k), which is within our query budget to simply perform all pair-wise comparisons in this final winner set. The formal argument is much more nuanced, but generally builds upon this intuition. Therefore, this naturally suggests a random sampling idea for the second round - if the remaining chunk is too large in size, say larger than $10 \max\{n^{2=3}; \sqrt{nk}\}$, we can sample say 10k items from this chunk and compare them against all other items in this chunk, costing only O(nk) comparisons in the worst case. By a standard Chernoff bound, we would be guaranteed to sample a $(; k)$-optimal subset into this set, which would be easily identifiable from noisy pairwise comparisons as they would be items that do not lose to anyone with an observed margin larger than 3=4. We can then cover the deficit in our cardinality requirement with an arbitrary set of such items that are not beaten by a large observed margin.

The following informal theorem describes our 2-round upper bound, which is formally stated in Section 3.1.

Theorem 2 (Informal). Given any items [n] with unknown pairwise preferences satisfying SST, any k n=2, > 0, and > 0, there exists a 2-round algorithm for $(;)$-PAC top-k selection with query complexity $O(n^{4=3} + nk)$.

Note that this bound is tight (upto log factors) as Braverman et al. (2019) show that any 2-round algorithm needs $(n^{4=3})$ comparisons, and our lower bound shows that (nk) is always necessary.

The 3-round algorithm. The 3-round algorithm is similar to the 2-round algorithm, except for a few key differences. We begin by sampling a larger set of $\Theta(\sqrt{n})$ anchors uniformly at random, and invest the first round into comparing all anchors amongst themselves to a precision slightly smaller than the specified tolerance , say =4. We retain just the best k anchors, which are anchors that do not lose to any of the discarded anchors by an observed margin larger than this precision. The rest of the algorithm is now identical to the 2-round algorithm, with the set of anchors being these k best anchors chosen at the end of the first round. The intuition behind this approach is based on two observations: firstly, supposing we could actually identify the best k anchors from the set of the randomly sampled $O(\sqrt{n})$ anchors, then an (; k)-optimal subset of items can be found amongst these best k anchors and the items that are superior (specifically, in their winner sets) to these k anchors alone. Therefore, the items that are inferior to these best k anchors can be safely ignored, making comparisons with the rest of the inferior anchors meaningless. Secondly, supposing we make a mistake in identifying the actual best k an-chors, then any inferior anchor that was chosen in place of an actual best-k anchor could not have been much worse. Specifically, the actual best-k anchor could not have beaten this inferior anchor by a true margin larger than =2 (twice the set precision) and consequently, due to SST, no item that lies between the best-k anchor and the chosen inferior anchor can beat the inferior anchor by a margin larger than this. Therefore, none of these items can be included into the winner set of this inferior anchor as none of these items can beat the inferior anchor with an observed margin larger than 3=4. Therefore, the winner set of this inferior anchor can only contain items that precede the actual best-k anchor in the true sorted ordering, effectively simu-lating selecting the actual best-k anchor itself. The formal proof is subtle, but builds upon this idea. The following theorem describes our 3-round upper bound, which is formally stated and proved in Section 3.2.

Theorem 3 (Informal). In the setting of Theorem 2, there exists a 3-round algorithm for (;)-PAC top-k selection with query complexity $\tilde{O}(nk)$.

(2 log n + 4)-round algorithm. Our algorithm is based on the idea of selecting good anchors and comparing other items to these anchors in order to find a small number of top-k candidates. This idea is similar to the top-k algorithm of Cohen-Addad et al. (2020) which finds an anchor with rank O(k) and filters items that are better than this anchor in successive rounds. However, as discussed in the overview of 2 and 3-round algorithms, our setting does not allow one to filter items based on such precise criterion as the margins in pairwise preferences can be arbitrarily small. More precisely, consider the example given in the overview of 2-round algorithm where there are 4 groups– G_0 to G_3. If one happens to select an anchor from the group G_1 then one will not be able to filter G_0 as it is very close to G_1 (underflow), and if the anchor lies in group G_3 then all of $G_0; G_1; G_2$ can be filtered which can be much larger than required (overflow).

In order to solve the overflow problem we define the notion of the -rank of an item, which is the number of items that beat the given item with margin at least . We then find an anchor with =3-rank of O(k) using a top-1 algorithm similar to Cohen-Addad et al. (2020) as a subroutine over a randomly chosen subset of items. Now, we compare items up to a precision of =6 and filter all items which beat the anchor with margin at least 2=3, and exclude all items which beat the anchor with margin less than =3 (including items that lose to the anchor). This solves the overflow problem as there are at most $\Theta(k)$ items that can be filtered in. However, it is possible that no items beat the anchor with a margin at least 2=3, still leaving us with the underflow problem. In this case, observe that the anchor itself is an -optimal item and can be included in the solution. Hence, if we repeat this in parallel for k different good anchors, we will have k items in the solution even if we have underflow for all of them. These ideas combined with an aggressive item-elimination strategy (Agarwal et al., 2017) gives us the following result.

Corollary 1 (Informal). In the setting of Theorem 2, there exists a (2 log n + 4)-round algorithm for (;)-PAC top-k selection with query complexity $O((nk + k^2 \log k) \log k)$.

1.3 Related Work

There is a substantial literature on the problem of top-k identification from pairwise comparisons in theoretical computer science and machine learning. Given a set of n items with an underlying ranking over them, the classical Selection algorithm finds the set of top-k items using O(n) noiseless comparisons and O(log n) rounds. Bollobás and Brightwell (1990) show that O(n) noiseless comparisons and 4 rounds are sufficient for a closely related problem of finding the k-th ranked item, and Braverman et al. (2016) show that one can even solve top-k identification with same number of noiseless comparisons and rounds.

The noisy comparison model was introduced by Feige et al. (1994), who showed that the top-1 item can be identified using O(n) comparisons and log n rounds. Braverman et al. (2016) show that one can find the set of top-k items under this comparison model using 4 rounds and O(n log n) comparisons. Braverman et al. (2019) further improve this understanding by showing that a 1-round algorithm needs (n^2) comparisons, while a 2-round algorithm needs $(n^{4=3})$ comparisons. Finally, Cohen-Addad et al. (2020) show that the optimal query and round complexity under this model is (n log k) comparisons and (log n) rounds, respectively. However, these results are for the noisy comparison model, which is considerably more restrictive than the SST model we consider.

Another line of work considers the top-k identification problem under parametric models such as the Bradley-Terry-Luce (BTL) model (Luce, 1959; Bradley and Terry, 1952). In particular, Szörényi et al. (2015) show that one can find a PAC top-1 item using O(n log n) comparisons. Chen and Suh (2015); Chen et al. (2017, 2019) show that one can find the exact top-k set using O(npoly(log n)) comparisons given the knowledge of a gap parameter. However, these results are in the passive setting where the algorithm has no control over which comparisons are performed. Moreover, these models have been shown to be much more restrictive than the SST model that we consider in this paper (Tversky, 1972; Ballinger and Wilcox, 1997).

Ranking under the SST model has been an active area of research in machine learning; here we focus on results directly related to top-k identification. Yue and Joachims (2011) show that one can find a PAC top-1 item using O(n log n) comparisons assuming that the comparison model also satisfies stochastic triangle inequality (STI) in addition to SST. Under the same assumption, Falahatgar et al. (2017a) further improve the query complexity for PAC top-1 identification to O(n). Finally, Ren et al. (2020) show that one can find a PAC top-k set using (n log k) comparisons and O(log n) rounds under SST+STI. However, this result crucially uses the STI condition which does not apply in our setting. Falahatgar et al. (2017b) relax the STI assumption, and show that one can find a PAC top-1 item under SST alone using O(n) comparisons. In a follow-up work, Falahatgar et al. (2018) show that the same query complexity holds for a slightly more general stochastic transitivity condition called MST. However, these results focussed on the special case of k = 1 and their algorithms were designed for unbounded rounds of interaction. Mohajer et al. (2017) consider the problem of finding exact top-k items under a more general model than SST, but assumes that the gap between k-th and (k + 1)-th ranked items is fixed and known, which is crucially used by their algorithm. In contrast, we make no assumptions on the gaps between items.

There has also been work on best item identification under more general, non-transitive models. However, the best item under these models is generally not well-defined and one has to resort to other notions of best item such as the Borda or Copeland winner (de Borda, 1781; Agarwal et al., 2017; Busa-Fekete et al., 2014; Busa-Fekete et al., 2013; Shah and Wainwright, 2015; Heckel et al., 2019). The most closely related work to ours is Agarwal et al. (2017) which shows that one can find the Borda winner for any pairwise probability model using $(n/^2_b)$ comparisons and (log n) rounds of querying, where $_b$ is the gap between Borda scores of the k-th and (k + 1)-th items. Under SST, the ordering with respect to Borda scores happens to be consistent with the true ordering, hence, one can use their algorithm for exact top-k identification under SST. However, the gap $_b$ between Borda scores can be (n) times smaller than the actual preference gap between the k-th and the (k + 1)-th items in our setting. Hence, the query complexity of their algorithm can have a (n^3) dependence which is much worse compared to our results. Moreover, their algorithm does not apply to the PAC setting and requires the knowledge of $_b$.

There is also a vast literature on recovering a full ranking over items using pairwise comparisons under SST and other models; we refer the reader to surveys provided in Agarwal (2016); Bengs et al. (2021). Shah et al. (2016) also consider the problem of estimating the entire pairwise preference matrix under SST. However, these results are tangential to the results in our paper as estimation of the preference matrix does not necessarily translate to identification of the top-k items.

Best-arm identification under the dueling bandits framework has also gained significant attention in recent years (Bengs et al., 2021). However, this framework only focusses on top-1 identification, whereas the focus of our work is the more general top-k identification problem. Moreover, we are not aware of any work on dueling bandits that considers limited adaptivity. Top-k identification under the multi-armed bandits setting has also been widely studied (Even-Dar et al., 2006; Kalyanakrishnan et al., 2012; Kalyanakrishnan and Stone, 2010), however, the algorithms here receive quantitative feedback on the quality of an item whereas in our setting the algorithm receives relative feedback between two items. The design of algorithms with limited adaptivity has also been an active area area in machine learning, and algorithms with limited rounds of adaptivity have been designed for various problems (Agarwal et al., 2017; Braverman et al., 2016, 2019; Zhang et al., 2020; Ruan et al., 2021).

2 LOWER BOUND FOR UNBOUNDED ADAPTIVITY

In this section we formally state the lower bound on the query complexity of (;)-PAC top-k identification under the SST model for paired comparisons.

Theorem 1. For any n and any k n=2,[7], there exist pairwise preferences over n items satisfying the SST

[7]The assumption k n=2 is without loss of generality, because otherwise, we can equivalently identify the bottom-(n k) items instead.

condition such that any algorithm for (;)-PAC top-k identification needs to perform at least $nk=4$ pairwise comparisons for k=(8(n 1)).

The hardness of this more general pairwise comparison model becomes apparent when we contrast this query-complexity lower bound with existing results for top-k identification in the noisy comparison model (Cohen-Addad et al., 2020) (and even SST+STI model (Ren et al., 2020)). Specifically, for k =
(n), in the noisy comparison model (and SST-STI model), O(n log n) comparisons are sufficient to solve this problem with high 1 1=poly(n) probability, whereas under this more general SST model, our lower bound shows that any algorithm that solves this problem with even con-stant probability given unbounded rounds of adaptivity, must necessarily perform
(n^2) comparisons. In the following sections, we design algorithms for top-k iden-tification under this more general comparison model, given limited rounds of adaptivity.

3 CONSTANT-ROUND ALGORITHMS FOR (;)-PAC TOP-k IDENTIFICATION

In this section, we present our constant round algorithms for PAC top-k identification under the SST model. Note that it is easy to design a 1-round algorithm with a query complexity of $\Theta(n^2)$ by comparing all items to each other sufficient number of times and identifying a set of top-k items based on realized preference probabilities. A standard Hoeffding's inequality will show that this algorithm will succeed with high probability. Braverman et al. (2019) also gives a lower bound showing that any 1-round algorithm needs to have a query complexity of
(n^2). This easily resolves the case of 1-round (upto log factors). Hence, our main focus here is on 2 and 3-round algorithms.

3.1 A 2-Round Algorithm

We begin by presenting an algorithm for PAC top-k identification with $\Theta(\max fn^{4=3}; nkg)$ query complexity using 2 rounds of adaptivity. The following theorem characterizes the 2-round upper bound.

Theorem 2. Given any set of items [n] with unknown pairwise preferences P 2 $[0; 1]^{nn}$ satisfying SST, any integer k 2 [1; n=2], tolerance 2 (0; 1], and confidence 2 (0; 1], there exists an algorithm for (;)-PAC top-k identification with query complexity $O((1=^2) \max fn^{4=3}; nkg \log^2(n=))$ and at most 2 rounds.

Note that the above bound is tight (upto log factors) as Braverman et al. (2019) show that one needs $(n^{4=3})$ comparisons for identifying top-k in 2-rounds, and our lower bound in Section 2 shows that (nk) is neces-sary. The proof of the above theorem is given in the supplementary material.

We present here an overview of the algorithm, which is formally specified as Algorithm 1. Our algorithm begins by sampling a set A of "anchor items" chosen at random. These anchors, which are roughly $\max fn^{1=3}; kg \log n$ in number, are subsequently compared to all items (including other anchors) up to a precision of =4 in parallel. The observed pairwise preference probabilities are then used to construct "winner sets" W_a for every anchor a 2 A, which consist of items that beat anchor a with an observed margin of at least 3=4. The idea behind these winner sets W_a is that they only consist of items that are better ranked than the anchor a, and necessarily contain every item that is -better than anchor a. We then order the anchors using both the observed preference probabilities between anchors, as well as the size of these winner sets: if an anchor i 2 A beats another anchor j 2 A with an observed margin of at least =4, or if the observed margin is strictly smaller than =4 but i had a smaller winner set $W_i < W_j$, then i precedes j in the ordering. Otherwise, ties are broken arbitrarily. Due to the precision with which pairs are compared, the former condition is a case where anchor i necessarily precedes anchor j in the true permutation, whereas the latter condition is a case where it is not possible to identify the relative ordering of anchors i and j in the true permutation in which case the anchor with the smaller winner set is preferred. We refer to this sorted ordering of anchors as $a_1; : : : ; a_{jAj}$.

The algorithm then processes these anchors in the sorted order $a_1; : : : ; a_{jAj}$, greedily constructing an (; jT j)-optimal partial solution T^0 by including entire winner sets W_a without any additional comparisons, halting either

1. When k anchors have been processed without the cardinality of T^0 exceeding k, i.e. t = k + 1, and $j [_{i k} W_{a_i} j < k$.
2. At the first anchor a_t where including its entire winner set W_{a_t} in T^0 would cause its cardinality to exceed k, i.e. t k : $j [_{i<t} W_{a_i} j <$ k; and $j [_{i t} W_{a_i} j$ k.

Let k^0 = jT^0j be the number of items in the partial solution constructed at this point. The remaining budget of k k^0 items is filled depending on the halting condition.

In the first case, the remaining budget of k k^0 items is filled by including k k^0 anchors chosen arbitrarily from amongst the first k anchors that have not already been included in the partial solution T^0 thus far. Since $k^0 <$ k, observe that we can always find such a set of anchor items.

In the second case, we do one of two things depending on the number of "candidate items" jW_{a_t} nT^0j to choose from. If this number is small enough that we can afford to perform all pairwise comparisons without exceeding our query budget of Θ(maxfn$^{4=3}$; nkg), we do so and select an (; k k^0)-optimal subset C of W_{a_t} n T^0 (ties broken arbitrarily) to include in our partial solution T^0 to cover the deficit. Specifically, C W_{a_t} n T^0 is a set of k k^0 items such that for any item i 2 C, and any item j 2 (W_{a_t} n T^0) n C, $\hat{P}_{ji}$ < 1=2 + 3=4. On the other hand, if the number jW_{a_t} n T^0j of candidate items is too large, then we first try to fill the remaining budget using items that are guaranteed to have rank at most that of any of the previously parsed pivots, i.e. items j 2 [n] n T that beat any of the previously parsed pivots with margin at least =4. We refer to these items as S, specifically S = fj 2 [n] n T^0 : $\hat{P}_{ja_h}$ 1=2 + =4 for some pivot a_h; where h < tg. If this set is large enough to cover our remaining budget, then we select any arbitrary k k^0 items from this set to include in our partial solution T^0. If not, we extend our partial solution by including all such elements, i.e. T^{00} = T^0 [S, and we refer to the size of this new partial solution as jT^{00}j = k^{00}. We sample a smaller set C_{a_t} of 10k log(1=) candidate items chosen at random from all candidate items and compare every sampled candidate item i 2 C_{a_t} to every item j 2 W_{a_t} n T^{00}. Finally, we select a set C C_{a_t} of k k^{00} items (ties broken arbitrarily) that do not lose to any other item with a margin larger than 3=4 to include in our partial solution T^0, filling the remaining budget. Specifically, C C_{a_t} is a set of k k^{00} items such that for any item i 2 C, and any item j 2 (W_{a_t} n T^{00}) n C, $\hat{P}_{ji}$ < 1=2+3=4.

Algorithm 1 A 2-round algorithm for (;) PAC top-k

Input: items [n], parameter k, accuracy , confidence
Let A ;; q 4 maxf$\sqrt{nk}$; $n^{2=3}$g.
For each element i 2 [n], add i to set A with probability 2 log(9n=)= minfn$^{2=3}$; n=kg
Output: 2-Round-Select([n]; A; k; ; ; q)

3.2 A 3-Round Algorithm

We now present an algorithm for PAC top-k selection that has query complexity Θ(nk) using 3 rounds of adaptivity. The following theorem describes the 3-round upper bound.

Theorem 3. Given any set of items [n] with unknown pairwise preferences P 2 [0; 1]^{nn} satisfying SST, any integer k 2 [1; n=2], tolerance 2 (0; 1], and confidence

Algorithm 2 2-Round-Select(X; A; k; ; ; q)

Input: set of items X, set of anchors A X, cardi-nality k, accuracy , confidence , maximum set size q.
Let n = jXj, ^{0} = =4, and m (1=^{02}) log(3n=).
Round 1 (in parallel): For every element a 2 A, compare a against every item i 2 X, with each comparison repeated m times. Let $\hat{P}_{ia}$ be the observed probability of i beating a.
For every a 2 A, let W_a fi 2 S : $\hat{P}_{ia}$ 1=2+3=4g be the set of elements in X that beat a with margin at least 3=4.
Sort A using the following rule: for any pair i; j 2 A : $\hat{P}_{ij}$ 1=2 + =4, or 1=2 =4 < $\hat{P}_{ij}$ < 1=2 + =4 and jW_ij < jW_jj, then i precedes j in the ordering. Else, break ties arbitrarily. Let a_1; a_2; : : : ; a_{jAj} be the corresponding sorted order.
Let T^0 ;, R ;, and i 1
while jT^0 [W_{a_i}j < k and i k do
  T^0 T^0 [W_{a_i}, R R [fa_ig, i i + 1.
end while
Let t i, and let k^0 jT^0j.
if Case 1: t = k + 1 then
  T T^0 [R_a, where R_a R n T^0, jR_aj = k k^0 chosen arbitrarily.
else Case 2: t k
  if Case 2a: jW_{a_t} n T^0j q then
    Round 2 (in parallel): Perform all pairwise comparisons between items in W_{a_t} n T^0 with each comparison repeated m times.
    Let T T^0 [C, where C W_{a_t} n T^0 is any arbitrary set of (k k^0) items such that for any item i 2 C, and any item j 2 (W_{a_t} n T^0) n C, $\hat{P}_{ji}$ < 1=2 + 3=4.
  else Case 2b: jW_{a_t} n T^0j > q
    For all 1 h < t, let S_h fj 2 X n T^0 : $\hat{P}_{ja_h}$ 1=2 + =4g be the set of items outside our partial solution T^0 that beat anchor a_h with margin at least =4. Let S $[_{h<t} S_h$.
    if jSj k k^0 then
      T T^0 [S_a, where S_a S, jS_aj = k k^0 chosen arbitrarily.
    else
      T^{00} T^0 [S, and let k^{00} jT^{00}j
      Round 2 (in parallel): Sample a set C_{a_t} of 6k log(3=) items uniformly at random from W_{a_t} n T^{00}, and compare every pair i; j : i 2 C_{a_t}; j 2 W_{a_t} n T^{00} with each comparison repeated m times.
      Let T T^{00} [C, where C C_{a_t} a set of k k^{00} elements such that for any element i 2 C and any other item j 2 W_{a_t} n T^{00}, $\hat{P}_{ji}$ < 1=2 + 3=4 (ties broken arbitrarily).
    end if
  end if
end if
Output: T, an (; k)-optimal subset of items

2 (0; 1], there exists an algorithm for (;)-PAC top-k selection with query complexity $O((1=^2)nk \log^2(n=))$ comparisons and at most 3 rounds of adaptivity.

The proof of the above theorem is given in the supplementary material. Our 3-round algorithm, which is formally presented as Algorithm 3, is a natural extension of the 2-round algorithm from the previous section, with the key difference being the way it utilizes this extra round of querying. As one might expect, we begin by sampling a set A of anchor items chosen at random. In the 3-round algorithm, we sample a lot more anchors than the 2-round algorithm, roughly maxf^p n; kg log n in number, and use the first round to compare all pairs of anchors in parallel up to a precision of =4. We use the outcomes of these comparisons to prune the set of anchors, retaining a set A_k of just the top-k anchors, i.e. A_k is a set of k anchors such that there is no item amongst the remaining anchors A n A_k that beats any anchor in A_k with an observed margin of at least =4. The rest of the algorithm then proceeds identically to the 2-round algorithm, using A_k as the effective set of anchors with a smaller query budget of Qe(nk) in Case 2a, 2b in the next 2 rounds.

Algorithm 3 A 3-round algorithm for (;) PAC top-k

Input: items [n], parameter k, accuracy , confidence

Let A ; ; q 4 ^{p} nk.

For each element i 2 [n], add i to set A with probability 2 log(9n=)= minf ^{p} n; n=kg.

Round 0 (in parallel): For every pair of anchors i; j 2 A, compare i against j with each comparison repeated $(1=(=4)^2) \log(9n=)$ times, and let $\hat{P}_{ij}$ be the observed probability of i beating j.

Let A_k A be a set of k anchors such that for any anchor i 2 A_k, and any anchor j 2 A n A_k, $\hat{P}_{ji}$ < 1=2 + =4 (ties broken arbitrarily).

Output: 2-Round-Select([n]; A_k; k; ; ; q)

4 A PARAMETERIZED ALGORITHM FOR (;)-PAC TOP-k SELECTION

In this section, we further improve the query complexity of our 3-round algorithm by log factors using few additional rounds. We achieve this by designing a parameterized algorithm whose query complexity and adaptivity scales as a function of an input round parameter r as described in the following theorem. Due to space constraints, the details of this result are deferred to the supplementary material.

Theorem 4. Given any items [n] with unknown pairwise preferences P 2 $[0; 1]^{nn}$ satisfying SST, an integer parameter k 2 [1; n=2], tolerance 2 (0; 1], and confidence 2 (0; 1], there exists an algorithm for (;)-PAC top-k selection with query complexity O $(1=^2)(nk(\log^{(r)}(n)+\log(k=))+k^2 \log^2 k \log(k=))$ at most (2r + 4) rounds of adaptivity for any integer parameter r, where $\log^{(r)}(a)$ denotes the iterated logarithms of order r, i.e. $\log^{(r)}(a)$ = max log $\log^{(r-1)}(a)$; 1 and $\log^{(0)}(a) = a$.

The above theorem establishes a non-trivial upper bound on the tradeoff between query complexity and round complexity for PAC top-k identification under SST. In order to simplify the exposition, let k = $O(n=\log^2 n)$. Then, by setting r = 1 we get a 5-round algorithm with query complexity O nk log n log k , and by setting r = log n we achieve the best complexity of O nk log k . Note that the latter bound is away from the lower bound of (nk) (Section 2) by only a log k factor.

Corollary 1. In the setting of Theorem 4, there exists an algorithm for (;)-PAC top-k identification with query complexity O $(1=^2)(nk+k^2 \log^2 k) \log(k=)$ and at most (2 log(n) + 4) rounds of adaptivity.

5 CONCLUSION

We studied the problem of identifying a PAC top-k solution under the SST noise model for pairwise comparisons given limited rounds of parallel interaction (adaptivity) with the comparison oracle. We established a query complexity lower bound of (nk) comparisons, even given unbounded rounds of adaptivity. This lower bound sharply contrasts with the known (n log k) query complexity results for both, the noisy comparison model and the SST+STI model, which are special cases of SST. We further complemented this lower bound with new algorithmic results for this setting. Specifically, we designed a 2-round algorithm with a tight query complexity of Oe$(nk + n^{4=3})$, and a 3-round algorithm that achieved a nearly optimal query complexity Oe(nk). In addition to these specific constant-round algorithms, we also designed a parameterized algorithm which achieves an improved query complexity of $O(nk(\log^{(r)}(n) + \log k) + k^2 \log^3 k)$ in 2r + 4 adaptive rounds for any input parameter r. This final result is interesting in its own right, as it establishes a non-trivial upper bound on the tradeoff between query complexity and round complexity.

In the future, it would be interesting to understand if we can bridge the polylogarithmic gap in our lower and upper bounds for 2 and 3 rounds. Also, is it possible to have low adaptive and sample efficient algorithms for top-k identification for even more general pairwise comparison models, such as medium stochastic transitivity (MST), or is SST the weakest model under which such a result is possible?

Acknowledgement

This work was supported in part by NSF awards CCF-1934876 and CCF-2008304.

References

Agarwal, A., Agarwal, S., Assadi, S., and Khanna, S. (2017). Learning with limited rounds of adaptivity: Coin tossing, multi-armed bandits, and ranking from pairwise comparisons. In Kale, S. and Shamir, O., editors, Proceedings of the 30th Conference on Learning Theory, COLT 2017, Amsterdam, The Netherlands, 7-10 July 2017, volume 65 of Proceedings of Machine Learning Research, pages 39–75. PMLR.

Agarwal, S. (2016). On ranking and choice models. In Kambhampati, S., editor, Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016, pages 4050–4053. IJCAI/AAAI Press.

Alon, N. and Azar, Y. (1988). Sorting, approximate sorting, and searching in rounds. SIAM Journal on Discrete Mathematics, 1(3):269–280.

Ballinger, T. P. and Wilcox, N. T. (1997). Decisions, error and heterogeneity. The Economic Journal, 107(443):1090–1105.

Baltrunas, L., Makcinskas, T., and Ricci, F. (2010). Group recommendations with rank aggregation and collaborative filtering. In Amatriain, X., Torrens, M., Resnick, P., and Zanker, M., editors, Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010, pages 119–126. ACM.

Bengs, V., Busa-Fekete, R., Mesaoudi-Paul, A. E., and Hüllermeier, E. (2021). Preference-based online learning with dueling bandits: A survey. J. Mach. Learn. Res., 22:7:1–7:108.

Bollobás, B. and Brightwell, G. (1990). Parallel selection with high probability. SIAM Journal on Discrete Mathematics, 3(1):21–31.

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.

Braverman, M., Mao, J., and Peres, Y. (2019). Sorted top-k in rounds. In Beygelzimer, A. and Hsu, D., editors, Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA, volume 99 of Proceedings of Machine Learning Research, pages 342–382. PMLR.

Braverman, M., Mao, J., and Weinberg, S. M. (2016). Parallel Algorithms for Select and Partition with Noisy Comparisons. In STOC.

Busa-Fekete, R., Szorenyi, B., Cheng, W., Weng, P., and Hullermeier, E. (2013). Top-k selection based on adaptive sampling of noisy preferences. In ICML.

Busa-Fekete, R., Szörényi, B., and Hüllermeier, E. (2014). PAC rank elicitation through adaptive sampling of stochastic pairwise preferences. In Brodley, C. E. and Stone, P., editors, Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada, pages 1701–1707. AAAI Press.

Cesa-Bianchi, N. and Lugosi, G. (2006). Prediction, learning, and games. Cambridge university press.

Chen, L., Li, J., and Qiao, M. (2017). Nearly Instance Optimal Sample Complexity Bounds for Top-k Arm Selection. arXiv preprint arXiv:1702.03605.

Chen, X., Bennett, P. N., Collins-Thompson, K., and Horvitz, E. (2013). Pairwise ranking aggregation in a crowdsourced setting. In Leonardi, S., Panconesi, A., Ferragina, P., and Gionis, A., editors, Sixth ACM International Conference on Web Search and Data Mining, WSDM 2013, Rome, Italy, February 4-8, 2013, pages 193–202. ACM.

Chen, Y., Fan, J., Ma, C., and Wang, K. (2019). Spectral method and regularized mle are both optimal for top-k ranking. Annals of statistics, 47(4):2204.

Chen, Y. and Suh, C. (2015). Spectral MLE: Top-k rank aggregation from pairwise comparisons. In ICML.

Cohen-Addad, V., Mallmann-Trenn, F., and Mathieu, C. (2020). Instance-optimality in the noisy value-and comparison-model. In Chawla, S., editor, Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020, pages 2124–2143. SIAM.

de Borda, J. C. (1781). Mémoire sur les élections au scrutin.

Even-Dar, E., Mannor, S., and Mansour, Y. (2006). Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. Journal of Machine Learning Research, 7:1079–1105.

Falahatgar, M., Hao, Y., Orlitsky, A., Pichapati, V., and Ravindrakumar, V. (2017a). Maxing and ranking with few assumptions. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R., editors, Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pages 7060–7070.

Falahatgar, M., Jain, A., Orlitsky, A., Pichapati, V., and Ravindrakumar, V. (2018). The limits of maxing, ranking, and preference learning. In Dy, J. G.

and Krause, A., editors, Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, volume 80 of Proceedings of Machine Learning Research, pages 1426–1435. PMLR.

Falahatgar, M., Orlitsky, A., Pichapati, V., and Suresh, A. T. (2017b). Maximum selection and ranking under noisy comparisons. In Precup, D. and Teh, Y. W., editors, Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, volume 70 of Proceedings of Machine Learning Research, pages 1088–1096. PMLR.

Feige, U., Raghavan, P., Peleg, D., and Upfal, E. (1994). Computing with Noisy Information. SIAM Journal on Computing, 23(5):1001–1018.

Fishburn, P. C. (1973). Binary choice probabilities: on the varieties of stochastic transitivity. Journal of Mathematical psychology, 10(4):327–352.

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

Kalyanakrishnan, S. and Stone, P. (2010). Efficient Selection of Multiple Bandit Arms: Theory and Practice. In ICML.

Kalyanakrishnan, S., Tewari, A., Auer, P., and Stone, P. (2012). PAC Subset Selection in Stochastic Multi-armed Bandits. In ICML.

Luce, R. D. R. D. (1959). Individual choice behavior : a theoretical analysis. Wiley.

Mohajer, S., Suh, C., and Elmahdy, A. M. (2017). Active learning for top-k rank aggregation from noisy comparisons. In Precup, D. and Teh, Y. W., editors, Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, volume 70 of Proceedings of Machine Learning Research, pages 2488–2497. PMLR.

Radlinski, F. and Joachims, T. (2007). Active exploration for learning rankings from clickthrough data. In Berkhin, P., Caruana, R., and Wu, X., editors, Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Min-ing, San Jose, California, USA, August 12-15, 2007, pages 570–579. ACM.

Radlinski, F., Kurup, M., and Joachims, T. (2008). How does clickthrough data reflect retrieval quality? In Shanahan, J. G., Amer-Yahia, S., Manolescu, I., Zhang, Y., Evans, D. A., Kolcz, A., Choi, K., and Chowdhury, A., editors, Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008, pages 43–52. ACM.

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

Ruan, Y., Yang, J., and Zhou, Y. (2021). Linear bandits with limited adaptivity and learning distributional optimal design. In Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing, pages 74–87.

Shah, N. B., Balakrishnan, S., Guntuboyina, A., and Wainwright, M. J. (2016). Stochastically transitive models for pairwise comparisons: Statistical and computational issues. In Balcan, M. and Weinberger, K. Q., editors, Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016, volume 48 of JMLR Workshop and Conference Proceedings, pages 11–20. JMLR.org.

Shah, N. B. and Wainwright, M. J. (2015). Simple, Robust and Optimal Ranking from Pairwise Comparisons. arXiv preprint arXiv:1512.08949.

Soufiani, H. A., Chen, W. Z., Parkes, D. C., and Xia, L. (2013). Generalized method-of-moments for rank aggregation. In Burges, C. J. C., Bottou, L., Ghahramani, Z., and Weinberger, K. Q., editors, Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, pages 2706–2714.

Szörényi, B., Busa-Fekete, R., Paul, A., and Hüllermeier, E. (2015). Online rank elicitation for plackett-luce: A dueling bandits approach. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R., editors, Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, De-cember 7-12, 2015, Montreal, Quebec, Canada, pages 604–612.

Tversky, A. (1972). Elimination by aspects: A theory of choice. Psychological review, 79(4):281.

Yue, Y. and Joachims, T. (2011). Beat the mean bandit. In Getoor, L. and Scheffer, T., editors, Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011, pages 241–248. Omnipress.

Zhang, K. W., Janson, L., and Murphy, S. A. (2020). Inference for batched bandits. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.

Supplementary Material:
PAC Top-k Identification under SST in Limited Rounds

A Concentration Inequalities

In this section, we record all of the concentration inequalities used in our proofs. These are all well known inequalities; see Cesa-Bianchi and Lugosi (2006) for example.

Theorem 5 (Multiplicative Chernoff Bounds). Let $X_1, \ldots, X_n$ be independent Bernoulli random variables, with X denoting their sum, and $E(X) = \mu$ denoting their mean. Then for any $0 < \epsilon < 1$, we have that

$$\Pr(|X - \mu| \geq \epsilon\mu) \leq 2\exp\left(-\frac{\epsilon^2 \mu}{3}\right)$$

Theorem 6 (Hoeffding's Inequality). Let $X_1, \ldots, X_n$ be independent Bernoulli random variables, with $\bar{X} = (1/n)\sum_{i=1}^{n} X_i$ denoting their empirical mean. Then for any $\epsilon > 0$, we have that

$$\Pr\left(\left|\bar{X} - E(\bar{X})\right| \geq \epsilon\right) \leq 2\exp\left(-2n\epsilon^2\right)$$

B Proof of Theorem 1

In this section, we formally prove our lower bound, which is restated here for convenience.

Theorem 1. For any n and any $k \leq n/2$,[8] there exist pairwise preferences over n items satisfying the SST condition such that any algorithm for (ϵ, δ)-PAC top-k identification needs to perform at least $nk/4$ pairwise comparisons for $\delta \leq k/(8(n-1))$.

Proof. By Yao's minimax theorem, it suffices to exhibit a distribution $\mathcal{D}$ over instances (SST models) such that any deterministic algorithm that succeeds on $\mathcal{D}$ with probability at least $1 - k/(8(n-1))$ performs at least $nk/4$ comparisons.

Distribution $\mathcal{D}$ over instances: Each instance in $\mathcal{D}$ contains a partition of n items into 3 groups G_1, G_2, G_3 where G_1 and G_3 contain one item each, and G_2 contains $n-2$ items. The pairwise preferences are defined as follows:

$$P_{ij} = \begin{cases} \frac{1}{2}, & \forall i \in G_1 \cup G_2, j \in G_2 \\ \frac{1}{2}, & \forall i \in G_2, j \in G_3 \\ \frac{1}{2} + \epsilon, & i \in G_1, j \in G_3 \end{cases}.$$

The distribution $\mathcal{D}$ over instances is generated by uniformly at random choosing an element $i \in [n]$ for G_1, uniformly at random choosing an element $j \in [n] \setminus \{i\}$ for G_3 and placing the rest in G_2. Hence, there are a total of $n(n-1)$ instance in the support of $\mathcal{D}$. Note that any subset S of k items that includes the element j above but not element i is invalid. Moreover, this is the only invalid solution.

Now, suppose there is a deterministic algorithm, say A that succeeds on $\mathcal{D}$ with probability $1 - k/8(n-1)$ by making at most q queries. We will show that q must be greater than $nk/4$. We will give algorithm A extra power– whenever it compares any two items i and j even once, the true preference probability P_{ij} between them is revealed. Note that this can only reduce the query complexity of A as any algorithm that uses samples drawn according to P_{ij} instead of actual value of P_{ij} can simply draw its own Bernoulli samples and use these samples.

[8] The assumption $k \leq n/2$ is without loss of generality, because otherwise, we can equivalently identify the bottom-$(n-k)$ items instead.

Observe that as long as A has not compared i and j, then answer is always "equal". Moreover, as soon as A compared i and j, it can output a valid solution– choose any k items that do not include j, for instance. So this means that the decision tree representing A is simply a path. Hence, without loss of generality, we can assume that the algorithm A queries a set Q of pairs of items where $|Q| = q$.

$$
\begin{aligned}
&\Pr_I[A \text{ outputs a valid set } S \text{ of } k \text{ items on } I]\\
&= \Pr[(i,j) \in Q]\cdot\Pr[A \text{ answers correctly} \mid (i,j) \in Q]\\
&\quad + \Pr[(i,j) \notin Q]\cdot\Pr[A \text{ answers correctly} \mid (i,j) \notin Q]\\
&\overset{(i)}{\le} \frac{q}{\binom{n}{2}}\cdot 1 + \left(1 - \frac{q}{\binom{n}{2}}\right)\left(1 - \frac{k(n-k) - q}{n(n-1)}\right)\\
&\overset{(ii)}{\le} \frac{k}{2(n-1)} + \left(1 - \frac{k}{2(n-1)}\right)\left(1 - \frac{kn/2 - kn/4}{n(n-1)}\right)\\
&= \frac{k}{2(n-1)} + 1 - \frac{k}{2(n-1)} - \frac{k}{4(n-1)} + \frac{k^2}{4(n-1)^2}\\
&= 1 - \frac{k}{4(n-1)} + \frac{k^2}{4(n-1)^2}\\
&\le 1 - \frac{k}{4(n-1)} + \frac{k}{8(n-1)}\\
&\le 1 - \frac{k}{8(n-1)},
\end{aligned}
$$

where the inequality (i) above follows from the fact that A fails if (j,i) sits on one of the unqueried edge slots in $|S|\,|V \setminus S|$, and inequality (ii) follows from the fact that $q \le nk/4$.

□

C Proof of Theorem 2

In this section, we will formally analyze the 2-round algorithm presented in Section 3.1 to establish Theorem 2, restated below for convenience.

Theorem 2. Given any set of items $[n]$ with unknown pairwise preferences $P \in [0,1]^{n \times n}$ satisfying SST, any integer $k \in [1, n/2]$, tolerance $\epsilon \in (0,1]$, and confidence $\delta \in (0,1]$, there exists an algorithm for (ϵ,δ)-PAC top-k identification with query complexity $O((1/\epsilon^2)\max\{n^{4/3}, nk\}\log^2(n/\delta))$ and at most 2 rounds.

We shall first prove the correctness guarantee of Algorithm 1, i.e. for any item $i \in T$ in the set T of k items returned by this algorithm, there is no item $j \in [n] \setminus T$ amongst the remaining items with preference probability $P_{ji} \ge 1/2 + \epsilon$. We shall then bound the total number of comparisons made by this algorithm. The fact that this algorithm requires at most 2 rounds of adaptivity is clear.

Given the underlying preferences P satisfying SST, let σ be the true strict ordering consistent with P. For any item $i \in [n]$, we use $\mathrm{rank}(i)$ to refer to the position of item i in σ (items with smaller rank being superior to items with larger rank). Given any $0 < \epsilon \le 1$, we begin by defining the following three events

$$
\begin{aligned}
E_1 &:= \forall i, j \in [n];\ |\hat{p}_{ij} - p_{ij}| < \frac{\epsilon}{4}\\
E_2 &:= \forall i \in [n];\ \exists a \in A : \mathrm{rank}(i) < \mathrm{rank}(a) < \mathrm{rank}(i) + n^{2/3}\\
E_3 &:= |A| > \max\{n^{1/3}, k\}\log(4n/\delta), \text{ and}\\
&\quad\ |A| < 4\max\{n^{1/3}, k\}\log(9n/\delta)
\end{aligned}
$$

Lemma 1. Let $E = E_1 \cap E_2 \cap E_3$. Then event E occurs with probability at least $1 - \delta/3$.

Proof. To prove this claim, we shall show via a standard Hoeffing's inequality that the complement $\bar{E}_1, \bar{E}_2, \bar{E}_2$ of each event occurs with probability at most $\delta/9$, which after a simple union bound implies the complement $\bar{E}$ of event E occurs with probability at most $\delta/3$.

To bound the probability of event E_1, observe that for any fixed pair $i, j \in [n]$, and any precision $\epsilon > 0$, the Hoeffding's inequality bounds the probability of deviation as

$$\Pr(|\hat{P}_{ij} - P_{ij}| \geq \epsilon) \leq 2\exp\left(-2m\epsilon^2\right);$$

where $P_{ij} = E(\hat{P}_{ij})$ is the true preference probability between i, j, and $\hat{P}_{ij}$ is the observed preference probability from m independent comparisons between i, j. Therefore, by a union bound over all pairs $i, j \in [n]$, we have

$$\Pr(E_1) \leq 2\binom{n}{2}\exp\left(-2m\epsilon^2\right);$$

which is at most $\delta/9$ for $m \geq (1/\epsilon^2)\log(3n/\delta)$. The careful reader will recognize that we in fact do not compare all pairs of items, just $O(n^{4/3} + \text{e}nk)$ of them. However, for ease of exposition, we consider an alternative sampling model, where the outcomes of m independent comparisons between all pairs of items are drawn in advance, and when the algorithm queries a pair $i, j \in [n]$, these pre-drawn outcomes are then revealed to the algorithm.

To bound the probability of event E_2, consider any item $i \in [n]$, and let $E_{i;2}$ be the event where there is no anchor $a \in A$ such that $\text{rank}(i) < \text{rank}(a) < \text{rank}(i) + n^{2/3}$, i.e. we do not sample any anchor amongst items in the interval $[\text{rank}(i), \ldots, \text{rank}(i) + n^{2/3}]$. Since every item is sampled into the set of anchors with probability at least $2\log(n/\delta)/n^{2/3}$, we expect $2\log(n/\delta)$ anchors to be chosen from the said interval. By the multiplicative Chernoff bound, we have that the probability of this event is bounded as

$$\Pr(E_{i;2}) \leq \exp\left(-\frac{2\log(9n/\delta)}{2}\right) \leq \frac{\delta}{9n}.$$

Taking a union bound over all items $i \in [n]$ gives us that $\Pr(E_2) \leq \delta/9$.

To bound the probability of event E_3, observe that in expectation, the number of anchors sampled is $2\max\{n^{1/3}, k\}\log(9n/\delta)$. By the multiplicative Chernoff bound, we have the probability of event E_3 is bounded as

$$\Pr(E_3) \leq 2\exp\left(-\frac{\max\{n^{1/3}, k\}\log(9n/\delta)}{4}\right);$$

which is at most $\delta/9$ for either k or n larger than a small constant. □

Henceforth, we shall assume that event E occurs. We first note an implication of event E that will be useful for proving the main theorem.

Corollary 2. Let $a \in A$ be any anchor. For any item $i \in [n]$ such that $p_{ia} \geq 1/2 + \epsilon$, it must be that $i \in W_a$. Furthermore for any item $j \in [n]$ such that $p_{ja} \leq 1/2 + \epsilon/2$, it must be that $j \notin W_a$. Therefore, $W_a \subseteq \{i \in [n] : \text{rank}(i) < \text{rank}(a)\}$, and $|W_a| < \text{rank}(a)$.

Proof. (of theorem 2)

Correctness. We shall first prove that Algorithm 1 produces a valid (ϵ, k)-optimal subset of items with probability at least $1 - \delta/3$ conditioned on event E.

We begin by showing that the initial $k' \leq k$ items added into the partial solution T' are an (ϵ, k')-optimal subset of $[n]$. For the sake of contradiction, let us assume that T' is not an (ϵ, k')-optimal subset, i.e. there exists some pair of items $i \in T'$, $j \in [n] \setminus T'$ such that $P_{ji} \geq 1/2 + \epsilon$. Since $i \in T'$, there must have been some anchor $a \in A$ such that $i \in W_a$ due to which i was added into set T' for the first time. By assumption of P_{ji}, and Corollary 2, it must be that $\text{rank}(j) < \text{rank}(i) < \text{rank}(a)$, and since $P_{ji} \geq 1/2 + \epsilon$, it must be that $p_{ja} \geq 1/2 + \epsilon$ due SST. Therefore, by Corollary 2, it must be that $j \in W_a$. Since T was constructed by including the entire set W_a, it must be that $j \in T$, which is a contradiction. Therefore, we can conclude that the partial solution T' constructed thus far is (ϵ, k')-optimal.

Next, we shall prove that the remaining $k - k'$ items added into the partial solution T', creating our final solution T are an $(\epsilon, k - k')$-optimal subset of the remaining items $[n] \setminus T'$. This together with our previous claim would imply that T is (ϵ, k)-optimal.

Case 1: $(t = k + 1)$.

In this case, we have that $T' = \cup_{i=1}^{k} W_{a_i}$. Therefore, by Corollary 2, it must be that for any $a \in R; j \notin T', p_{ja} < 1/2 + \epsilon$, implying that every item $a \in R \setminus T'$ is $(\epsilon; 1)$-optimal amongst the remaining items, and can be added into set T'. Furthermore, it is also easy to see that $|R \setminus T'| \geq k - k'$, as $|R| = k$ by definition of Case 1. Thus, $T = T' \cup R_a$ is $(\epsilon; k)$-optimal.

Case 2: $(t \leq k)$.

In this case, we will first show that there exists a set of $k - k'$ items in $W_{a_t} \setminus T'$ itself that is $(\epsilon; k - k')$-optimal amongst all remaining items $[n] \setminus T'$. By definition of Case 2, it must be that $|T \cup W_{a_t}| > k$ which implies that $|W_{a_t} \setminus T| > k - k'$. For the sake of contradiction, let us assume that $W_{a_t} \setminus T'$ does not contain an $(\epsilon; k - k')$-optimal subset amongst all remaining items, i.e. there exists a pair of items $i \in W_{a_t} \setminus T'; j \in ([n] \setminus T') \setminus W_{a_t}$ such that $p_{ji} \geq 1/2 + \epsilon$. Due to Corollary 2, it must be that rank$(j) <$ rank$(i) <$ rank(a_t), and since $p_{ji} \geq 1/2 + \epsilon$, it must be that $p_{ja_t} \geq 1/2 + \epsilon$ due to SST. Therefore, by Corollary 2, it must be that $j \in W_{a_t}$, which is a contradiction to the assumption $j \in ([n] \setminus T') \setminus W_{a_t}$. Therefore, it suffices to look inside set $W_{a_t} \setminus T'$ alone to find an $(\epsilon; k - k')$-optimal subset of the remaining items $[n] \setminus T'$ to fill the available budget, and the rest can be safely discarded.

Case 2a: ($t \leq k$ and $|W_{a_t} \setminus T'| \leq 4 \max\{\sqrt{nk}, n^{2/3}\}$).

In this case, we compare all pairs of items in the set $W_{a_t} \setminus T'$ and select a set $C \subseteq W_{a_t} \setminus T'$ of $(k - k')$ items such that for any item $i \in C$, and any item $j \in (W_{a_t} \setminus T') \setminus C$, $\hat{P}_{ji} < 1/2 + 3\epsilon/4$. Therefore by Event E_1, it must be that for any pair $i \in C; j \in (W_{a_t} \setminus T') \setminus C$, $p_{ji} < 1/2 + \epsilon$. Thus, $T = T' \cup C$ is $(\epsilon; k)$-optimal.

We further note that if $t = 1$, i.e. $|W_{a_1}| > k$, then we must fall into this Case 2a. To see this, consider the anchor $a_{min} \in A$ of minimum rank. Observe that Event E_2 guarantees the existence of an anchor $a \in A$ such that rank$(a) < n^{2/3}$, and Corollary 2 consequently guarantees that $|W_a| < n^{2/3}$. Therefore, we have $|W_{a_{min}}| <$ rank$(a_{min}) < n^{2/3}$. the first anchor a_1 can have one of two possible relations to a_{min}: (1) either $a_1 = a_{min}$, which directly puts us in Case 2a as proved earlier, or (2) $a_1 \neq a_{min}$, which implies $1/2 - \epsilon/4 < \hat{P}_{a_1 a_{min}} < 1/2 + \epsilon/4$, and $|W_{a_1}| \leq |W_{a_{min}}|$ due to our sorting rule, which also puts us in Case 2a. The case $a_1 \neq a_{min}$ and $\hat{P}_{a_1 a_{min}} \geq 1/2 + \epsilon/4$ is refuted by Event E_1. Henceforth, we shall assume that at least one anchor has been parsed, i.e. $t > 1$.

Case 2b: ($t \leq k$ and $|W_{a_t} \setminus T'| > 4 \max\{\sqrt{nk}, n^{2/3}\}$)

Let $a^{T'}_{max} := \arg\max_{a_i \in A: i < t}$ rank(a_i) be the highest ranking anchor amongst $a_1, \ldots, a_{t-1}$ (such an anchor must exist since $t > 1$), and let $k^{T'}_{min} := \arg\min_{i \in [n]: i \notin T}$ rank(i) be the lowest ranking "true" top-k item not already included into our partial solution T' (such an item must exist since $|T'| < k$). Since $k^{T'}_{min} \notin T'$, it must be that $k^{T'}_{min} \notin W_{a^{T'}_{max}}$. This can only occur if $\hat{P}_{k^{T'}_{min} a^{T'}_{max}} < 1/2 + 3\epsilon/4$, implying $P_{k^{T'}_{min} a^{T'}_{max}} < 1/2 + \epsilon$ due to Corollary 2. Therefore, due to SST, every item $i \notin T'$ with rank$(i) \leq$ rank$(a^{T'}_{max})$ must be $(\epsilon; 1)$-optimal amongst the remaining set of items $[n] \setminus T'$, and any arbitrary subset of $k - k'$ items from this set can be added to set T'. Let $G := \{i : i \notin T'; \text{rank}(i) \leq \text{rank}(a^{T'}_{max})\}$ be this set. Consider any set $S_h = \{j \in [n] \setminus T' : \hat{P}_{ja_h} \geq 1/2 + \epsilon/4\}$ for $1 \leq h < t$. By event E_1, it must be that for any item $i \in S_h$, $P_{ia_h} > 1/2$, and therefore, $P_{ia^{T'}_{max}} > 1/2$ by definition of anchor $a^{T'}_{max}$, due to which we can conclude that $S_h \subseteq G$ for every $1 \leq h < t$. Therefore, if we have that $|S| = |\cup_{h<t} S_h| \geq k - k'$, then any arbitrary $k - k'$ items from S can be added into T' to fill the available budget. If not, then we extend our partial solution $T'' = T' \cup S$ by including all of S. Therefore, for any $(\epsilon; 1)$-optimal item $i \in G \setminus T''$ that was left out, it must be that $\hat{P}_{ia^{T'}_{max}} < 1/2 + \epsilon/4$, which would imply $P_{ia^{T'}_{max}} < 1/2 + \epsilon/2$ due to Event E_1.

Let $a^{[n]\setminus T'}_{min} := \arg\min_{a_i \in A: i \geq t}$ rank(a) be the lowest ranking anchor amongst the unparsed anchors $a_t, \ldots, a_{|A|}$. We again have the following two cases:

Case 2b (1): $a_t = a^{[n]\setminus T'}_{min}$.

By Event E_2, we have rank$(a_t) <$ rank$(a^{T'}_{max}) + n^{2/3}$, and by Corollary 2, we can conclude that W_{a_t} can only contain items with rank at most rank(a_t). However, as proved earlier, any item with rank at most rank$(a^{T'}_{max})$ is $(\epsilon; 1)$-optimal amongst the remaining items. Therefore, within $W_{a_t} \setminus T''$, at most $n^{2/3}$ items will have rank larger than $\text{rank}(a^{T^0}_{max})$, and therefore might not be $(\epsilon;1)$-optimal. However, observe that $|W_{a_t}$

$$\setminus T^{00}| > |W_{a_t} \setminus T^0| - k > 4\max\{\sqrt{nk}; n^{2/3}\} - k > 3n^{2/3};$$

with the first inequality following due to the fact that $T^{00} = T^0 \cup S$ and $|S| < k - k^0$, and the second inequality following by definition of Case 2b. Therefore, at least a 2/3 fraction of items within $W_{a_t} \setminus T^{00}$ will have rank at most $\text{rank}(a^{T^0}_{max})$ and consequently, will belong to set G (are $(\epsilon;1)$-optimal). By a standard Chernoff bound, the probability that we do not sample at least $k - k^0$ items from set G in the set C_{a_t} of $6k\log(3/\delta)$ items chosen uniformly at random from $W_{a_t} \setminus T^{00}$ is at most $\delta/3$. Let us condition on the event that we sample at least $k - k^{00}$ items from set G into set C_{a_t}, and let $C^0 \subseteq G$ be this corresponding set. We shall finally prove that for any item $i \in C^0$, there is no item $j \in W_{a_t} \setminus T^{00}$ such that $P_{ji} \geq 1/2 + 3\epsilon/4$. To see this, observe that for any item $i \in C^0$, there is no item in $j \in W_{a_t} \setminus T^{00}$ with pairwise preference $P_{ji} \geq 1/2 + \epsilon/2$. This follows from our previously proved claim that for any $i \in G \setminus T^{00}; P_{i a^{T^0}_{max}} < 1/2 + \epsilon/2$, implying that for any pair $i;j \in G \setminus T^{00}; P_{ij} < 1/2 + \epsilon/2$ due to SST. Event E_1 subsequently guarantees that for any pair $i;j \in G \setminus T^{00}; \hat{P}_{ij} < 1/2 + 3\epsilon/4$. Lastly, for any pair $i \in G \setminus T^{00}; j \in (W_{a_t} \setminus T^{00}) \setminus G; P_{ji} < 1/2$ since $\text{rank}(j) < \text{rank}(i)$. Event E_1 subsequently guarantees that $\hat{P}_{ji} < 1/2 + \epsilon/4$ for such pairs. Therefore, conditioning on sampling at least $k - k^0$ items from set G into set C_{a_t}, we are guaranteed to find such a set C^0.

Case 2b (2): $a_t = a^{[n]\setminus T^0}_{min}$.

We shall further assume that $\text{rank}(a_t) > \text{rank}(a^{[n]\setminus T^0}_{min})$ since otherwise, every item in $W_{a_t} \setminus T^{00}$ is $(\epsilon;1)$-optimal amongst the remaining items in $[n] \setminus T^{00}$ and the rest of the proof would follow identically to that of Case 2b (1). If $\text{rank}(a_t) > \text{rank}(a^{[n]\setminus T^0}_{min})$, then by our sorting rule for set A, it must be the case that $\hat{P}_{a^{[n]\setminus T^0}_{min} a_t} < 1/2 + \epsilon/4$, implying $P_{a^{[n]\setminus T^0}_{min} a_t} < 1/2 + \epsilon/2$ due to Event E_1, which also refutes the other possibility of $\hat{P}_{a^{[n]\setminus T^0}_{min} a_t} \geq 1/2 + \epsilon/4$. Therefore, due to SST, it must be that for any item $j \in [n] : \text{rank}(a^{[n]\setminus T^0}_{min}) \leq \text{rank}(j) < \text{rank}(a_t)$, $P_{j a_t} < 1/2 + \epsilon/2$. Therefore, by Corollary 2, W_{a_t} cannot contain any item with rank larger than $\text{rank}(a^{[n]\setminus T^0}_{min})$. The rest of the proof is now identical to that of Case 2b (1).

Since event E occurs with probability at least $1 - \delta/3$, and conditioned on event E, the algorithm succeeds with probability at least $1 - \delta/3$ in Case 2b, we can conclude that the algorithm succeeds in returning a set T, which is an $(\epsilon;k)$-optimal subset of $[n]$ with probability at least $1 - \delta$.

Rounds and Query Complexity. It is clear that the algorithm has at most 2 sequential rounds of queries, with the total number of queries bounded by $O(nm|A|)$ in the first round, and one of either $O(m\max\{nk; n^{4/3}\})$ or $O(nmk\log(1/\delta))$ in the second round. Therefore, the total number of comparisons is bounded by $O\left((1/\epsilon^2)\max\{n^{4/3}; nk\}\log^2(n/\delta)\right)$. □

D Proof of Theorem 3

In this section, we will formally analyze the 3-round algorithm presented in Section 3.2 to establish Theorem 3, restated below for convenience.

Theorem 3. Given any set of items $[n]$ with unknown pairwise preferences $P \in [0;1]^{n \times n}$ satisfying SST, any integer $k \in [1; n/2]$, tolerance $\epsilon \in (0;1]$, and confidence $\delta \in (0;1]$, there exists an algorithm for $(\epsilon;\delta)$-PAC top-k selection with query complexity $O((1/\epsilon^2)nk\log^2(n/\delta))$ comparisons and at most 3 rounds of adaptivity.

This analysis is almost identical to the proof of Theorem 2, with a few minor differences. We include the entire proof here nevertheless. We shall first prove the correctness guarantee of Algorithm 3, i.e. for any item $i \in T$ in the set T of k items returned by this algorithm, there is no item $j \in [n] \setminus T$ amongst the remaining items with preference probability $P_{ji} \geq 1/2 + \epsilon$. We shall then bound the total number of comparisons made by this algorithm. The fact that this algorithm requires at most 3 sequential rounds of querying is clear.

Given the underlying preferences P satisfying SST, let σ be the true strict ordering consistent with P. For any item $i \in [n]$, we use $\text{rank}(i)$ to refer to the position of item i in σ (items with smaller rank being superior to items with larger rank). Given any $0 < \delta \leq 1$, we begin by defining the following three events

$$
\begin{aligned}
E_1 &:= \forall i, j \in [n];\ |\hat{p}_{ij} - p_{ij}| < \frac{\epsilon}{4} \\
E_2 &:= \forall i \in [n];\ \exists a \in A : \mathrm{rank}(i) < \mathrm{rank}(a) < \mathrm{rank}(i) + \sqrt{n} \\
E_3 &:= |A| > \max\{\sqrt{n}, k\} \log(9n/\delta); \text{ and} \\
&\quad\ |A| < 4 \max\{\sqrt{n}, k\} \log(9n/\delta)
\end{aligned}
$$

Lemma 2. Let $E = E_1 \cap E_2 \cap E_3$. Then event E occurs with probability at least $1 - \delta/3$.

Proof. To prove this claim, we shall show via a standard Hoeffing's inequality that the complement $\bar{E}_1; \bar{E}_2; \bar{E}_2$ of each event occurs with probability at most $\delta/9$, which after a simple union bound implies the complement $\bar{E}$ of event E occurs with probability at most $\delta/3$.

To bound the probability of event E_1, observe that for any fixed pair $i, j \in [n]$, and any precision $\epsilon > 0$, the Hoeffding's inequality bounds the probability of deviation as

$$\Pr(|\hat{P}_{ij} - P_{ij}| \geq \epsilon) \leq 2 \exp\left(-2m\epsilon^2\right);$$

where $P_{ij} = E(\hat{P}_{ij})$ is the true preference probability between i, j, and $\hat{P}_{ij}$ is the observed preference probability from m independent comparisons between i, j. Therefore, by a union bound over all pairs $i, j \in [n]$, we have

$$\Pr(\bar{E}_1) \leq 2 \binom{n}{2} \exp\left(-2m\epsilon^2\right);$$

which is at most $\delta/9$ for $m \geq (1/\epsilon^2) \log(3n/\delta)$. The careful reader will recognize that we in fact do not compare all pairs of items, just $O(nk)$ of them. However, for ease of exposition, we consider an alternative sampling model, where the outcomes of m independent comparisons between all pairs of items are drawn in advance, and when the algorithm queries a pair $i, j \in [n]$, these pre-drawn outcomes are then revealed to the algorithm.

To bound the probability of event E_2, consider any item $i \in [n]$, and let $E_{i,2}$ be the event where there is no anchor $a \in A$ such that $\mathrm{rank}(i) < \mathrm{rank}(a) < \mathrm{rank}(i) + \sqrt{n}$, i.e. we do not sample any anchor amongst items in the interval $[\mathrm{rank}(i), \ldots, \mathrm{rank}(i) + \sqrt{n}]$. Since every item is sampled into the set of anchors with probability at least $2\log(n/\delta)/\sqrt{n}$, we expect $2\log(n/\delta)$ anchors to be chosen from the said interval. By the multiplicative Chernoff bound, we have that the probability of this event is bounded as

$$\Pr(E_{i,2}) \leq \exp\left(-\frac{2\log(9n/\delta)}{2}\right) \leq \frac{\delta}{9n}.$$

Taking a union bound over all items $i \in [n]$ gives us that $\Pr(\bar{E}_2) \leq \delta/9$.

To bound the probability of event E_3, observe that in expectation, the number of anchors sampled is $2\max\{\sqrt{n}, k\} \log(9n/\delta)$. By the multiplicative Chernoff bound, we have the probability of event E_3 is bounded as

$$\Pr(\bar{E}_3) \leq 2 \exp\left(-\frac{\max\{\sqrt{n}, k\} \log(9n/\delta)}{4}\right);$$

which is at most $\delta/9$ for either k or n larger than a small constant. □

Henceforth, we shall assume that event E occurs. We first note an implication of event E that will be useful for proving the main theorem.

Corollary 3. Let $a \in A_k$ be any anchor amongst the pruned set of anchors. For any item $i \in [n]$ such that $p_{ia} \geq 1/2 + \epsilon$, it must be that $i \in W_a$. Furthermore for any item $j \in [n]$ such that $p_{ja} \leq 1/2 + \epsilon/2$, it must be that $j \notin W_a$. Therefore, $W_a \subseteq \{i \in [n] : \mathrm{rank}(i) < \mathrm{rank}(a)\}$, and $|W_a| < \mathrm{rank}(a)$.

Proof. (of theorem 3)

Correctness. We shall first prove that Algorithm 1 produces a valid (; k)-optimal subset of items with probability at least 1 =3 conditioned on event E.

We begin by showing that the initial k^0 k items added into the partial solution T^0 are an (; k^0)-optimal subset of [n]. For the sake of contradiction, let us assume that T^0 is not an (; k^0)-optimal subset, i.e. there exists some pair of items i 2 T^0, j 2 [n] n T^0 such that P_{ji} 1=2 + . Since i 2 T^0, there must have been some anchor a 2 A_k such that i 2 W_a due to which i was added into set T^0 for the first time. By assumption of P_{ji}, and Corollary 2, it must be that rank(j) < rank(i) < rank(a), and since P_{ji} 1=2 + , it must be that p_{ja} 1=2 + due SST. Therefore, by Corollary 2, it must be that j 2 W_a. Since T was constructed by including the entire set W_a, it must be that j 2 T, which is a contradiction. Therefore, we can conclude that the partial solution T^0 constructed thus far is (; k^0)-optimal.

Next, we shall prove that the remaining k k^0 items added into the partial solution T^0, creating our final solution T are an (; k k^0)-optimal subset of the remaining items [n] n T^0. This together with our previous claim would imply that T is (; k)-optimal.

Case 1: (t = k + 1).

In this case, we have that T^0 = [$^k_{i=1}$ W_{a_i}. Therefore, by Corollary 3, it must be that for any a 2 R; j 2 T^0, p_{ja} < 1=2 + , implying that every item a 2 R n T^0 is (; 1)-optimal amongst the remaining items, and can be added into set T^0. Furthermore, it is also easy to see that jR n T^0j k k^0, as jRj = k by definition of Case 1. Thus, T = T^0 [R_a is (; k)-optimal.

Case 2: (t k).

In this case, we will first show that there exists a set of k k^0 items in W_{a_t} n T^0 itself that is (; k k^0)-optimal amongst all remaining items [n] n T^0. By definition of Case 2, it must be that jT [W_{a_t} j > k which implies that jW_{a_t} n Tj > k k^0. For the sake of contradiction, let us assume that W_{a_t} n T^0 does not contain an (; k k^0)-optimal subset amongst all remaining items, i.e. there exists a pair of items i 2 W_{a_t} n T^0; j 2 ([n] n T^0) n W_{a_t} such that p_{ji} 1=2 + . Due to Corollary 2, it must be that rank(j) < rank(i) < rank(a_t), and since p_{ji} 1=2 + , it must be that p_{ja} 1=2 + due to SST. Therefore, by Corollary 3, it must be that j 2 W_{a_t}, which is a contradiction to the assumption j 2 ([n] n T^0) n W_{a_t}. Therefore, it suffices to look inside set W_{a_t} n T^0 alone to find an (; k k^0)-optimal subset of the remaining items [n] n T^0 to fill the available budget, and the rest can be safely discarded.

Case 2a: (t k and jW_{a_t} n T^0j 4 $\sqrt{nk}$).

In this case, we compare all pairs of items in the set W_{a_t} n T^0 and select a set C W_{a_t} n T^0 of (k k^0) items such that for any item i 2 C, and any item j 2 (W_{a_t} n T^0) n C, $\hat{P}_{ji}$ < 1=2 + 3=4. Therefore by Event E_1, it must be that for any pair i 2 C; j 2 (W_{a_t} n T^0) n C, p_{ji} < 1=2 + . Thus, T = T^0 [C is (; k)-optimal.

We further note that if t = 1, i.e. jW_{a_1}j > k, then we must fall into this Case 2a. To see this, consider the anchor a_{min} 2 A of minimum rank. Observe that Event E_2 guarantees the existence of an anchor a 2 A such that rank(a) < $\sqrt{n}$, and Corollary 3 consequently guarantees that jW_aj < $\sqrt{n}$. Therefore, we have j$W_{a_{min}}$j < rank(a_{min}) < $\sqrt{n}$. The first anchor a_1 can have one of two possible relations to a_{min}: (1) either a_1 = a_{min}, which directly puts us in Case 2a as proved earlier, or (2) a_1 = a_{min}. In this case, we again have to deal with two cases: either a_{min} 2 A_k, in which case it must be that $\hat{P}_{a_{min} a_1}$ < 1=2 + =4 due to our selection rule, implying $P_{a_{min} a_1}$ < 1=2 + =2 due to Event E_1. In this case, due to SST, it must be that for any item i : rank(a_{min}) < rank(i) < rank(a_1), P_{ia_1} < 1=2 + =2, implying i 2 W_{a_1} by Corollary 3. Therefore, it must be that jW_{a_1} j rank(a_{min}) < $\sqrt{n}$, putting us in Case 2a. Otherwise, a_{min} 2 A_k in which case it must be that 1=2 =4 < $\hat{P}_{a_1 a_{min}}$ < 1=2 + =4 and jW_{a_1} j j$W_{a_{min}}$ j due to our sorting rule, which again puts us in Case 2a. The case a_{min} 2 A_k and $\hat{P}_{a_1 a_{min}}$ 1=2 + =4 is refuted by Event E_1. Henceforth, we shall assume that at least one anchor has been parsed, i.e. t > 1.

Case 2b: (t k and jW_{a_t} n T^0j > 4 $\sqrt{nk}$)

Let $a^{T^0}_{max}$:= argmax$_{a_i 2 A: i<t}$ rank(a_i) be the highest ranking anchor amongst $a_1; \ldots ; a_{t-1}$ (such an anchor must exist since t > 1), and let $k^{T^0}_{min}$:= argmin$_{i 2 [n]: i 2 T}$ rank(i) be the lowest ranking "true" top-k item not already included into our partial solution T^0 (such an item must exist since jT^0j < k). Since $k^{T^0}_{min}$ 2 T^0, it must be that

$k^{T^0}_{min} \notin W_{a^T_{max}}$. This can only occur if $\hat{P}_{k^{T^0}_{min} a^{T^0}_{max}} < 1/2 + 3\epsilon/4$, implying $P_{k^{T^0}_{min} a^{T^0}_{max}} < 1/2 + \epsilon$ due to Corollary 2. Therefore, due to SST, every item $i \notin T^0$ with $\mathrm{rank}(i) \le \mathrm{rank}(a^{T^0}_{max})$ must be $(\epsilon, 1)$-optimal amongst the remaining set of items $[n] \setminus T^0$, and any arbitrary subset of $k - k^0$ items from this set can be added to set T^0. Let $G := \{i : i \notin T^0, \mathrm{rank}(i) \le \mathrm{rank}(a^{T^0}_{max})\}$ be this set. Consider any set $S_h = \{j \in [n] \setminus T^0 : \hat{P}_{j a_h} \ge 1/2 + \epsilon/4\}$ for $1 \le h < t$. By event E_1, it must be that for any item $i \in S_h$, $P_{i a_h} > 1/2$, and therefore, $P_{i a^{T^0}_{max}} > 1/2$ by definition of anchor $a^{T^0}_{max}$, due to which we can conclude that $S_h \subseteq G$ for every $1 \le h < t$. Therefore, if we have that $|S| = |\cup_{h<t} S_h| \ge k - k^0$, then any arbitrary $k - k^0$ items from S can be added into T^0 to fill the available budget. If not, then we extend our partial solution $T^{00} = T^0 \cup S$ by including all of S. Therefore, for any $(\epsilon, 1)$-optimal item $i \in G \setminus T^{00}$ that was left out, it must be that $\hat{P}_{i a^T_{max}} < 1/2 + \epsilon/4$, which would imply $P_{i a^T_{max}} < 1/2 + \epsilon/2$ due to Event E_1.

Let $a^{[n]\setminus T^0}_{min} := \mathrm{argmin}_{a_i \in A_k : i \ge t}\ \mathrm{rank}(a)$ be the lowest ranking anchor amongst the unparsed anchors $a_t, \ldots, a_k$. We again have the following two cases:

Case 2b (1): $a_t = a^{[n]\setminus T^0}_{min}$.

By Event E_2, we have $\mathrm{rank}(a_t) < \mathrm{rank}(a^{T^0}_{max}) + \sqrt{n}$, and by Corollary 2, we can conclude that W_{a_t} can only contain items with rank at most $\mathrm{rank}(a_t)$. However, as proved earlier, any item with rank at most $\mathrm{rank}(a^{T^0}_{max})$ is $(\epsilon, 1)$-optimal amongst the remaining items. Therefore, within $W_{a_t} \setminus T^{00}$, at most $\sqrt{n}$ items will have rank larger than $\mathrm{rank}(a^{T^0}_{max})$, and therefore might not be $(\epsilon, 1)$-optimal. However, observe that

$$|W_{a_t} \setminus T^{00}| \ge |W_{a_t} \setminus T^0| - k > 4\sqrt{nk} - k > 3\sqrt{nk},$$

with the first inequality following due to the fact that $T^{00} = T^0 \cup S$ and $|S| < k - k^0$, and the second inequality following by definition of Case 2b. Therefore, at least a $2/3$ fraction of items within $W_{a_t} \setminus T^{00}$ will have rank at most $\mathrm{rank}(a^{T^0}_{max})$ and consequently, will belong to set G (are $(\epsilon, 1)$-optimal). By a standard Chernoff bound, the probability that we do not sample at least $k - k^0$ items from set G in the set C_{a_t} of $6k\log(3/\delta)$ items chosen uniformly at random from $W_{a_t} \setminus T^{00}$ is at most $\delta/3$. Let us condition on the event that we sample at least $k - k^{00}$ items from set G into set C_{a_t}, and let $C^0 \subseteq G$ be this corresponding set. We shall finally prove that for any item $i \in C^0$, there is no item $j \in W_{a_t} \setminus T^{00}$ such that $P_{ji} \ge 1/2 + 3\epsilon/4$. To see this, observe that for any item $i \in C^0$, there is no item in $j \in W_{a_t} \setminus T^{00}$ with pairwise preference $P_{ji} \ge 1/2 + \epsilon/2$. This follows from our previously proved claim that for any $i \in G \setminus T^{00}, P_{i a^{T^0}_{max}} < 1/2 + \epsilon/2$, implying that for any pair $i, j \in G \setminus T^{00}, P_{ij} < 1/2 + \epsilon/2$ due to SST. Event E_1 subsequently guarantees that for any pair $i, j \in G \setminus T^{00}, \hat{P}_{ij} < 1/2 + 3\epsilon/4$. Lastly, for any pair $i \in G \setminus T^{00}, j \in (W_{a_t} \setminus T^{00}) \setminus G$, $P_{ji} < 1/2$ since $\mathrm{rank}(j) < \mathrm{rank}(i)$. Event E_1 subsequently guarantees that $\hat{P}_{ji} < 1/2 + \epsilon/4$ for such pairs. Therefore, conditioning on sampling at least $k - k^0$ items from set G into set C_{a_t}, we are guaranteed to find such a set C^0.

Case 2b (2): $a_t = a^{[n]\setminus T^0}_{min}$.

We shall further assume that $\mathrm{rank}(a_t) > \mathrm{rank}(a^{[n]\setminus T^0}_{min})$ since otherwise, every item in $W_{a_t} \setminus T^{00}$ is $(\epsilon, 1)$-optimal amongst the remaining items in $[n] \setminus T^{00}$ and the rest of the proof would follow identically to that of Case 2b (1). If $\mathrm{rank}(a_t) > \mathrm{rank}(a^{[n]\setminus T^0}_{min})$, then by our sorting rule for set A_k, it must be the case that $\hat{P}_{a^{[n]\setminus T^0}_{min} a_t} < 1/2 + \epsilon/4$, implying $P_{a^{[n]\setminus T^0}_{min} a_t} < 1/2 + \epsilon/2$ due to Event E_1, which also refutes the other possibility of $\hat{P}_{a^{[n]\setminus T^0}_{min} a_t} \ge 1/2 + \epsilon/4$. Therefore, due to SST, it must be that for any item $j \in [n] : \mathrm{rank}(a^{[n]\setminus T^0}_{min}) \le \mathrm{rank}(j) < \mathrm{rank}(a_t)$, $P_{j a_t} < 1/2 + \epsilon/2$. Therefore, by Corollary 2, W_{a_t} cannot contain any item with rank larger than $\mathrm{rank}(a^{[n]\setminus T^0}_{min})$. The rest of the proof is now identical to that of Case 2b (1).

Since event E occurs with probability at least $1 - 2\delta/3$, and conditioned on event E, the algorithm succeeds with probability at least $1 - \delta/3$ in Case 2b, we can conclude that the algorithm succeeds in returning a set T, which is an (ϵ, k)-optimal subset of $[n]$ with probability at least $1 - \delta$.

Rounds and Query Complexity. It is clear that the algorithm has at most 3 sequential rounds of queries, with the total number of queries bounded by $O\left(m|A|^2\right)$ in the first round, $O(nmk)$ in the second round, and one of either $O(nmk)$ or $O(nmk\log(1/\delta))$ in the third round. Therefore, the total number of comparisons is bounded by $O\left((1/\epsilon^2) nk \log^2(n/\delta)\right)$.

□

E Almost Optimal Query Complexity for Top-k Identification in O(log n) Rounds

In this section we will design an algorithm that further improves the query complexity achieved by the 3-rounds algorithm by a log(n) factor using O(log n) rounds. The following theorem gives the main result, which is restated here for convenience.

Theorem 4. There exists an algorithm that given any integer k 2 [n=2], rounds r, set of items [n] with an unknown underlying preference matrix P 2 $[0; 1]^{nn}$ satisfying the SST condition, tolerance , confidence , returns a (; k)-optimal set of items with probability at least 1 using O $_2$ $\frac{1}{}$nk($\log^{(r)}$(n) + log(k=)) + $k^2 \log^3$(k=) comparisons and 2r + 4 rounds of adaptivity.

Corollary 1. In the setting of the above theorem, there exists an algorithm that returns a (; k)-optimal set of items with probability at least 1 using O $_2$$\frac{1}{}$(nk + $k^2 \log^2$(k=)) log(k=) comparisons and 2 log(n) + 4 rounds of adaptivity.

We will first present our algorithm and following by its analysis.

E.1 Algorithm

A common approach in designing an algorithm for top-k identification is to find an anchor that has rank close to k, and then find all items that are better than this anchor. However, as discussed in the introduction (Section 1), the main difficulty under our PAC SST setting is that such filtering of items based on comparisons with the anchor is difficult. This is because we operate under precision whereas the gaps between items might be arbitrarily small. More precisely, even if we find an anchor of rank close to k, the gap between the top ranked items and the anchor might be too small, so we might not be able to filter out any of these items as we are using a very coarse funnel. However, one can observe that the anchor can be a part of the PAC top-k solution if these gaps are very small. Hence, the idea is to find k unique anchors of rank close to k, so that we have at least k items to fall back, in case we are not able to find better items.

Therefore, the first step in this algorithm is to select a partition of n items into k groups fS_ig$_{i2[k]}$, where each item is assigned to one of the k groups chosen uniformly at random. Using a standard concentration bound, one can show that there is at least one item a$_i$ of rank O(k log k) in each group S_i, which can potentially serve as an anchor. Hence, the next step is to find such an anchor from each group. However, since we are operating under the (;)-PAC setting, we might not be able to find this item a$_i$ exactly. Instead, we are only guaranteed to find an item a_i which is ^{0}-close to a$_i$ for a given precision ^{0}. In other words, we can only find an item a_i in S_i with ^{0}-rank at most O(k log k), i.e. the number of items that are ^{0}-better than a_i are at most O(k log k). In our analysis we show that these anchors can act as coarse funnels and filter items that are -better than them if we set ^{0} to be =3. If there are very few items that are -better than any anchor in fa_ig$^k_{i=1}$ then we can fall back to some of these anchors.

In order to find these anchors fa_ig$^k_{i=1}$, we call our Top-1 algorithm (Algorithm 6) in parallel for each group in fS_ig$^k_{i=1}$. The Top-1 algorithm guarantees that each a_i is an =3-best item in S_i which in turn guarantees that a_i has =3-rank of O(k log k). This Top-1 algorithm is discussed in detail in Appendix F and is similar to the top-1 algorithm of Cohen-Addad et al. (2020) designed for the noisy comparison model. These k parallel calls to the Top-1 algorithm use the first r + 4 rounds of adaptivity.

Now, once we have found these k anchors, we need to find all the items that are -better than any of these anchors. A simple approach is to simply compare all the n items to each of the anchors O(log n) times and find all -better items. However, the complexity of this operation will be O(nk log n) which can be off by a log n factor for small k. Hence, in order to improve upon this we use an elimination algorithm (Algorithm 5) similar to the Aggressive-Elimination algorithm of Agarwal et al. (2017). This algorithm does not use log n comparisons for each pair, rather it eliminates items in sequential rounds increasing the number of comparisons per pair in each round. We make k parallel calls to the Eliminate subroutine, where the i-th call is with respect to anchor a$_i$. This subroutine uses r rounds of interaction, and in round r compares each of the remaining items with the anchor t_r := O($_{"2}$ $\log^{(r)}$$\frac{1}{2}$(n)) times. It then calculates empirical estimates of the preference probabilities of these items against the anchor. Based on these empirical estimates it decides to retain a $1/\log^{(r-1)}(n)$ fraction of the current items for the next round, and eliminate all the other items. The elimination strategy gets more aggressive over rounds as $\log^{(r-1)}(n)$ increases monotonically with r. The Eliminate subroutine corresponding to anchor a_i returns all the items in $[n]$ which are -better than a_i and necessarily excludes any item which is $/3$-worse than a_i.

The final step in the algorithm is to combine all the 'good' items obtained through parallel calls to Eliminate into one group and perform all pairwise comparisons within that group. The set of top-k items is then any k items that are not -worse than any other remaining items. The pseudo-code for the algorithm is given in Algorithm 4.

Algorithm 4 (;)-PAC top-k

Input: items $[n]$, parameter k, rounds $2r+4$, accuracy , confidence

Let $\{S_i\}_{i=1}^k$ be a partition of $[n]$ created by assigning each element $j \in [n]$ to $S \in \{S_i\}_{i=1}^k$ uniformly at random.

In Parallel for all $i \in [k]$, a_i Top-1$(S_i; r+3; /3; /4k)$ (Algorithm 6):

In Parallel for all $i \in [k]$, W_i Eliminate$([n]; r; a_i; /4k; /2; /6; k\log(4k/))$

Let W $\bigcup_{i \in [k]} W_i$

Compare all pairs in W, $O(\log(4k/)/\ )^2$ times.

Output: if $|W|$ k then output an $(; k)$-optimal solution in W, else output W plus an arbitrary set of $k - |W|$ anchors from $\{a_i\}_{i=1}^k \setminus W$

Algorithm 5 Eliminate$(S; r; a; ; ; \varepsilon; k')$

Input: set of items S, remaining rounds r, $m := |S|$, anchor a, confidence , accuracy , precision ε, upper bound k'

Let $t_r := \frac{2}{\varepsilon^2}\ \ \log^{(r)}(m) + \log(8k'/)$.

Compare each item $i \in S$ with a for t_r times.

Let $\hat{p}_{ia}$ be the empirical probability of i beating a

Sort the items in decreasing order of $\hat{p}_{ia}$ values

if $r = 1$ **then**

  Return: S' $\{i \in S : \hat{p}_{ia} \ \frac{1}{2} + \}$

else

  Let $m_{r-1} := k' + \frac{m}{\log^{(r-1)}(m)}$ and let S' be the m_{r-1} top most items according to $\hat{p}$

end if

if m_{r-1} $2k'$ **then**

  Return: Eliminate$(S'; 1; a; /2; ; \varepsilon; k')$.

else

  Return: Eliminate$(S'; r-1; a; /2; ; \varepsilon; k')$.

end if

E.2 Analysis

In order to prove Theorem 4, we will use the following theorem about the correctness of Top-1 algorithm (Algorithm 6) given in Appendix F.

Theorem 7 (Top-1 Correctness). For any set S $[n]$, rounds r 1, confidence > 0, accuracy > 0, the Top-1 algorithm given in Algorithm 6 returns an item a such that any $i \in S$ satisfies $P_{ia} < \frac{1}{2} + $. The algorithm succeeds with probability at least 1 , uses $O\ \ \frac{}{\varepsilon^2}\ \ \log^{(r)}(n) + \log(1/)$ comparisons and at most $r+4$ rounds of adaptivity.

For a given > 0 and $j \in [n]$, we will define the -rank of j to be the number of items that beat j with probability more than $\frac{1}{2} + $, i.e.

$$\text{rank}(j) := |\{i \in [n] : P_{ij}\ \ 1/2 + \}|$$

The first lemma bounds the -rank of each of the k anchors selected using the Top-1 algorithm.

Lemma 3 (Bounded rank). Given $\delta > 0$, for each $i \in [k]$, if the Top-1 algorithm succeeds in finding an $\epsilon/3$-best item a_i in S_i, then the $\epsilon/3$-rank of a_i is bounded as

$$\text{rank}_{\epsilon/3}(a_i) \le k \log(k/\delta);$$

with probability at least $1 - \delta$.

Proof. For the set S_i in the partition, consider the "true" best item $a_i^* \in S_i$, i.e. the item with the best rank in S_i. We begin by claiming that for all $i \in [k]$; $\text{rank}(a_i^*) \le k \log(k/\delta)$ with probability at least $1 - \delta$. To see this, consider a thought experiment of assigning items to these partitions sequentially in order of rank. Each item is assigned to one of the sets in the partition uniformly at random. For a particular partition S_i, $\text{rank}(a_i^*) > k \log(k/\delta)$ can only happen if no element was assigned to S_i from the first $k \log(k/\delta)$ items. The probability of this event is bounded by $(1 - 1/k)^{k \log(k/\delta)} \le \delta/k$. Taking a union bound over all partition gives us our claimed bound on the rank of the "true" best element in every partition.

Assuming every run of the Top-1 algorithm succeeds in identifying an $\epsilon/3$-best element a_i from their corresponding input set S_i, we can further claim that for all $i \in [k]$; $\text{rank}_{\epsilon/3}(a_i) \le k \log(k/\delta)$. This follows by definition of SST, and the fact that a_i is an $\epsilon/3$-best item in S_i. To see this, consider any item $b \in [n]$ for whom $\text{rank}(a_i) > \text{rank}(b) > \text{rank}(a_i^*)$ where $\text{rank}(i)$ represents the rank of i in the sorted order with 1 being the rank of the best item. We must have that $P_{a_i^* a_i} \ge \max\{P_{a_i^* b}, P_{b a_i}\}$. However, we must have that $P_{a_i^* a_i} < 1/2 + \epsilon/3$, which gives us that $P_{b a_i} < 1/2 + \epsilon/3$. Therefore, the only elements in $[n]$ that can beat element a_i with probability at least $1/2 + \epsilon/3$ are the ones whose rank is smaller than $\text{rank}(a_i^*)$. Therefore, $\text{rank}_{\epsilon/3}(a_i) \le \text{rank}(a_i^*) \le k \log(k/\delta)$. □

The above lemma guarantees that the number of elements in $[n]$ that can beat any anchor a_i with a margin of at least $\epsilon/3$ is at most $k \log(k/\delta)$. The next lemma will show that the set of items that are returned by each call to the Eliminate subroutine is smaller than $k \log(k/\delta)$.

Lemma 4 (Eliminate Correctness). Given set $S \subseteq [n]$, rounds $r \ge 1$, anchor $a \in [n]$, confidence $\delta > 0$, accuracy $\epsilon > 0$ and precision $0 < \epsilon' < \epsilon$, let $S^* = \{i \in S : P_{ia} \ge \frac{1}{2} + \frac{\epsilon}{2} + \epsilon'\}$, $S_{bad} = \{i \in S : P_{ia} < \frac{1}{2} + \frac{\epsilon}{2} - \epsilon'\}$, $k' \ge |S \setminus S_{bad}|$ and $m = |S|$. The Eliminate subroutine given in Algorithm 5 returns a set S' such that $S^* \subseteq S'$ and $S' \cap S_{bad} = \emptyset$. The algorithm succeeds with probability at least $1 - \delta$ and uses at most r rounds of adaptivity and makes at most $\frac{10m}{\epsilon'^2}\left(\log^{(r)}(m) + \log(8k'/\delta)\right)$ comparisons.

Proof. The proof of this lemma is very similar to the proof of correctness for the Aggressive-Elimination algorithm (Agarwal et al., 2017), and we only provide it here for the sake of completeness. Given a target number of rounds r, the Eliminate algorithm clearly uses at most r rounds of adaptivity. We first start with the following claim:

Claim 1. For any round $r \ge 1$, and any item $i \in S$, $\Pr\left(|\hat{P}_{ia} - P_{ia}| \ge \epsilon'\right) \le \frac{\delta}{4k' \log^{(r-1)}(m)}$.

Proof. By Hoeffding's inequality, we have,

$$\Pr\left(|\hat{P}_{ia} - P_{ia}| \ge \epsilon'\right) \le 2\exp\left(-2\epsilon'^2 t_r\right) \le 2\exp\left(-\log^{(r)}(m) - \log(8k'/\delta)\right) \le \frac{\delta}{4k' \log^{(r-1)}(m)}$$

as $\log^{(r)}(m) = \log\log^{(r-1)}(m)$. □

The proof of correctness of the algorithm is by induction on the number of rounds r. In the following, we use $\mathcal{A}_r$ to denote Algorithm 5 with input number of rounds r.

Base case: For $r = 1$, Claim 1 ensures that for any $i \in S$, $\Pr\left(|\widehat{P}_{ia} - P_{ia}| \geq \varepsilon\right) \leq \frac{\delta}{4k\log^{(0)}(m)} \leq \frac{\delta}{m}$ as $\log^{(r-1)}(m) = \log^{(0)}(m) = m$ by definition. By taking a union bound over all m items, we obtain that w.p. $1-\delta$, simultaneously for all items $i \in S$, $|\widehat{P}_{ia} - P_{ia}| < \varepsilon$. Then, for any arm $i \in S$ such that $P_{ia} \geq \frac{1}{2} + \gamma + \varepsilon$, we will have that $\widehat{P}_{ia} \geq \frac{1}{2} + \gamma$. Hence, the set S' will contain all items from S^*. On the other hand, for any item $i \in S$ such that $P_{ia} < \frac{1}{2} + \gamma - \varepsilon$ we will have that $\widehat{P}_{ia} < \frac{1}{2} + \gamma$. Hence, no arm from S_{bad} will be included in S'. This proves the base case.

Induction step: Suppose the lemma is true for all number of rounds smaller than $r \leq \log^*(m) - 3$ and we prove it for the case of r rounds, i.e., for $\mathcal{A}_r$.

Let S_r be the set that is given as input to $\mathcal{A}_r$. Let $I = \left\{ i \in S^* : \widehat{P}_{ia} < P_{ia} - \varepsilon \right\}$ and $J = \{ j \in S_{bad} : \widehat{P}_{ja} > P_{ja} + \varepsilon \}$. We know that for all $i \in S^*$ and $j \in S_{bad}$, $P_{ia} - P_{ja} \geq 2\varepsilon$. As the algorithm identifies a set of $m_{r-1} \leq k' + \frac{m_r}{\log^{(r-1)}(m_r)}$ items to recurse upon, we have,

$$\Pr(\mathcal{A}_r \text{ errs}) \leq \Pr(|I| > 0) + \Pr\left(|J| > \frac{m_r}{\log^{(r-1)}(m_r)}\right) + \Pr(\mathcal{A}_{r-1} \text{ errs} \mid \mathcal{E}) \tag{1}$$

where $\mathcal{E}$ denotes the event that $|I| = 0$ and $|J| \leq \frac{m_r}{\log^{(r-1)}(m_r)}$, i.e., the complement of the first two events above.

In the following, we bound probability of each event above. We first have,

$$\Pr(|I| > 0) \leq \sum_{i \in S^*} \Pr\left(\widehat{P}_{ia} < P_{ia} - \varepsilon\right) \overset{\text{Claim 1}}{\leq} k' \cdot \frac{\delta}{4k' \log^{(r-1)}(m_r)} \leq \frac{\delta}{4} \tag{2}$$

where the last inequality is true because $\log^{(r-1)}(m_r) \geq 1$.

We next bound the probability that $|J| > \frac{m_r}{\log^{(r-1)}(m_r)}$. For all $j \in S_{bad}$, we define an indicator random variable Y_j which is 1 iff $\widehat{P}_{ja} > P_{ja} + \varepsilon$. We further define $Y := \sum_j Y_j$. We have,

$$\mathbb{E}[Y] = \sum_j \mathbb{E}[Y_j] = \sum_j \Pr\left(\widehat{P}_{ja} > P_{ja} + \varepsilon\right) \overset{\text{Claim 1}}{\leq} \sum_j \frac{\delta}{4k' \log^{(r-1)}(m_r)} \leq \frac{\delta \cdot m_r}{4\log^{(r-1)}(m_r)}$$

Notice that $Y = |J|$; hence,

$$\Pr\left(|J| > \frac{m_r}{\log^{(r-1)}(m_r)}\right) \leq \Pr\left(Y > \frac{4}{\delta} \mathbb{E}[Y]\right) \leq \frac{\delta}{4} \tag{3}$$

where the last inequality is by Markov bound.

We calculate the probability of error of $\mathcal{A}_{r-1}$ conditioned on that none of the two events above happens (i.e., the event $\mathcal{E}$). In that case, we have $S^* \subseteq S_{r-1}$ and $|S_{r-1} \cap S_{bad}| \leq \frac{m_r}{\log^{(r-1)}(m_r)}$. As $r \leq \log^*(m_r) - 3$ (by the lemma statement), we have $r - 1 \leq (\log^*(m_r) - 1) - 3 \leq \log^*(\log m_r) - 3 \leq \log^*(m_{r-1}) - 3$. Therefore, the input to $\mathcal{A}_{r-1}$ satisfies the assumptions in the lemma statement as well and since the confidence parameter for $\mathcal{A}_{r-1}$ is $\delta/2$, we obtain that $\Pr(\mathcal{A}_{r-1} \text{ errs} \mid \mathcal{E}) \leq \delta/2$. By plugging in this bound, together with Eq (2) and Eq (3) to Eq (1), we obtain that $\mathcal{A}_r$ is also a δ-error algorithm, finalizing the proof of induction step.

Next, the final step is to prove an upper bound on the query complexity of $\mathcal{A}_r$ for any $r \geq 1$. The proof is again by induction on the number of rounds r. The base case of $r = 1$ is trivially true. Now suppose the bounds are true for all integers smaller than $r \leq \log^*(m) - 3$ and we prove the lemma for the case of r rounds, i.e., for $\mathcal{A}_r$. Note that the total number of comparisons in $\mathcal{A}_r$ is the sum of comparisons in step 3 (which is $m \cdot t_r$) and the comparisons in the recursive call which we bound below. For the recursive call there are two cases to consider depending on which of step 12 (Case 1) or step 14 (Case 2) in Algorithm 5 is being executed.

Case 1: In this case A_1 is called with the confidence parameter $=2$ on at most $2k^0$ items. We do not use the induction hypothesis here and instead argue directly that,

$$
\begin{aligned}
\mathrm{cost}(A_r) &= m\, t_r + \mathrm{cost}(A_1) \\
& m\, t_r + \frac{4k^0}{"^2}(\log(2k^0) + \log(16k^0=)) \\
& m\, t_r + \frac{8k^0}{"^2}\log(8k^0=) \\
& m\, t_r + \frac{8m}{"^2}\log(8k^0=) && (\text{as } k^0 \ m) \\
&= \frac{2m}{"^2}\left(\log^{(r)}(m) + \log(8k^0=)\right) + \frac{8m}{"^2}\log(8k^0=) < && (\text{by plugging in the value of } t_r) \\
& \frac{10m}{"^2}\left(\log^{(r)}(m) + \log(8k^0=)\right)
\end{aligned}
$$

which proves the induction step in this case.

Case 2: In this case, A_{r-1} is called with the confidence parameter $=2$ on at most $\frac{2m}{\log^{(r-1)}(m)}$ items. Hence, by induction, the total number of comparisons made in recursive calls is

$$
\begin{aligned}
\mathrm{cost}(A_r) &= m\, t_r + \mathrm{cost}(A_{r-1}) \\
& m\, t_r + \frac{20m}{"^2 \log^{(r-1)}(m)}\left(\log^{(r-1)}(2m) + \log(16k^0=)\right) \\
& m\, t_r + \frac{20m}{"^2 \log^{(r-1)}(m)}\left(\log^{(r-1)}(m) + 1 + \log(8k^0=) + 1\right) \\
&< m\, t_r + \frac{20m}{"^2} + \frac{22m \log(8k^0=)}{"^2 \log^{(r-1)}(m)} \\
&< \frac{2m}{"^2}\left(\log^{(r)}(m) + \log(8k^0=)\right) + \frac{8m \log^{(r)}(m)}{"^2} + \frac{8m \log(8k^0=)}{"^2}
\end{aligned}
$$

where in the last inequality we used the bound on t_r plus the fact that $\log^{(r)}(m)\ \ 16$ as $r\ \ \log(m)\ \ 3$. This concludes the proof of Lemma 4. □

Proof. (of Theorem 4) We will first show that our algorithm will return a $(;k)$-optimal solution with probability at least 1 . Using the correctness of the Top-1 algorithm (Theorem 7) and the union bound, we can argue that we will find the $=3$-best item a_i for each S_i, with probability at least $1\ \ =4$. Using Lemma 3, for each $i\ 2\ [k]$, the $=3$-rank of a_i is at most $k\log(4k=)$ with probability at least $1\ \ =4$. For $i\ 2\ [k]$, since $\mathrm{rank}_{=3}(a_i)\ \ k\log(4k=)$, the size of $jS\ n\ S_{bad}j$ is at most $k\log(4k=)$ for the i-th call to Eliminate which fulfills the requirement for k in Lemma 4. Hence, using Lemma 4, w.p. $1\ \ =4$, the Eliminate algorithm succeeds for each of the k calls.

Now, we show that an $(;k)$-optimal solution is contained in the set $A = fa_ig_{i=1}^k$ of anchors and the surviving items W. In order to see this, consider the "worst" anchor $a^w\ 2\ A$, i.e. the anchor with the worst rank. Then we have by property of our algorithm that the set W must contain all items that beat this worst anchor a^w with a margin of at least . Hence, any item $i\ 2\ W$ is of higher rank than any other item $j\ 2\ W$ as otherwise j would also have to be contained in W. If $jWj\ \ k$ then it is easy to see that the exact top-k items are a subset of W, and we can find a PAC top-k solution with probability at least $1\ \ =4$ by comparing all items in W a sufficient number of times. Moreover, for any item that is excluded from the set W, it must be that this item cannot beat any of the k anchors with margin of at least $=3$, therefore making any anchor a valid substitute for the rejected item. Hence, if $jWj < k$ then we can output the set W along with any $k\ \ jWj$ anchors in $A\ n\ W$ chosen arbitrarily. By uniqueness of the anchors, we are guaranteed that there are at least $k\ \ jWj$ anchors in $A\ n\ W$. This will constitute a valid $(;k)$-optimal solution. Moreover, using the union bound the probability of failure is at most .

Finally, we need to prove a bound on the number of comparisons. The k calls to the Top-1 algorithm take $O(\frac{nk(\log^{(r)}(n)+\log(k=))}{2})$ comparisons in total. The k calls to Eliminate also take $O(\frac{nk(\log^{(r)}(n)+\log(k=))}{2})$

comparisons in total. Since, the size of W is at most k log(4k=), comparing all items in W against each other also takes $O(\frac{k^2 \log^3(k=)}{2})$. After summing these, we get the final bound on query complexity.

□

F An Algorithm for PAC Top-1 Identification

In this section we will present an algorithm for top-1 identification that is used as a subroutine in our top-k algorithm in Section E. This algorithm is similar to the algorithm of Cohen-Addad et al. (2020) and is again based on the idea of finding a good 'anchor' arm and filtering items based on this anchor. Precisely, we chose a uniformly at random set S of size roughly $n^{2=3}$ and find an =3-best item in this set using our 2-round algorithm. Similar, to the analysis of our top-k algorithm in Section E, we can show that this item has =3-rank of $O(n^{1=3})$. We then use the Eliminate algorithm discussed in Section E to find all items that are -better than a, and exclude all items that are =3-close. Since, there are not more than $O(n^{1=3})$ such items we can compare them against each other in order to find an -best item.

Algorithm 6 (;)-PAC top-1

Input: items [n], rounds r + 4, accuracy , confidence
Sample a set S by including each i 2 [n] in S with probability $1=n^{1=3}$
Call the 2-round algorithm (Algorithm 1) over S to find an (=3; =4)-PAC top-1 item a in S
W Eliminate([n]; r; a; =4; =2; =6; 1) (Algorithm 5)
Compare all pairs in W, $O(\log(1=)=^2)$ times.
Output: If jWj 1 then output an (; 1)-optimal item in W, else output a

We now give a proof of correctness for this algorithm.

Proof. (of Theorem 7) We begin by showing that $\text{rank}_{=3}(a) = n^{1=3} \log(4=)$ w.p. 1 =4. Firstly, observe that the best item in S has rank at most $n^{1=3} \log(4=)$ w.p. 1 =4. For the sake of contradiction, suppose that this is not true, i.e. no item from the top $n^{1=3} \log(4=)$ items makes it into S. The probability of this is bounded as $(1 \quad 1=n^{1=3})^{n^{1=3}\log(4=)}$ =4 which leads to a contradiction. Secondly, using the correctness of our 2-round algorithm (Theorem 2), we can argue that item a is =3-close to the best item in S w.p. 1 =4. Finally, using an argument similar to Lemma 3 we can show that a being =3-close to the best item in S implies that $\text{rank}_{=3}(a) = n^{1=3} \log(4=)$.

Now, using the correctness of Eliminate (Lemma 4), w.p. 1 =4 the set W is such that any item -better than a is returned and any item i with $P_{ia} < \frac{1}{2} + =3$ is excluded. This implies that the size of W is at most $n^{1=3} \log(4=)$. Moreover, if jWj 1 then the true best item is contained in w as otherwise it would lead to a contradiction. Hence, in this case we can find an (; 1)-optimal item in W w.p. 1 =5 using sufficient number of comparisons. If jWj = 0 then item a is a valid solution as there is no other item that is -better than a. Hence, we can find a valid solution in both cases. Moreover, using the union bound this happens w.p. 1 .

Now, we count the number of comparisons used by the algorithm. Note that using the Hoeffding's concentration inequality, the number of items in S is at most $2n^{2=3}$ with very high probability. Hence, the 2-round algorithm over S takes at most $O(n^{8=9} \log^2(n=)=^2) = O(n \log(1=)=^2)$ comparisons. The Eliminate subroutine takes $O(\frac{n(\log^{(r)}(n)+\log(1=))}{2})$ comparisons. Since, the size of W is at most $n^{1=3} \log(4=)$, comparing all items in W against each other also takes $O(\frac{n^{2=3} \log^3(1=)}{2}) = O(n \log(1=)=^2)$ comparisons. After summing these, we get the final bound on query complexity.

□