Secretary Problems with Non-Uniform Arrival Order

Thomas Kesselheim[*] Robert Kleinberg[†] Rad Niazadeh[†]

Abstract

For a number of problems in the theory of online algorithms, it is known that the assumption that elements arrive in uniformly random order enables the design of algorithms with much better performance guarantees than under worst-case assumptions. The quientessential example of this phenomenon is the secretary problem, in which an algorithm attempts to stop a sequence at the moment it observes the maximum value in the sequence. As is well known, if the sequence is presented in uniformly random order there is an algorithm that succeeds with probability $1/e$, whereas no non-trivial performance guarantee is possible if the elements arrive in worst-case order.

In many of the applications of online algorithms, it is reasonable to assume there is some randomness in the input sequence, but unreasonable to assume that the arrival ordering is uniformly random. This work initiates an investigation into relaxations of the random-ordering hypothesis in online algorithms, by focusing on the secretary problem and asking what performance guarantees one can prove under relaxed assumptions. Toward this end, we present two sets of properties of distributions over permutations as sufficient conditions, called the (p, q, δ)-*block-independence property* and (k, δ)-*uniform-induced-ordering property*. We show these two are asymptotically equivalent by borrowing some techniques from the celebrated *approximation theory*. Moreover, we show they both imply the existence of secretary algorithms with constant probability of correct selection, approaching the optimal constant $1/e$ as the related parameters of the property tend towards their extreme values. Both of these properties are significantly weaker than the usual assumption of uniform randomness; we substantiate this by providing several constructions of distributions that satisfy (p, q, δ)-block-independence. As one application of our investigation, we prove that $\Theta(\log \log n)$ is the minimum entropy of any permutation distribution that permits constant probability of correct selection in the secretary problem with n elements. While our block-independence condition is sufficient for constant probability of correct selection, it is not necessary; however, we present complexity-theoretic evidence that no simple necessary and sufficient criterion exists. Finally, we explore the extent to which the performance guarantees of other algorithms are preserved when one relaxes the uniform random ordering assumption to (p, q, δ)-block-independence, obtaining a positive result for Kleinberg's multiple-choice secretary algorithm and a negative result for the weighted bipartite matching algorithm of Korula and Pál.

[*]Max-Planck-Institut für Informatik, Campus E1 4, 66123 Saarbrücken, Germany, E-Mail: thomas.kesselheim@mpi-inf.mpg.de.

[†]Department of Computer Science, Cornell University, Gates Hall, Ithaca, NY 14853, USA, E-Mail: {rdk, rad}@cs.cornell.edu.

1 Introduction

A recurring theme in the theory of online algorithms is that algorithms may perform much better when their input is in (uniformly) random order than when the ordering is worst-case. The quientessential example of this phenomenon is the secretary problem, in which an algorithm attempts to stop a sequence at the moment it observes the maximum value in the sequence. As is well known, if the sequence is presented in uniformly random order there is an algorithm that succeeds with probability $\frac{1}{e}$, whereas no non-trivial performance guarantee is possible if the elements arrive in worst-case order.

In many of the applications of online algorithms, it is reasonable to assume there is some randomness in the input sequence, but unreasonable to assume that the input ordering is *uniformly* random. It is therefore of interest to ask which algorithms have *robust* performance guarantees, in the sense that the performance guarantee holds not only when the input order is drawn from the uniform distribution, but whenever the input order is drawn from a reasonably broad family of distributions that includes the uniform one. In other words, we seek relaxations of the standard random-ordering hypothesis which are weak enough to include many distributions of interest, but strong enough to enable one to prove the same (or qualitatively similar) performance guarantees for online algorithms.

This work initiates an investigation into relaxations of the random-ordering hypothesis in online algorithms, by focusing on the secretary problem and asking what performance guarantees one can prove under relaxed assumptions. In the problems we consider there are three parties: an *adversary* that assigns values to items, *nature* which permutes the items into a random order, and an *algorithm* that observes the items and their values in the order specified by nature. To state our results, let us say that a distribution over permutations, is *secretary-admissible* (abbreviated *s-admissible*) if it is the case that when nature uses this distribution to sample the ordering of items, there exists an algorithm that is guaranteed at least a constant probability of selecting the element of maximum value, no matter what values the adversary assigns to elements. If this constant probability approaches $\frac{1}{e}$ as the number of elements, n, goes to infinity, we say that the distribution is *secretary-optimal (s-optimal)*.

> **Question 1:** *What natural properties of a distribution suffice to guarantee that it is s-admissible? What properties suffice to guarantee that it is s-optimal?*

For example, rather than assuming that ordering of the entire n-tuple of items is uniformly random, suppose we fix a constant k and assume that for every k-tuple of distinct items, the relative order in which they appear in the input sequence is δ-close to uniform. Does this imply that the distribution is s-admissible? In §2 we formalize this (k, δ)-*uniform-induced-ordering property (UIOP)*, and we prove that it implies s-admissibility for $k \geq 3$ and approaches s-optimality as $k \to \infty$ and $\delta \to 0$. To prove this, we relate the uniform-induced-ordering property to another property, the (p, q, δ)-*block-independence property (BIP)*, which may be of independent interest. Roughly speaking, the block-independence property asserts that the joint distribution of arrival times of any p distinct elements, when considered at coarse enough granularity, is δ-close to p i.i.d. samples from the uniform distribution. While this property may sound much stronger than the UIOP, we show that it is actually implied by the UIOP for sufficiently large k and small δ.

To substantiate the notion that these properties are satisfied by many interesting distributions that are far from uniform, we show that they apply to several natural families of permutation distributions, including almost every uniform distribution with support size $\omega(\log n)$, and the distribution over linear orderings defined by taking any n sufficiently "incoherent" vectors and projecting them onto a random line.

A distinct but related topic in the theory of computing is pseudorandomness, which shares a similar emphasis on showing that performance guarantees of certain classes of algorithms are preserved when one replaces the uniform distribution over inputs with suitably chosen non-uniform distributions, specifically those having low entropy. While our interest in s-admissibility and the (k, δ)-UIOP is primarily motivated by the considerations of robustness articulated earlier, the analogy with pseudorandomness prompts a natural set of questions.

> **Question 2:** *What is the minimum entropy of an s-admissible distribution? What is the minimum entropy of a distribution that satisfies the (k, δ)-UIOP? Is there an explicit construction that achieves the minimum entropy?*

In §2 and §3 we supply matching upper and lower bounds to answer the first two questions. The answer is the same in both cases, and it is surprisingly small: $\Theta(\log \log n)$ bits. Moreover, $\Theta(\log \log n)$ bits suffice not just for s-admissibility, but for s-optimality! We also supply an explicit construction, using Reed-Solomon codes, of distributions with $\Theta(\log \log n)$ bits of entropy that satisfy all of these properties.

Given that the (k, δ)-UIOP is a sufficient condition for s-admissibility, that it is satisfied in every natural construction of s-admissible distributions that we know of, and that the minimum entropy of (k, δ)-UIOP distributions matches the minimum entropy of s-admissible distributions, it is tempting to hypothesize that the (k, δ)-UIOP (or something very similar) is both necessary and sufficient for s-admissibility.

> **Question 3:** *Find a natural necessary and sufficient condition that characterizes the property of s-admissibility.*

In §4 we show that, unfortunately, this is probably impossible. We construct a strange distribution over input orderings that is s-admissible, but any algorithm achieving constant probability of correct selection must use a stopping rule that cannot be computed by circuits of size $2^{n/\log^2(n)}$. The construction makes use of a coding-theoretic construction that may be of independent interest: a binary error-correcting code of block length n and message length $m = o(n)$, such that if one erases any $n - 2m$ symbols of the received vector, most messages can still be uniquely decoded even if $\Omega(m)$ of the remaining $2m$ symbols are adversarially corrupted.

Finally, we broaden our scope and consider other online problems with randomly-ordered inputs.

> **Question 4:** *Are the performance guarantees of other online algorithms in the uniform-random-order model (approximately) preserved when one relaxes the assumption about the input order to the (k, δ)-UIOP or the (p, q, δ)-BIP? If the performance guarantee is not always preserved in general, what additional properties of an algorithm suffice to ensure that its performance guarantee is preserved?*

This is an open-ended question, but we take some initial steps toward answering it by looking at two generalizations of the secretary problem: the multiple-choice secretary problem (a.k.a. the uniform matroid secretary problem) and the online bipartite weighted matching problem. We show that the algorithm of Kleinberg [25] for the former problem preserves its performance guarantee, and the algorithm of Korula and Pál [26] for the latter problem does not.

Related Work. The secretary problem was solved by Lindley [28] and Dynkin [15]. A sequence of papers relating secretary problems to online mechanism design [20, 25, 5] touched off a flurry of CS research during the past 10 years. Much of this research has focused on the so-called *matroid secretary problem*, which remains unsolved despite a string of breakthroughs including a recent pair of $O(\log \log r)$-competitive algorithms [27, 18], where r is the matroid rank. Generalizations are known for weighted matchings in graphs and hypergraphs [14, 23, 26], independent sets [19], knapsack constraints [4], and submodular payoff functions [7, 17], among others. Of particular relevance to our work is the *free order model* [21]; our results on the minimum entropy s-admissible distribution can be regarded as a randomness-efficient secretary algorithm in the free-order model.

The uniform-random-ordering hypothesis has been applied to many other problems in online algorithms, perhaps most visibly to the AdWords problem [12, 16] and its generalizations to online linear programming with packing constraints [2, 13, 24, 32], and online convex programming [1]. Applications of the random-order hypothesis in minimization settings are more rare; see [29, 30] for applications in the context of facility location and network design.

In seeking a middle ground between worst-case and average-case analysis, our work contributes to a broad-based research program going by the name of 'beyond worst-case analysis" [34]. In terms of motivation, there are clear conceptual parallels between our paper and the work of Mitzenmacher and Vadhan [31], who study hashing and identify hypotheses on the data-generating process, much weaker than uniform randomness, under which random hashing using a 2-universal hash family has provably good performance, although at a technical level our paper bears no relation to theirs.

The properties of permutation distributions that we identify in our work bear a resemblance to almost k-wise independent permutations (e.g., [22]), but the (k, δ)-UIOP and (p, q, δ)-BIP are much weaker, and consequently permutation distributions satisfying these properties are much more prevalent than almost k-wise independent permutations.

Setting and Notations. We consider problems in which an algorithm selects one or more elements from a set $\mathcal{U}$ of n items. Items are presented sequentially, and an algorithm may only select items at the time when they are presented. In the *secretary problem* the items are totally ordered by value, and the algorithm is allowed to select only one element of the input sequence, with the goal of selecting the item of maximum value. Algorithms for the secretary problem are assumed to be comparison-based[1], meaning their decision whether to select the item presented at time t must be based only on the relative ordering (by value) of the first t elements that arrived. Algorithms are evaluated according to their *probability of correct selection*, i.e., the probability of selecting the item of maximum value.

We assume that the set $\mathcal{U}$ of items is $[n] = \{1, \ldots, n\}$. The order in which items are presented is then represented by a permutation π of $[n]$, where $\pi(i)$ denotes the position of item i in the input sequence. Similarly, the ordering of items by value can be represented by a permutation σ of $[n]$, where $\sigma(j) = i$ means that the j^{th} largest item is i. Then, the input sequence observed by the algorithm is completely described by the composition $\pi\sigma$.

2 Sufficient Properties of Non-Uniform Probability Distributions

In §1, we introduced two properties of non-uniform probability distributions which suffice to ensure existence of a secretary algorithm with constant probability of correct selection. (In other words, the two properties imply s-admissibility.) We begin by formally defining these two properties.

Definition 1. A distribution $\underline{\pi}$ over permutations of $[n]$ satisfies the (k, δ)-uniform-induced-ordering property, abbreviated (k, δ)-UIOP, if and only if, for every k distinct items $x_1, \ldots, x_k \in [n]$, if π is a random sample from $\underline{\pi}$ then $\mathbf{Pr}\left[\pi(x_1) < \pi(x_2) < \cdots < \pi(x_k)\right] \geq (1 - \delta)\frac{1}{k!}$.

[1]This assumption of comparison-based algorithms is standard in the literature on secretary problems. Samuels [35] proved that when the input order is uniformly random, it is impossible to achieve probability of correct selection $1/e + \varepsilon$ for any constant $\varepsilon > 0$, even if the algorithm is allowed to observe the values.

The (k, δ)-uniform-induced-ordering property is a very natural assumption and it is rather easy to show that it is fulfilled by a probability distribution. We will demonstrate this with a few examples in §2.3. However, it is not clear how to analyze algorithms for secretary problems based on this property. To this end, the more technical (p, q, δ)-block-independence property is more helpful. We show this by analyzing the classic algorithm for the secretary problem in Section 2.1 and the k-uniform matroid secretary problem in Section 5. However, one of our main results in Section 2.2 is that these two properties are in fact equivalent, in the limit as the parameters $k, p, q \to \infty$ and $\delta \to 0$.

Definition 2. Given a positive integer $q \leq n$, partition $[n]$ into q consecutive disjoint blocks of size between $\lfloor n/q \rfloor$ and $\lceil n/q \rceil$ each, denoted by $B_1, \ldots, B_q \subseteq [n]$. A permutation distribution $\underline{\pi}$ satisfies the (p, q, δ)-block-independence property, abbreviated (p, q, δ)-BIP, if for any distinct $x_1, \ldots, x_p \in [n]$, and any $b_1 \ldots, b_p \in [q]$

$$\mathbf{Pr}\left[\bigwedge_{j \in [p]} \pi(x_i) \in B_{b_i}\right] \geq (1-\delta)\left(\frac{1}{q}\right)^p \quad ,$$

Note that $b_1 \ldots, b_p$ do not necessarily have to be distinct. To simplify notation, given a permutation π of $[n]$, we define a function $\pi^B \colon \mathcal{U} \to [q]$ by setting $\pi^B(x) = i$ if and only if $\pi(x) \in B_i$ for all $x \in \mathcal{U}$.

2.1 Secretary Algorithms and the (p, q, δ)-block-independence property

Next, we will analyze the standard threshold algorithm for the secretary problem under probability distributions that only fulfill the (p, q, δ)-block-independence property rather than being uniform. The algorithm only observes the first $\frac{n}{\mathrm{e}}$ items. Afterwards, it accepts the first item whose value exceeds all values seen up to this point. Under a uniform distribution, this algorithm picks the best items with probability at least $\frac{1}{\mathrm{e}} - o(1)$. We show that already for small constant values of p and q and rather large constant values of δ this algorithm has constant success probability. At the same time, for large p and q and small δ, the probability converges to $\frac{1}{\mathrm{e}}$.

Theorem 1. *Under a (p, q, δ)-block-independent probability distribution, the standard secretary algorithm picks the best item with probability at least $\frac{1}{\mathrm{e}} - \frac{\mathrm{e}+1}{q} - \delta - \left(1 - \frac{1}{\mathrm{e}}\right)^{p-1}$.*

Proof Sketch. Let $T = \lfloor \frac{q}{\mathrm{e}} \rfloor$ denote the index of the block in which the threshold is located. Furthermore, let $x_j \in \mathcal{U}$ be the jth best item. We condition on the event that x_1 comes in block with index i. To ensure that our algorithm picks this item, it suffices that x_2 comes in blocks $1, \ldots, T-1$. Alternatively, we also pick x_1 if the x_2 comes in blocks $i+1, \ldots, q$ and x_3 comes in blocks $1, \ldots, T-1$. Continuing this argument, we get

$$\mathbf{Pr}\left[\text{correct selection}\right] \geq \sum_{i=T+1}^{q} \sum_{j=2}^{p} \mathbf{Pr}\left[\pi^B(x_1) = i, \pi^B(x_2), \ldots, \pi^B(x_{j-1}) > i, \pi^B(x_j) < T\right] \quad .$$

Note that the (p, q, δ)-BIP implies the (p', q, δ)-BIP for any $p' < p$, simply by marginalizing over the remaining indices in the tuple. This gives us:

$$\mathbf{Pr}\left[\text{correct selection}\right] \geq \sum_{i=T+1}^{q} \sum_{j=2}^{p} (1-\delta) \frac{1}{q} \left(\frac{q-i}{q}\right)^{j-2} \frac{T-1}{q} \quad ,$$

and the lemma follows after manipulating the expression on the right side and applying some standard bounds. □

2.2 Relationship Between the Two Properties

We will show that the two properties defined in the preceding section are in some sense equivalent in the limit as the parameters $k, p, q \to \infty$ and $\delta \to 0$. (For $k = 2$, a distribution satisfying (k, δ)-UIOP is not even necessarily s-admissible—this is an easy consequence of the lower bound in §3 and the fact that the $(2, 0)$-UIOP is achieved by a distribution with support size 2, that uniformly randomizes between a single permutation and its reverse. Already for $k = 3$ and any constant $\delta < 1$, the (k, δ)-UIOP implies s-admissibility; this is shown in Appendix A.)

Our first result is relatively straightforward: Any probability distribution that fulfills the (p, q, δ)-BIP also fulfills the $(p, \delta + \frac{q^2}{p})$-UIOP. The (easy) proof is deferred to Appendix B.1.2.

Theorem 2. *If a distribution over permutation fulfills the (p, q, δ)-BIP, then it also fulfills the $(p, \delta + \frac{p^2}{q})$-UIOP.*

The other direction is far less obvious. Observe that the (k, δ)-uniform-induced-ordering property works in a purely local sense: even for a single item $\xi \in \mathcal{U}$, the distribution of its position $\pi(x)$ can be far from uniform. For example, the case $k = 2$ is even fulfilled by a two-point distribution that only include one permutation and its reverse. Then $\pi(x)$ can only attain two different values. Nevertheless, we have the following result.

Theorem 3. *If a distribution over permutation fulfills the (k, δ)-uniform-induced-ordering property, then it also satisfies (p, q, δ)-block-independence property for $p = o(k^{\frac{1}{5}})$, $q = O(k^{\frac{1}{5}})$ as k goes to infinity.*

The proof (again in Appendix B.1.2) applies the *theory of approximation of functions*, which addresses the question of how well one can approximate arbitrary functions by polynomials. The main insight underlying the proof is the following. If $\underline{\pi}$ satisfies the (k, δ)-UIOP, then for any k-tuple of distinct elements $x_1, \ldots, x_k$ if one defines random variables $X_i \triangleq \pi(x_i)/n$, then the expected value of any monomial of total degree $k/2$ in the variables $\{X_i\}$ approximates the expected value of that same monomial under the distribution of a uniformly-random permutation (Lemma 5). With this lemma in hand, proving Theorem 3 becomes a matter of quantifying how well the indicator function of a (multi-dimensional) rectangle can be approximated by low-degree polynomials. Approximation theory furnishes such estimates readily.

2.3 Constructions of Probability Distributions Implying the Properties

2.3.1 Randomized One-Dimensional Projections

In this section we present one natural construction leading to a distribution that satisfies the (k, δ)-UIOP. The starting point for the construction is an n-tuple of vectors $x_1, \ldots, x_n \in \mathbb{R}^d$. If one sorts these vectors according to a random one-dimensional projection (i.e., ranks the vectors in increasing order of $w \cdot x_i$, for a random w drawn from a spherically symmetric distribution), when does the resulting random ordering satisfy the (k, δ)-UIOP? In Appendix B.1.3 we recall the definition of the restricted isometry property (RIP) from [9], and we prove the following.

Theorem 4. *Let $x_1, \ldots, x_n$ be the columns of a matrix that satisfies the RIP of order k with restricted isometry constant $\delta_k = \frac{\delta}{k}$. If w is drawn at random from a spherically symmetric distribution and we use w to define a permutation of $[n]$ by sorting its elements in order of increasing $w \cdot x_i$, the resulting distribution over S_n satisfies the (k, δ)-UIOP.*

2.3.2 Constructions with Low Entropy

This subsection presents two constructions showing that there exist permutation distributions with entropy $\Theta(\log\log n)$ satisfying the (k,δ)-UIOP for arbitrarily large constant k and arbitrarily small constant δ. The proofs of both results are in Appendix B.1.4. The proof of the first result is an easy application of the probabilistic method. The proof of the second result uses Reed-Solomon codes to supply an explicit construction.

Theorem 5. *Fix some $\xi \geq \frac{2(k+1)!}{\delta^2}\ln n$. If S is a random ξ-element multiset of permutations $\pi\colon [n] \to [n]$, then the uniform distribution over S fulfills the (k,δ)-UIOP with probability at least $1-\frac{1}{n}$.*

Theorem 6. *There is a distribution over permutations that has entropy $O(\log\log n)$ and fulfills the (k,δ)-uniform-induced-ordering property where $\delta = O(\frac{k^2}{\log\log\log n})$.*

3 Tight Bound on Entropy of Distribution

One of the consequences of the previous section is the fact that there are s-admissible—in fact, even s-optimal—distributions with entropy $O(\log\log n)$. In this section, we show that this bound is actually tight. We show that *every* probability distribution of entropy $o(\log\log n)$ is not s-admissible. The crux of the proof lies in defining a notion of "semitone sequences"—sequences which satisfy a property similar to, but weaker than, monotonicity—and showing that an adversary can exploit the existence of long semitone sequences to force every algorithm to have a low probability of success.

Theorem 7. *A permutation distribution $\underline{\pi}$ of entropy $H = o(\log\log n)$ cannot be s-admissible.*

The full proof of this theorem can be found in Appendix B.2. We use the fact for distributions of entropy H there is a subset of the support of size k that is selected with probability at least $1-\frac{8H}{\log(k-3)}$. It then suffices to show that if the distribution's support size is at most k, then any algorithm's probability of success against a worst-case adversary is at most $\frac{k+1}{\log n}$. The theorem then follows by setting $k=\sqrt{\log n}$. To bound the algorithm's probability of success, we the notion of *semitone sequences*, defined recursively as follows: an empty sequence is semitone with respect to any permutation π, and a sequence $(x_1,\ldots,x_s)$ is *semitone* w.r.t. π if $\pi(x_s) \in \{\min_{i\in[s]}\pi(x_i), \max_{i\in[s]}\pi(x_i)\}$ and $(x_1,\ldots,x_{s-1})$ is semitone w.r.t. π. We will show that given k arbitrary permutations of $[n]$, there is always a sequence of length $\frac{\log n}{k+1}$ that is semitone with respect to all k permutations. Later on, we show how an adversary can exploit this sequence to make any algorithm's success probability small.

Lemma 1. *Suppose $\Pi = \{\pi_1,\ldots,\pi_k\}$, where each π_i is a permutation over $\mathcal{U}$. Then there exists a sequence $(x_1,\ldots,x_s)$ that is semitone with respect to each π_i and $s=\frac{\log n}{k+1}$.*

Proof. For a fixed permutation π_i and a fixed item $y\in\mathcal{U}$, we define a function $h_i^y\colon \mathcal{U}\setminus\{y\}\to\{0,1\}$ that indicates whether π_i maps x is to a higher than y or not. Formally,

$$h_i^y(x) = \begin{cases} 0 & \text{if } \pi_i(x) < \pi_i(y) \\ 1 & \text{if } \pi_i(x) > \pi_i(y) \end{cases} .$$

Still keeping one item $y\in\mathcal{U}$ fixed, we now get a k-dimensional vector by concatenating the values for different π_i. This way, we obtain a hash function $\mathbf{h}^y\colon \mathcal{U}\setminus\{y\}\to\{0,1\}^k$, where $\mathbf{h}^y(x) = \big(h_1^y(x),\ldots,h_k^y(x)\big)$.

Starting from $U^{(0)} = \mathcal{U}$, we now construct a sequence of nested subsets $U^{(0)} \supseteq U^{(1)} \supseteq \dots$ iteratively. At iteration $t+1$, given set $U^{(t)} \neq \emptyset$, we do the following. For an arbitrary element x_{s-t} of $U^{(t)}$, we hash each element of $U^{(t1)} \backslash \{x_{s-t}\}$ to a value in $\{0,1\}^k$ by using $\mathbf{h}^{x_{s-t}}$. Now $U^{(t+1)} \subseteq U^{(t)} \backslash \{x_{s-t}\}$ is defined to be the set of occupants of the most occupied hash bucket in $\{0,1\}^k$.

Note that if we place x_{s-t} at the end of any semitone sequence in $U^{(t+1)}$ it will remain semitone with respect to each π_i. This in turn implies that for any t' the sequence $(x_1, \dots, x_{t'})$ is semitone with respect to all π_i.

It now remains to bound the length of the sequence $(x_1, \dots, x_s)$ we are able to generate. We achieve length s if and only if $U^{(s)}$ is the first empty set. At iteration t of the above construction, we have $|U^{(t)}| - 1$ elements to hash and we have 2^k hash buckets, so $|U^{(t+1)}| \geq (|U^{(t)}| - 1)2^{-k} \geq |U^{(t)}|2^{-(k+1)}$ and therefore $|U^{(t)}| \geq 2^{-t(k+1)}|U^{(0)}| = 2^{-t(k+1)}n$. As $|U^{(s)}| < 1$, this implies $2^{-s(k+1)}n < 1$. So $s > \frac{\log n}{k+1}$. □

An adversary can exploit the above semitone sequence and force any algorithm to only have a $\frac{1}{s}$ probability of success. To show this we look at the performance of the best deterministic algorithm against a particular distribution over assignments of values to items.

Lemma 2. *Let $\mathcal{V} = \{1, 2, \dots, s\}$. Assign values from $\mathcal{V}$ to items $(x_1, \dots, x_s)$ at random by*

$$\text{value}(x_s) = \begin{cases} \max(\mathcal{V}) & \text{with probability } 1/s \\ \min(\mathcal{V}) & \text{with probability } 1 - 1/s \end{cases}$$

and then assigning values from $\mathcal{V} \backslash \{\text{value}(x_s)\}$ to items $(x_1, \dots, x_{s-1})$ recursively. Assign a value 0 to all other items.

Consider an arbitrary algorithm following permutation π such that $(x_1, \dots, x_s)$ is semitone with respect to π. This algorithm selects the best item with probability at most $\frac{1}{s}$.

Lemmas 1 and 2 with Yao's principle then imply that any algorithm's probability of success against a worst-case adversary is at most $\frac{k+1}{\log n}$.

4 Easy Distributions Are Hard to Characterize

Which distributions are s-admissible, meaning that they allow an algorithm to achieve constant probability of correct selection in the secretary problem? The results in §2 and §3 inspire hope that the (k, δ)-UIOP, the (p, q, δ)-BIP, or something very similar, is both necessary and sufficient for s-admissibility. Unfortunately, in this section we show that in some sense, it is hopeless to try formulating a comprehensible condition that is both necessary and sufficient. We construct a family of distributions $\underline{\pi}$ with associated algorithms ALG having constant success probability when the items are randomly ordered according to $\underline{\pi}$, but the complicated and unnatural structure of the distribution and algorithm underscore the pointlessness of precisely characterizing s-admissible distributions. In more objective terms, we construct a $\underline{\pi}$ which is s-admissible, yet for any algorithm whose stopping rule is computable by circuits of size less than $2^{n/\log(n)}$, the probability of correct selection is $o(1)$.

Throughout this section (and its corresponding appendix) we will summarize the adversary's assignment of values to items by a permutation σ; the j^{th} largest value is assigned to item $\sigma(j)$. If $\underline{\sigma}$ is any probability distribution over such permutations, we will let $V^{\underline{\pi}}(\text{ALG}, \underline{\sigma})$ denote the probability that ALG makes a correct selection when the adversary samples the value-to-item assignment from $\underline{\sigma}$, and nature independently samples the item-to-time-slot assignment from $\underline{\pi}$. We will also let $V^{\underline{\pi}}(*, \underline{\sigma}) = \max_{\text{ALG}} V^{\underline{\pi}}(\text{ALG}, \underline{\sigma})$, $V^{\underline{\pi}}(\text{ALG}, *) = \min_{\underline{\sigma}} V^{\underline{\pi}}(\text{ALG}, \underline{\sigma})$, and $V^{\underline{\pi}} = \min_{\underline{\sigma}} \max_{\text{ALG}} V^{\underline{\pi}}(\text{ALG}, \underline{\sigma})$. Thus, for example, the property that $\underline{\pi}$ is s-admissible is expressed by the formula $V^{\underline{\pi}} = \Omega(1)$.

As a preview of the techniques underlying our construction, it is instructive to first consider a game against nature in which there is no adversary, and the algorithm is simply trying to pick out the maximum element when items numbered in order of decreasing value arrive in the random order specified by $\underline{\pi}$. This amounts to determining $V^{\underline{\pi}}(*, \iota)$, where ι is the distribution that assigns probability 1 to the identity permutation. Our construction is based on the following intuition. In the secretary problem with uniformly random arrival order, the arrival order of items that arrived before time t is uncorrelated with the order in which items arrive after time t, and so the ordering of past elements is irrelevant to the question of whether to stop at time t. However, there is a great deal of entropy in the ordering of elements that arrived before time t; it encodes $\Theta(t \log t)$ bits of information. We will construct a distribution $\underline{\pi}$ in which this information contained in the ordering of the elements that arrived before time $t = n/2$ fully encodes the time when the maximum element will arrive after time t, but in an "encrypted" way that cannot be decoded by polynomial-sized circuits. We will make use of the well-known fact that a random function is hard on average for circuits of subexponential size.

Lemma 3. *If $g : \{0,1\}^n \to [k]$ is a random function, then with high probability there is no circuit of size $s(n) = 2^n/(8kn)$ that outputs the function value correctly on more than $\frac{2}{k}$ fraction of inputs.*

The simple proof of Lemma 3 is included in the appendix, for reference.

Theorem 8. *There exists a family of distributions $\underline{\pi} \in \mathbf{\Delta}(S_n)$ such that $V^{\underline{\pi}}(*, \iota) = 1$, but for any algorithm* ALG *whose stopping rule can be computed by circuits of size $s(n) = 2^{n/8}$, we have $V^{\underline{\pi}}(\text{ALG}, \iota) = O(1/n)$.*

Proof. Assume for convenience that n is divisible by 4. Fix a function $g : \{0,1\}^{n/4} \to [n/2]$ such that no circuit of size $s(n) = 2^{\frac{n}{4}}/(n^2)$ outputs the value of g correctly on more than $\frac{4}{n}$ fraction of inputs. The existence of such functions is ensured by Lemma 3. We use g to define a permutation distribution $\underline{\pi}$ as follows. For any binary string $x \in \{0,1\}^{n/4}$, define a permutation $\pi(x)$ by performing the following sequence of operations. First, rearrange the items in order of increasing value by mapping item i to position $n - i + 1$ for each i. Next, for $i = 1, \ldots, \frac{n}{4}$, swap the items in positions i and $i + \frac{n}{4}$ if and only if $x_i = 1$. Finally, swap the items in positions n and $\frac{n}{2} + g(x)$. (Note that this places the maximum-value item in position $\frac{n}{2} + g(x)$.) The permutation distribution $\underline{\pi}$ is the uniform distribution over $\{\pi(x) \mid x \in \{0,1\}^{n/4}\}$.

It is easy to design an algorithm which always selects the item of maximum value when the input sequence π is sampled from $\underline{\pi}$. The algorithm first decodes the unique binary string x such that $\pi = \pi(x)$, by comparing the items arriving at times i and $i + \frac{n}{4}$ for each i and setting the bit x_i according to the outcome of this comparison. Having decoded x, we then compute $g(x)$ and select the item that arrives at time $\frac{n}{2} + g(x)$. By construction, when π is drawn from $\underline{\pi}$ this is always the element of maximum value.

Finally, if ALG is any secretary algorithm we can attempt use ALG to guess the value of $g(x)$ for any input $x \in \{0,1\}^{n/4}$ by the following simulation procedure. First, define a permutation $\pi'(x)$ by performing the same sequence of operations as in $\pi(x)$ except for the final step of swapping the items in positions n and $n/2 + g(x)$; note that this means that $\pi'(x)$, unlike $\pi(x)$, can be constructed from input x by a circuit of polynomial size. Now simulate ALG on the input sequence $\pi'(x)$, observe the time t when it selects an item, and output $t - \frac{n}{2}$. The circuit complexity of this simulation procedure is at most poly(n) times the circuit complexity of the stopping rule implemented by ALG, and the fraction of inputs x on which it guesses $g(x)$ correctly is precisely $V^{\underline{\pi}}(\text{ALG}, \iota)$. (To verify this last statement, note that ALG makes its selection at time $t = \frac{n}{2} + g(x)$ when observing input sequence $\pi(x)$ *if and only if* if also makes its selection at time t when observing input sequence $\pi'(x)$, because the two input sequences are indistinguishable to comparison-based algorithms at that time.) Hence, if $V^{\underline{\pi}}(\text{ALG}, \iota) > \frac{4}{n}$ then the stopping rule of ALG cannot be implemented by circuits of size $2^{n/8}$. $\square$

Our main theorem in this section derives essentially the same result for the standard game-against-adversary interpretation of the secretary problem, rather than the game-against-nature interpretation adopted in Theorem 8.

Theorem 9. *For any function $\kappa(n)$ such that $\lim_{n\to\infty} \kappa(n) = 0$ while $\lim_{n\to\infty} \frac{n\cdot\kappa(n)}{\log n} = \infty$, there exists a family of distributions $\underline{\pi} \in \mathbf{\Delta}(S_n)$ such that $V^{\underline{\pi}} = \Omega(1)$, but any algorithm* ALG *whose stopping rule can be computed by circuits of size $s(n) = 2^{n\,\kappa(n)/4}$ satisfies $V^{\underline{\pi}}(\text{ALG}, *) = O(\kappa(n))$.*

The full proof is provided in Appendix B.3. Here we sketch the main ideas.

Proof sketch. As in Theorem 8, the algorithm and "nature" (i.e., the process sampling the input order) will work in concert with each other to bring about a correct selection, using a form of coordination that is information-theoretically easy but computationally hard. The difficulty lies in the fact that the adversary is simultaneously working to thwart their efforts. If nature, for example, wishes to use the first half of the input sequence to "encrypt" the position where item 1 will be located in the second half of the sequence, then the adversary is free to assign the maximum value to item 2 and a random value to item 1, rendering the encrypted information useless to the algorithm.

Thus, our construction of the permutation distribution $\underline{\pi}$ and algorithm ALG will be guided by two goals. First, we must "tie the adversary's hands" by ensuring that ALG has constant probability of correct selection unless the adversary's permutation, σ, is in some sense "close" to the identity permutation. Second, we must ensure that ALG has constant probability of correct selection whenever σ is close to the identity, not only when it is equal to the identity as in Theorem 8. To accomplish the second goal we modify the construction in Theorem 8 so that the first half of the input sequence encodes the binary string x using an error-correcting code. To accomplish the first goal we define $\underline{\pi}$ to be a convex combination of two distributions: the "encrypting" distribution described earlier, and an "adversary-coercing" distribution designed to make it easy for the algorithm to select the maximum-value element unless the adversary's permutation σ is close to the identity in an appropriate sense. $\square$

5 Extensions Beyond Classic Secretary Problem

We look at two generalizations of classic secretary problem in this section, namely *the multiple-choice secretary problem*, studied in [25], and *online weighted bipartite matching problem*, studied extensively in [26, 23], under our non-uniform permutation distributions. We give a positive result showing that a natural variant of the algorithm in [25] achieves a $(1 - o(1))$-competitive ratio under our pseudo-random properties defined in §2, while for the latter we show the algorithm proposed by [26] fails to achieve any constant competitive ratio under our pseudo-random properties.

Multiple-choice secretary problem We consider multiple-choice secretary problem (a.k.a. k-uniform matroid secretary problem). In this setting not only a single secretary has to be selected but up to k. An algorithm observes items with non-negative values based on the ordering $\pi\colon \mathcal{U} \to [n]$

and chooses at most k items in an online fashion. The goal is to maximize the sum of values of selected items. We consider distributions over permutations π that fulfill the (p,q,δ)-BIP, for some $p \geq k$. We show that a slight adaptation of the algorithm in [25] achieves competitive ratio $1-o(1)$, for large enough values of k and q and small enough δ.

The algorithm is defined recursively. We denote by $\mathsf{ALG}(n', k', q')$ the call of the algorithm that operates on the prefix of length n' of the input. It is allowed to choose k' items and expects q' number of blocks. For $k' = 1$, $\mathsf{ALG}(n', k', q')$ is simply the standard secretary algorithm that we analyzed in Section 2.1. For $k' > 1$, the algorithm first draws a random number $\tau(q')$ from a binomial distribution $\mathrm{Binom}(q', \frac{1}{2})$ and then executes $\mathsf{ALG}(\frac{\tau(q')}{q'}n', \lfloor k'/2 \rfloor, \tau(q'))$. After round $\frac{\tau(q')}{q'}n'$ (we assume n' is always a multiplier of q'), the algorithm accepts every item whose value is greater than the $\lfloor k'/2 \rfloor$-highest item arrived during rounds $1, \ldots, \frac{\tau(q')}{q'}n'$, until k' items are selected by the algorithm or until round n'. Output is the union of all items returned by the recursive call and all items algorithm picked after the threshold round. We now prove the following theorem.

Theorem 10. *Suppose π is drawn from a permutation distribution that satisfies (p,q,δ)-BIP for some $p \geq k$ and $\delta \leq \frac{1}{k^{\frac{1}{2}}}$. Then for all permutations σ, $\mathsf{ALG}(\mathcal{U}, k, q)$ is $(1 - O(\frac{1}{k^{\frac{1}{3}}}) - \epsilon)$-competitive for the k-uniform matroid secretary problem, where ϵ can be arbitrary small for large enough value of q and small enough value of δ.*

Online weighted bipartite matching Next, we consider online weighted biparitte matching, where the vertices on the offline side of a bipartite graph are given in advance and the vertices on the online side arrive online in a random order (not necessarily uniform). Whenever a vertex arrives, its adjacent edges with the corresponding weights are revealed and the online algorithm has to decide which of these edges should be included in the matching. The objective is to maximize the weight of the matching selected by online algorithm. A celebrated result of Korula and Pál [26] shows the existence of a constant competitive online algorithm under uniform random order of arrival; nevertheless, this algorithm does not achieve any constant competitive ratio under our non-uniform assumptions for permutation distributions.

Theorem 11. *For every k and δ, there is an instance and a probability distribution that fulfills the (k,δ)-uniform-induced-ordering property such that the competitive ratio of the Korula-Pál algorithm is at least $\Omega\left(\frac{\delta^2}{(k+1)!}\frac{n}{\ln n}\right)$.*

6 Conclusion

In this paper we have studied how secretary algorithms perform when the arrival order satisfies relaxations of the uniform-random-order hypothesis. We presented a pair of closely-related properties (the (k,δ)-UIOP and the (p,q,δ)-BIP) that ensure that the standard secretary algorithm has constant probability of correct selection, and we derived some results on the minimum amount of entropy and the minimum circuit complexity necessary to achieve constant probability of correct selection in secretary problems with non-uniform arrival order.

We believe this work represents a first step toward obtaining a deeper understanding of the amount and type of randomness required to obtain strong performance guarantees for online algorithms. The next step is to expand this study beyond the setting of secretary problems. A very promising domain for future investigation is online packing LP and its generalization, online convex programming. Our positive result on the uniform matroid secrerary problem constitutes a first step toward obtaining a general positive result confirming that existing algorithms such as the algorithms of [24] and [1] preserve their performance guarantees when the input ordering satisfies (k, δ)-UIOP or some other relaxation of the uniform randomness assumption.

References

[1] Agrawal, S. and Devanur, N. (2015). Fast algorithms for online stochastic convex programming. In *Proc. 25th Annual ACM-SIAM Symposium on Discrete Algorithms*.

[2] Agrawal, S., Wang, Z., and Ye, Y. (2014). A dynamic near-optimal algorithm for online linear programming. *Operations Research*, 62:867–890.

[3] Arora, S. and Barak, B. (2009). *Computational complexity: A modern approach*. Cambridge University Press.

[4] Babaioff, M., Immorlica, N., Kempe, D., and Kleinberg, R. (2007a). A knapsack secretary problem with applications. In *Proc. 2007 Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX)*, pages 16–28. Springer.

[5] Babaioff, M., Immorlica, N., and Kleinberg, R. (2007b). Matroids, secretary problems, and online mechanisms. In *Proc. 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 434–443.

[6] Baraniuk, R., Davenport, M., DeVore, R., and Wakin, M. (2008). A simple proof of the restricted isometry property for random matrices. *Constructive Approximation*, 28(3):253–263.

[7] Bateni, M., Hajiaghayi, M., and Zadimoghaddam, M. (2013). Submodular secretary problem and extensions. *ACM Transactions on Algorithms (TALG)*, 9(4):32.

[8] Borchardt, C. W. (1860). Uber eine Interpolationsformel für eine art symmetrischer functionen und über deren anwendung. *Math. Abh. der Akademie der Wissenschaften zu Berlin*, pages 1–20.

[9] Candes, E. J. and Tao, T. (2005). Decoding by linear programming. *IEEE Trans. Information Theory*, 51(12):4203–4215.

[10] Carothers, N. L. (2009). A short course on approximation theory. http://personal.bgsu.edu/∼carother/Approx.html. Manuscript.

[11] Cayley, A. (1889). A theorem on trees. *Quarterly J. Math*, 23:376–378.

[12] Devanur, N. and Hayes, T. P. (2009). The AdWords problem: Online keyword matching with budgeted bidders under random permutations. In *Proc. 10th ACM Conference on Electronic Commerce*, pages 71–78.

[13] Devanur, N. R., Jain, K., Sivan, B., and Wilkens, C. A. (2011). Near optimal online algorithms and fast approximation algorithms for resource allocation problems. In *Proc. 12th ACM Conference on Electronic Commerce*, pages 29–38. ACM.

[14] Dimitrov, N. B. and Plaxton, C. G. (2012). Competitive weighted matching in transversal matroids. *Algorithmica*, 62(1-2):333–348.

[15] Dynkin, E. B. (1963). The optimum choice of the instant for stopping a Markov process. *Sov. Math. Dokl.*, 4.

[16] Feldman, J., Henzinger, M., Korula, N., Mirrokni, V. S., and Stein, C. (2010). Online stochastic packing applied to display ad allocation. In *Algorithms–ESA 2010*, pages 182–194. Springer.

[17] Feldman, M., Naor, J. S., and Schwartz, R. (2011). Improved competitive ratios for submodular secretary problems. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 218–229. Springer.

[18] Feldman, M., Svensson, O., and Zenklusen, R. (2015). A simple O(log log(rank))-competitive algorithm for the matroid secretary problem. In *Proc. 25th Annual ACM-SIAM Symposium on Discrete Algorithms*.

[19] Göbel, O., Hoefer, M., Kesselheim, T., Schleiden, T., and Vöcking, B. (2014). Online independent set beyond the worst-case: Secretaries, prophets, and periods. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II*, pages 508–519.

[20] Hajiaghayi, M. T., Kleinberg, R., and Parkes, D. C. (2004). Adaptive limited-supply online auctions. In *Proc. 5th ACM conference on Electronic commerce*, pages 71–80. ACM Press.

[21] Jaillet, P., Soto, J. A., and Zenklusen, R. (2013). Advances on matroid secretary problems: Free order model and laminar case. In *Integer Programming and Combinatorial Optimization*, pages 254–265. Springer.

[22] Kaplan, E., Naor, M., and Reingold, O. (2009). Derandomized constructions of k-wise (almost) independent permutations. *Algorithmica*, 55(1):113–133.

[23] Kesselheim, T., Radke, K., Tönnis, A., and Vöcking, B. (2013). An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions. In *Algorithms–ESA 2013*, pages 589–600. Springer.

[24] Kesselheim, T., Radke, K., Tönnis, A., and Vöcking, B. (2014). Primal beats dual on online packing lps in the random-order model. In *Proc. ACM Symposium on Theory of Computing*, pages 303–312.

[25] Kleinberg, R. D. (2005). A multiple-choice secretary algorithm with applications to online auctions. In *Proc. 16th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 630–631.

[26] Korula, N. and Pál, M. (2009). Algorithms for secretary problems on graphs and hypergraphs. In *ICALP (2)*, pages 508–520.

[27] Lachish, O. (2014). O(log log rank) competitive-ratio for the matroid secretary problem (the known cardinality variant). In *Proc. 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*.

[28] Lindley, D. V. (1961). Dynamic programming and decision theory. *Applied Statistics*, 10:39–51.

[29] Meyerson, A. (2001). Online facility location. In *Proc. 42nd Annual Symposium on Foundations of Computer Science*, pages 426–431.

[30] Meyerson, A., Munagala, K., and Plotkin, S. A. (2001). Designing networks incrementally. In *Proc. 42nd Annual Symposium on Foundations of Computer Science*, pages 406–415.

[31] Mitzenmacher, M. and Vadhan, S. (2008). Why simple hash functions work: exploiting the entropy in a data stream. In *Proc. 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 746–755. Society for Industrial and Applied Mathematics.

[32] Molinaro, M. and Ravi, R. (2015). The geometry of online packing linear programs. *Math. of Operations Research.* to appear.

[33] Motwani, R. and Raghavan, P. (1995). *Randomized Algorithms.* Cambridge University Press.

[34] Roughgarden, T. and Trevisan, L. (2011). Workshop on beyond worst-case analysis. Stanford University, September 2011. http://theory.stanford.edu/∼tim/bwca/bwca.html.

[35] Samuels, S. M. (1981). Minimax stopping rules when the underlying distribution is uniform. *J. Amer. Statist. Assoc.*, 76:188–197.

A A secretary algorithm for $(3, \delta)$-induced-ordering property

Theorem 12. *If a probability distribution fulfills the $(3, \delta)$-induced-ordering property, there is an algorithm for the secretary problem that selects the best item with probability $\frac{(1-\delta)^2}{6(1+\delta)}$.*

Proof. Consider the following algorithm: First we draw a threshold τ uniformly at random. Then we observe all items until round τ. After round τ, we accept the first item that is better than all items seen so far.

To analyze this algorithms let $x_1, x_2, \ldots, x_n$ be the items in order of decreasing value. To select x_1 it suffices that x_2 comes until round τ and x_1 comes after round τ. For $i \geq 3$, let Y_i be a 0/1 random variable indicating if $\pi(x_2) < \pi(x_i) < \pi(x_1)$.

Conditioned on $\sum_{i=3}^n Y_i = a$ and $\pi(x_2) < \pi(x_1)$, the probability that x_2 comes until round τ and x_1 comes after round τ is exactly $\frac{a+1}{n}$ because there are a items coming between x_2 and x_1, giving $a+1$ positive outcomes for τ.

We have $\mathbf{E}\left[Y_i\right] \geq (1-\delta)\frac{1}{3!} = \frac{1-\delta}{6}$. As $Y_i = 1$ implies $\pi(x_2) < \pi(x_1)$, we get $\mathbf{E}\left[Y_i \mid \pi(x_2) < \pi(x_1)\right] \geq \frac{1-\delta}{6\mathbf{Pr}[\pi(x_2)<\pi(x_1)]} = \frac{1-\delta}{6(1-\mathbf{Pr}[\pi(x_2)>\pi(x_1)])} \geq \frac{1-\delta}{6(1-\frac{1-\delta}{2})} = \frac{1-\delta}{3(1+\delta)}$.

Overall, we get

$$\begin{aligned}
\mathbf{Pr}\left[\text{select } x_1 \mid \pi(x_2) < \pi(x_1)\right] &\geq \sum_{a=0}^{n-3} \mathbf{Pr}\left[\sum_{i=3}^n Y_i = a \,\middle|\, \pi(x_2) < \pi(x_1)\right] \frac{a+1}{n} \\
&= \frac{1}{n}\left(1 + \sum_{a=0}^{n-3} a \mathbf{Pr}\left[\sum_{i=3}^n Y_i = a \,\middle|\, \pi(x_2) < \pi(x_1)\right]\right) \\
&= \frac{1}{n}\left(1 + \mathbf{E}\left[\sum_{i=3}^n Y_i \,\middle|\, \pi(x_2) < \pi(x_1)\right]\right) \\
&\geq \frac{1}{n}\left(1 + \mathbf{E}\left[\sum_{i=3}^n Y_i \,\middle|\, \pi(x_2) < \pi(x_1)\right]\right) \\
&\geq \frac{1}{n} + \frac{n-3}{n}\frac{1-\delta}{3(1+\delta)} \\
&\geq \frac{1-\delta}{3(1+\delta)} \quad .
\end{aligned}$$

Multiplying with $\mathbf{Pr}\left[\pi(x_2) < \pi(x_1)\right] \geq \frac{1-\delta}{2}$, we get

$$\mathbf{Pr}\left[\text{select } x_1\right] \geq \frac{(1-\delta)^2}{6(1+\delta)} \quad .$$

□

B Deferred proofs

B.1 Proofs deferred from §2

In this section we restate some of the results from §2 and provide complete proofs.

B.1.1 Full Proof of Theorem 1

The (p, q, δ)-block-independence property only makes statements about p-tuples. We will need the bound also for smaller tuples. Indeed, using a simple counting argument we can show that this is already implicit in the definition.

Lemma 4. *If a distribution over permutations is (p, q, δ)-block-independent, then it is also (p', q, δ)-block-independent for any $p' < p$.*

Proof. Given $x_1, \ldots, x_{p'} \in \mathcal{U}$ and $b_1 \ldots, b_{p'} \in [q]$, fill up the first tuple with arbitrary distinct entries $x_{p'+1}, \ldots, x_p \in \mathcal{U}$. The event $\bigwedge_{j\in[p']} \pi^B(x_i) = b_i$ can now be expressed as the union of all events $\bigwedge_{j\in[p]} \pi^B(x_i) = b_i$ over all tuples $(b_{p'+1} \ldots, b_p) \in [q]^{p-p'}$. Note that these events are pairwise disjoint. Therefore, the probability of their union is the sum of their probabilities, i.e.,

$$\mathbf{Pr}\left[\bigwedge_{j\in[p']} \pi^B(a_i) = b_i\right] = \mathbf{Pr}\left[\bigvee_{(b_{p'+1},\ldots,b_p)\in[q]^{p-p'}} \bigwedge_{j\in[p]} \pi^B(x_i) = b_i\right]$$

$$= \sum_{(b_{p'+1},\ldots,b_p)\in[q]^{p-p'}} \mathbf{Pr}\left[\bigwedge_{j\in[p]} \pi^B(x_i) = b_i\right] \quad .$$

Using (p, q, δ)-block-independence and $|[q]^{p-p'}| = q^{p-p'}$, we get

$$\mathbf{Pr}\left[\bigwedge_{j\in[p']} \pi^B(x_i) = b_i\right] \geq \sum_{(b_{p'+1},\ldots,b_p)\in[q]^{p-p'}} (1-\delta)\left(\frac{1}{q}\right)^p$$

$$= q^{p-p'}(1-\delta)\left(\frac{1}{q}\right)^p$$

$$= (1-\delta)\left(\frac{1}{q}\right)^{p'} \quad .$$

□

Proof of Theorem 1. Let $T = \lfloor \frac{q}{\mathrm{e}} \rfloor$ denote the index of the block in which the threshold is located. Furthermore, let $x_j \in \mathcal{U}$ be the jth best item. We condition on the event that x_1 comes in block with index i. To ensure that our algorithm picks this item, it suffices that x_2 comes in blocks $1, \ldots, T-1$. Alternatively, we also pick x_1 if the x_2 comes in blocks $i+1, \ldots, q$ and x_3 comes in blocks $1, \ldots, T-1$. Continuing this argument, we get

$$\mathbf{Pr}\left[\text{correct selection}\right] \geq \sum_{i=T+1}^{q} \sum_{j=2}^{p} \mathbf{Pr}\left[\pi^B(x_1) = i, \pi^B(x_2), \ldots, \pi^B(x_{j-1}) > i, \pi^B(x_j) < T\right] \quad .$$

We can now use Lemma 4 and apply (j, q, δ)-block-independence for $j \leq p$. This gives us

$$\mathbf{Pr}\left[\text{correct selection}\right] \geq \sum_{i=T+1}^{q} \sum_{j=2}^{p} (1-\delta) \frac{1}{q} \left(\frac{q-i}{q}\right)^{j-2} \frac{T-1}{q} \enspace .$$

We now reorder the sums and use the formula for finite geometric series. This gives us

$$\begin{aligned}
\mathbf{Pr}\left[\text{correct selection}\right] &\geq (1-\delta)\frac{T-1}{q} \sum_{i=T+1}^{q} \frac{1}{q} \left(\sum_{j=2}^{p} \left(\frac{q-i}{q}\right)^{j-2} \right) \\
&= (1-\delta)\frac{T-1}{q} \sum_{i=T+1}^{q} \frac{1}{q} \frac{1 - \left(\frac{q-i}{q}\right)^{p-1}}{\frac{i}{q}} \\
&= (1-\delta)\frac{T-1}{q} \sum_{i=T+1}^{q} \frac{1}{i} \left(1 - \left(\frac{q-i}{q}\right)^{p-1} \right) \\
&\geq (1-\delta)\frac{T-1}{q} \left(1 - \left(\frac{q-T}{q}\right)^{p-1} \right) \sum_{i=T+1}^{q} \frac{1}{i} \enspace .
\end{aligned}$$

We now apply the following bounds

$$\frac{T-1}{q} \geq \frac{1}{\mathrm{e}} - \frac{2}{q} \enspace , \qquad \frac{q-T}{q} \leq 1 - \frac{1}{\mathrm{e}} \enspace , \qquad \text{and}$$

$$\sum_{i=T+1}^{q} \frac{1}{i} \geq \int_{T+1}^{q+1} \frac{1}{x} dx = \ln\left(\frac{q+1}{T+1}\right) \geq \ln\left(\frac{q+1}{\frac{q}{\mathrm{e}}+1}\right) = 1 - \ln\left(\frac{q+\mathrm{e}}{q+1}\right) \geq 1 - \left(\frac{q+\mathrm{e}}{q+1} - 1\right) = 1 - \frac{\mathrm{e}-1}{q+1} \enspace .$$

In combination, they imply

$$\begin{aligned}
\mathbf{Pr}\left[\text{correct selection}\right] &\geq \left(\frac{1}{\mathrm{e}} - \frac{2}{q}\right)(1-\delta)\left(1 - \left(1 - \frac{1}{\mathrm{e}}\right)^{p-1}\right)\left(1 - \frac{\mathrm{e}-1}{q+1}\right) \\
&\geq \left(\frac{1}{\mathrm{e}} - \frac{\mathrm{e}+1}{q}\right)(1-\delta)\left(1 - \left(1 - \frac{1}{\mathrm{e}}\right)^{p-1}\right) \\
&\geq \frac{1}{\mathrm{e}} - \frac{\mathrm{e}+1}{q} - \delta - \left(1 - \frac{1}{\mathrm{e}}\right)^{p-1} \enspace .
\end{aligned}$$

□

B.1.2 Relation between the two properties

Theorem 13. *If a distribution over permutation fulfills the (p, q, δ)-BIP, then it also fulfills the $(p, \delta + \frac{p^2}{q})$-UIOP.*

Proof. Note that it is safe to assume $p \leq q$ as the statement is trivially fulfilled otherwise. Consider p distinct items $x_1, \ldots, x_p \in \mathcal{U}$. To have $\pi(x_1) < \pi(x_2) < \ldots < \pi(x_p)$, it suffices that these elements are mapped to different blocks and with an increasing sequence of indices. There are $\binom{q}{p}$ such sequences. So, overall the probability is at least

$$\binom{q}{p}(1-\delta)\left(\frac{1}{q}\right)^p \geq \frac{(q-p)^p}{p!}(1-\delta)\left(\frac{1}{q}\right)^p \geq \left(1 - \frac{p}{q}\right)^p (1-\delta)\frac{1}{p!} \geq \left(1 - \delta\frac{p^2}{q}\right)\frac{1}{q!} \enspace .$$

□

To show this direction, we borrow techniques from *theory of approximation of functions* and prove the aforementioned two properties are essentially equivalent in limit. We start by some definitions and notations from approximation theory; see, e.g., the textbook by Carothers [10].

Definition 3 ([10])**.** If f is any bounded function over $[0,1]$, we define the sequence of *Bernstein polynomials* for f by

$$(B_d(f))(x) = \sum_{k=0}^{d} f(k/d)\binom{d}{k} x^k (1-x)^{d-k}, \quad 0 \le x \le 1. \tag{1}$$

Remark 1. $B_d(f)$ is a polynomial of degree at most d.

Definition 4 ([10])**.** The *modulus of continuity* of a bounded function f over $[a,b]$ is defined by

$$\omega_f(\delta) = \sup\{|f(x_1) - f(x_2)| : x_1, x_2 \in [a,b], |x_1 - x_2| \le \delta\} \tag{2}$$

Remark 2. bounded function f is *continuous* over interval $[a,b]$ if and only if $\omega_f(\delta) = O(\delta)$. Moreover, f is *uniformly continuous* if and only if $\omega_f(\delta) = o(\delta)$.

We are now ready to state our main ingredient, i.e. *Bernstein's approximation theorem*, which shows bounded functions with enough continuity are well approximated by Bernstein polynomials.

Theorem 14 ([10])**.** *For any bounded function f over $[0,1]$ we have*

$$\|f - B_d(f)\|_\infty \le \frac{3}{2}\omega_f(\frac{1}{\sqrt{d}}) \tag{3}$$

where for any bounded functions f_1 and f_2, $\|f_1 - f_2\|_\infty \triangleq \sup\{|f_1(x) - f_2(x)| : x \in [0,1]\}$.

To prove our claim, we start by showing (k,δ)-uniform-induced-ordering property forces the arrival time of items to have almost the same higher-order moments as uniform independent random variables. More precisely, we have the following lemma.

Lemma 5. *Suppose π is drawn from a permutation distribution satisfying (k,δ)-uniform-induced-ordering property, and $\{x_1, \ldots, x_p\}$ is an arbitrary set of p items. Let $\phi : [n] \to \{i/n : i \in [n]\}$ be a uniform random mapping, and random variables $X_i \triangleq \pi(x_i)/n$ for all $i \in [p]$. Then for every $k_i \le \frac{k}{2p}$ we have $\mathbf{E}\left[\prod_{i=1}^{p} X_i^{k_i}\right] \ge (1-\delta)\mathbf{E}\left[\prod_{i=1}^{p} \phi(i)^{k_i}\right]$.*

Proof. We first define random variables $I_{i,j} \triangleq \mathbf{I}_{\pi(i)\le\pi(j)}$ and $\tilde{I}_{i,j} \triangleq \mathbf{I}_{\phi(i)\le\phi(j)}$ for all $i,j \in [n]$. Note that for all $i \in [p]$, $X_i = \pi(x_i)/n = \frac{\sum_{j=1}^{n} I_{i,j}}{n}$. This implies that

$$\mathbf{E}\left[\prod_{i=1}^{p} X_i^{k_i}\right] = \mathbf{E}\left[\prod_{i=1}^{p}\left(\frac{\sum_{j=1}^{n} I_{i,j}}{n}\right)^{k_i}\right] = \frac{\mathbf{E}\left[\prod_{i=1}^{p}\left(\sum_{j=1}^{n} I_{i,j}\right)^{k_i}\right]}{n^{(\sum_{i=1}^{p} k_i)}} \tag{4}$$

By expanding the numerator due to linearity of expectation, we will have a sum of expectation of algebraic terms in the numerator, where each algebraic term multiplication of at most $\frac{k}{2p} \times p = k/2$ indicators $I_{i,j}$. There are at most $2 \times k/2 = k$ particular items involved in these indicator functions. Now, lets look at one of the terms, e.g. $\mathbf{E}\left[\prod_{l=1}^{k/2} I_{i_l,j_l}\right]$ in which k items $\{x_{s_1}, \ldots, x_{s_k}\}$ are involved. The product $\prod_{l=1}^{k} I_{i_l,j_l}$ forces the induced ordering of elements $\{x_{s_1}, \ldots, x_{s_k}\}$ be in a particular subset $S \subseteq S_k$.

Hence, $\mathbf{E}\left[\prod_{l=1}^{k/2} I_{i_l,j_l}\right] = \mathbf{Pr}\left[\text{induced ordering by } \pi \text{ over } \{x_{s_1},\ldots,x_{s_k}\} \text{ will be in } S\right]$. Now as π satisfies the (k,δ)-uniform-induced-ordering property, we have

$$\begin{aligned}\mathbf{E}\left[\prod_{l=1}^{k/2} I_{i_l,j_l}\right] &= \mathbf{Pr}\left[\text{induced ordering by } \pi \text{ over } \{x_{s_1},\ldots,x_{s_k}\} \text{ will be in } S\right] \\ &\geq (1-\delta)\mathbf{Pr}\left[\text{induced ordering by } \phi \text{ over } \{x_{s_1},\ldots,x_{s_k}\} \text{ will be in } S\right] \\ &= (1-\delta)\mathbf{E}\left[\prod_{l=1}^{k/2} \tilde{I}_{i_l,j_l}\right] \end{aligned} \tag{5}$$

From (5) one can conclude that

$$\mathbf{E}\left[\prod_{i=1}^{p} X_i^{k_i}\right] \geq (1-\delta)\mathbf{E}\left[\prod_{i=1}^{p}\left(\frac{\sum_{j=1}^{n}\tilde{I}_{i,j}}{n}\right)^{k_i}\right] = (1-\delta)\mathbf{E}\left[\prod_{i=1}^{p}\left(\frac{n\phi(i)}{n}\right)^{k_i}\right] = (1-\delta)\mathbf{E}\left[\prod_{i=1}^{p}\phi(i)^{k_i}\right] \tag{6}$$

which completes the proof. □

Given Lemma 5, roughly speaking the key idea for the rest of the proof is looking at probabilities as the expectation of the indicator functions, and then trying to approximate the indicator functions by polynomials. Now, to compute probabilities all we need are moments, which due to Lemma 5 are almost equal to those of uniform independent random variables. Rigorously, we prove the following probabilistic lemma using this idea.

Lemma 6. *Let $\phi : [n] \to \{i/n : i \in [n]\}$ be a uniform random mapping. Furthermore, let $X_1, X_2, \ldots, X_p$ be random variables over $[0,1]$ such that for every $k_i \leq d$ we have $\mathbf{E}\left[\prod_{i=1}^{p} X_i^{k_i}\right] \geq \mathbf{E}\left[\prod_{i=1}^{p}\phi(i)^{k_i}\right](1-\delta)$. Then for any disjoint intervals $\{(a_i,b_i)\}_{i=1}^{p}$ of $[0,1]$ (where a_i and b_i are multiples of $1/n$) we have:*

$$\mathbf{Pr}\left[\bigwedge_{i=1}^{p}(X_i \in [a_i,b_i])\right] \geq \left(\prod_{i=1}^{p}(b_i - a_i)\right)(1-\delta) - \frac{7p}{d^{\frac{1}{4}}} \tag{7}$$

Proof. We define continuous functions $f_i \colon [0,1] \to \mathbb{R}$ for $i \in [p]$ by

$$f_i(x) = \begin{cases} 0 & \text{for } x < a_i \text{ or } x > b_i \\ \frac{x-a_i}{\gamma} & \text{for } a_i \leq x \leq a_i + \gamma \\ -\frac{x-b_i}{\gamma} & \text{for } b_i - \gamma \leq x \leq b_i \\ 1 & \text{for } a_i + \gamma \leq x \leq b_i - \gamma \end{cases}$$

Note that all of these functions are continuous and satisfy condition of Theorem 14 for $\omega_{f_i}(x) = \frac{x}{\gamma}$.

Observe that f_i is point-wise smaller than the indicator function $\mathbf{1}_{[a_i,b_i]}$ that has value 1 between a_i and b_i and 0 otherwise. Therefore, we have $\mathbf{Pr}\left[\bigwedge_{i=1}^{p}(X_i \in [a_i,b_i])\right] = \mathbf{E}\left[\prod_{i=1}^{p}\mathbf{1}_{[a_i,b_i]}(X_i)\right] \geq \mathbf{E}\left[\prod_{i=1}^{p} f_i(X_i)\right]$.

By theorem 14, for every i there is a polynomial function $g_i \colon [0,1] \to \mathbb{R}$ of degree d such that $\|f_i - g_i\|_\infty \leq \frac{3}{2}\omega_{f_i}(\frac{1}{\sqrt{d}}) \leq \frac{3}{2\gamma\sqrt{d}}$. We now have $g_i(\phi(i)) \geq f_i(\phi(i)) - \frac{3}{2\gamma\sqrt{d}} \geq \mathbf{1}_{[a_i+\gamma,b_i-\gamma]}(\phi(i)) - \frac{3}{2\gamma\sqrt{d}}$

and therefore

$$\mathbf{E}\left[\prod_{i=1}^{p} g_i(\phi(i))\right] \geq \mathbf{E}\left[\prod_{i=1}^{p}\left(\mathbf{1}_{[a_i+\gamma,b_i-\gamma]}(\phi(i)) - \frac{3}{2\gamma\sqrt{d}}\right)\right] \geq \prod_{i=1}^{p}\mathbf{Pr}\left[\phi(i)\in[a_i+\gamma,b_i-\gamma]\right] - \frac{3p}{2\gamma\sqrt{d}}$$
$$= \prod_{i=1}^{p}(b_i-a_i-2\gamma) - \frac{3p}{2\gamma\sqrt{d}} \geq \prod_{i=1}^{p}(b_i-a_i) - (2p\gamma + \frac{3p}{2\gamma\sqrt{d}}) \overset{(1)}{\geq} \prod_{i=1}^{p}(b_i-a_i) - \frac{4p}{d^{0.25}} \tag{8}$$

where to get inequality (1) we set $\gamma = \frac{1}{2}d^{-0.25}$. Furthermore, as g_i is a polynomial function of degree at most d and $\mathbf{E}\left[\prod_{i=1}^{p} X_i^{k_i}\right] \geq \mathbf{E}\left[\prod_{i=1}^{p}\phi(i)^{k_i}\right](1-\delta)$ for all $k_i \leq d$, we get by linearity of expectation $\mathbf{E}\left[\prod_{i=1}^{p} g(X_i)\right] \geq (1-\delta)\mathbf{E}\left[\prod_{i=1}^{p} g_i(\phi(i))\right]$. Now we use $g_i(x) \leq f_i(x) + \frac{3}{2\gamma\sqrt{d}} = \frac{3}{d^{0.25}}$, giving us

$$\mathbf{E}\left[\prod_{i=1}^{p} f_i(X_i)\right] \geq \mathbf{E}\left[\prod_{i=1}^{p}(g_i(X_i) - \frac{3}{d^{0.25}})\right] \geq \mathbf{E}\left[\prod_{i=1}^{p} g_i(X_i)\right] - \frac{3p}{d^{0.25}} \geq (1-\delta)\mathbf{E}\left[\prod_{i=1}^{p} g_i(\phi(i))\right] - \frac{3p}{d^{0.25}}$$
$$\geq (1-\delta)\left(\prod_{i=1}^{p}[b_i-a_i] - \frac{4p}{d^{0.25}}\right) - \frac{3p}{d^{0.25}} \geq (1-\delta)\left(\prod_{i=1}^{p}[b_i-a_i]\right) - \frac{7p}{d^{0.25}} \tag{9}$$

Overall, we get $\mathbf{Pr}\left[\bigwedge_{i=1}^{p}(X_i \in [a_i,b_i])\right] \geq (1-\delta)\left(\prod_{i=1}^{p}(b_i-a_i)\right) - \frac{7p}{d^{0.25}}$, as desired. □

Now, by combining Lemma 5 and Lemma 6 we essentially prove the following theorem.

Theorem 15. *If a distribution over permutation fulfills the (k,δ)-uniform-induced-ordering property, then it also satisfies (p,q,δ)-block-independence property for $p = o(k^{\frac{1}{5}})$, $q = O(k^{\frac{1}{5}})$ as k goes to infinity.*

Proof. We try checking the (p,q,δ)-block-independence property. Start by setting $d = \frac{k}{2p}$. By Lemma 6 the probability approximation error from what desired will be $O\left(\frac{p}{d^{0.25}}\right)=O\left(\frac{p^{\frac{5}{4}}}{k^{\frac{1}{4}}}\right)$. This error goes to zero as $k \to \infty$ if we set $p = o(k^{\frac{1}{5}})$. Moreover, we need $|b_i - a_i| \geq 2\gamma$. So, $\frac{1}{q} = \Omega\left(\frac{1}{d^{\frac{1}{4}}}\right)$. As $d = \omega(k^{\frac{4}{5}})$, if we set $q = O(k^{\frac{1}{5}})$ we are fine. □

B.1.3 Full proof of Theorem 4

In this section we present one natural construction leading to a distribution that satisfies the (k,δ)-UIOP. The starting point for the construction is an n-tuple of vectors $x_1,\ldots,x_n \in \mathbb{R}^d$. If one sorts these vectors according to a random one-dimensional projection (i.e., ranks the vectors in increasing order of $w \cdot x_i$, for a random w drawn from a spherically symmetric distribution), when does the resulting random ordering satisfy the (k,δ)-UIOP? Note that if any k of these vectors comprise an orthonormal k-tuple and one ranks them in increasing order of $w \cdot x_i$, where w is drawn from a spherically symmetric distribution, then a trivial symmetry argument shows that the induced ordering of the k vectors is uniformly random. Intuitively, then, if the vectors $x_1,\ldots,x_n$ are sufficiently "incoherent", then any k-tuple of them should be nearly orthonormal and their induced ordering when projected onto the 1-dimensional subspace spanned by w should be approximately uniformly random. The present section is devoted to making this intuition quantitative. We begin by recalling the definition of the restricted isometry property [9].

Definition 5. A matrix X satisfies the restricted isometry property (RIP) of order k with restricted isometry constant δ_k if the inequalities

$$(1-\delta_k)\|x\|^2 \leq \|X_T x\|^2 \leq (1+\delta_k)\|x\|^2$$

hold for every submatrix X_T composed of $|T| \leq k$ columns of X and every vector $x \in \mathbb{R}^{|T|}$. Here $\|\cdot\|$ denotes the Euclidean norm.

Several random matrix distributions are known to give rise to matrices satisfying the RIP with high probability. The simplest such distribution is a random d-by-n matrix with i.i.d. entries drawn from the normal distribution $\mathcal{N}\big(0, \frac{1}{d}\big)$. It is known [6, 9] that, with high probability, such a matrix satisfies the RIP of order k with restricted isometry constant δ provided that $d = \Omega\big(\frac{k \log n}{\delta^2}\big)$. Even if the columns $x_1, \ldots, x_n$ of X are not random, if they are sufficiently "incoherent" unit vectors, meaning that $x_i \cdot x_j = 1$ if $i = j$ and $x_i \cdot x_j < \delta_k / k$ otherwise, then X satisfies the RIP.

Theorem 16. *Let $x_1, \ldots, x_n$ be the columns of a matrix that satisfies the RIP of order k with restricted isometry constant $\delta_k = \frac{\delta}{3k}$. If w is drawn at random from a spherically symmetric distribution and we use w to define a permutation of $[n]$ by sorting its elements in order of increasing $w \cdot x_i$, the resulting distribution over S_n satisfies the (k, δ)-UIOP.*

Proof. For any k-tuple of indices $(i_1, \ldots, i_k)$ we must show that each of the $k!$ possible orderings of $(w \cdot i_1), \ldots, (w \cdot i_k)$ has probability at least $\frac{1-\delta}{k!}$. By symmetry it suffices to show that the probability of the event $\{w \cdot x_1 < w \cdot x_2 < \cdots < w \cdot x_k\}$ is at least $\frac{1-\delta}{k!}$. This event is unchanged by rescaling w, so we are free to substitute whatever spherically-symmetric distribution we wish. Henceforth assume w is sampled from the multivariate normal distribution $\mathcal{N}(0, 1)$, whose density function is $(2\pi)^{-d/2} \exp\big(\frac{1}{2}\|w\|^2\big)$.

Let X_k denote the matrix whose k columns are the vectors $x_1, \ldots, x_k$ and let $A = X_k^{\intercal}$ denote its transpose. Scaling $x_1, \ldots, x_k$ by a common scalar, if necessary, we are free to assume that $\det(X_k^{\intercal} X_k) = 1$. The RIP implies that the ratio of the largest and smallest right singular values of X_k is at most $\frac{1+\delta_k}{1-\delta_k}$, and since their product is 1 this means that the smallest singular value is at least $\frac{1-\delta_k}{1+\delta_k}$.

Now let $\mathcal{C} = \{z \in \mathbb{R}^k \mid z_1 < z_2 < \cdots < z_k\}$. The event $\{w \cdot x_1 < w \cdot x_2 < \cdots < w \cdot x_k\}$ can be expressed more succinctly as $\{Aw \in \mathcal{C}\}$, and its probability is

$$\mathbf{Pr}\left[Aw \in \mathcal{C}\right] = \int_{w \in A^{-1}(\mathcal{C})} (2\pi)^{-d/2} \exp\big(-\tfrac{1}{2}\|w\|^2\big)\, dw.$$

The matrix A is not square, hence not invertible; the notation $A^{-1}(\mathcal{C})$ merely means the inverse-image of $\mathcal{C}$ under the linear transformation $\mathbb{R}^d \to \mathbb{R}^k$ represented by A. The Moore-Penrose pseudoinverse of A is the matrix $X_k(X_k^{\intercal} X_k)^{-1}$, which we denote henceforth by A^+. We can write any $w \in A^{-1}(\mathcal{C})$ uniquely as $z + A^+ y$, where $y \in \mathcal{C}$, $z \in \ker(A)$, and z is orthogonal to $A^+ y$. By our scaling assumption, the product of the singular values of A^+ equals 1, which justifies the second line in the following calculation

$$
\begin{aligned}
\mathbf{Pr}\left[Aw \in \mathcal{C}\right] & \int_{w \in A^{-1}(\mathcal{C})} (2\pi)^{-d/2} \exp\left(-\tfrac{1}{2}\|w\|^2\right) dw \\
&= \int_{y \in \mathcal{C}} \int_{z \in \ker(A)} (2\pi)^{-d/2} \exp\left(-\tfrac{1}{2}\|z\|^2 - \tfrac{1}{2}\|A^+ y\|^2\right) dz\, dy \\
&= \left(\int_{y \in \mathcal{C}} (2\pi)^{-k/2} \exp\left(-\tfrac{1}{2}\|A^+ y\|^2\right) dy\right) \left(\int_{z \in \ker(A)} (2\pi)^{-(d-k)/2} \exp\left(-\tfrac{1}{2}\|z\|^2\right) dz\right) \\
&= \left(\int_{y \in \mathcal{C}} (2\pi)^{-k/2} \exp\left(-\tfrac{1}{2}\|A^+ y\|^2\right) dy\right).
\end{aligned}
$$

We can rewrite the right side as an integral in spherical coordinates. Let $d\omega$ denote the volume element of the unit sphere $S^{k-1} \subset \mathbb{R}^k$ and let $\mathcal{S} = \mathcal{C} \cap S^{k-1}$. Then writing $y = ru$, where $r \geq 0$ and u is a unit vector, we have

$$
\begin{aligned}
\mathbf{Pr}\left[Aw \in \mathcal{C}\right] &= \int_{u \in \mathcal{S}} \int_{r=0}^{\infty} (2\pi)^{-k/2} \exp\left(-\tfrac{1}{2} r^2 \|A^+ u\|^2\right) r^{k-1}\, dr\, d\omega(u) \\
&= \int_{u \in \mathcal{S}} (2\pi)^{-k/2} \|A^+ u\|^{-k} \int_{s=0}^{\infty} \exp\left(-\tfrac{1}{2} s^2\right) s^{k-1}\, ds\, d\omega(u).
\end{aligned}
$$

The singular values of A^+ are the multiplicative inverses of the singular values of X_k, hence the largest singular value of A^+ is at most $\frac{1+\delta_k}{1-\delta_k}$. In other words, $\|A^+ u\| \leq \frac{1+\delta_k}{1-\delta_k}$ for any unit vector u. Plugging this bound into the integral above, we find that

$$
\mathbf{Pr}\left[Aw \in \mathcal{C}\right] \geq \left(\tfrac{1-\delta_k}{1+\delta_k}\right)^k \int_{u \in \mathcal{S}} (2\pi)^{-k/2} \int_{s=0}^{\infty} \exp\left(-\tfrac{1}{2} s^2\right) s^{k-1}\, ds\, d\omega(u) = \frac{1}{k!} \left(\tfrac{1-\delta_k}{1+\delta_k}\right)^k,
$$

where the last equation is derived by observing that the integral is equal to the Gaussian measure of $\mathcal{C}$. Finally, by our choice of δ_k, we have $\left(\frac{1-\delta_k}{1+\delta_k}\right)^k > 1 - \delta$, which concludes the proof. □

B.1.4 Proofs for Uniform Distributions Over Small Support

Theorem 17. *Fix some $\xi \geq \frac{2(k+1)!}{\delta^2} \ln n$. If S is a random ξ-element multiset of permutations $\pi\colon [n] \to [n]$, then the uniform distribution over S fulfills the (k, δ)-UIOP with probability at least $1 - \frac{1}{n}$.*

Proof. We show this claim using the probabilistic method. Permutation $\pi_i\colon \mathcal{U} \to [n]$ is drawn uniformly at random from the set of all permutations with replacement. We claim that the set $S = \{\pi_1, \ldots, \pi_\xi\}$ fulfills the stated condition with probability at least $1 - \frac{1}{n}$.

Fix k distinct items $x_1, \ldots, x_k \in \mathcal{U}$. Let $Y_i = 1$ if $\pi_i(x_1) < \pi_i(x_2) < \ldots < \pi_i(x_k)$. As π_i is drawn uniformly from the set of all permutations, we have $\mathbf{Pr}\left[Y_i = 1\right] = \frac{1}{k!}$. That is, we have $\mathbf{E}\left[\sum_{i=1}^{\xi} Y_i\right] = \frac{\xi}{k!}$. As the random variables Y_i are independent, we can apply a Chernoff bound. This gives us

$$
\mathbf{Pr}\left[\sum_{i=1}^{\xi} Y_i \leq (1-\delta)\frac{\xi}{k!}\right] \leq \exp\left(-\frac{\delta^2}{2}\frac{\xi}{k!}\right) = n^{k+1} \quad .
$$

Note that if $\sum_{i=1}^{\xi} Y_i \leq (1-\delta)\frac{\xi}{k!}$ then the respective sequence $x_1, \ldots, x_k \in \mathcal{U}$ has probability at least $(1-\delta)\frac{1}{k!}$ when drawing one permutation at random from S.

There are fewer than n^k possible sequences. Therefore, applying a union bound, with probability at least $1-\frac{1}{n}$ the bound is fulfilled for all sequences simultaneously and therefore S fulfills the stated condition. □

Next we derive Theorem 6.

Lemma 7. *For some $\ell = \Omega(\log^2 \log\log n)$, there is a distribution over functions $f\colon \mathcal{U} \to [\ell]$ with entropy $O(\log\log n)$ such that for any $x, x' \in \mathcal{U}$, $x \neq x'$, we have $\mathbf{Pr}\left[f(x) = f(x')\right] = O(\frac{1}{\log\log\log n})$.*

Proof. We will define a function f, parameterized by α_1, α_2, and α_3, as a composition of 8 functions, which are mostly injective.

For $i = 1, 2, 3$, let $K_i = \log^{(i)} n$ and $q_i = K_i^2 + 1$. Let α_i be drawn independently uniformly from $[q_i - 1]$. This is the only randomization involved in the construction. It has entropy $\log(q_1 - 1) + \log(q_2 - 1) + \log(q_3 - 1)$.

Let C_i be a Reed-Solomon code of message length K_i and alphabet size q_i. This yields block length $N_i = q_i - 1$ and distance $d_i = N_i - K_i + 1 = q_i - K_i \geq \left(1 - \frac{1}{K_i}\right)(K_i^2 + 1)$. In other words, C_i is a function $C_i\colon D_i \to R_i$ with $D_i = [q_i]^{K_i}$ and $R_i = [q_i]^{N_i}$ such that for any $w, w' \in D_i$ with $w \neq w'$, we $C_i(w)$ and $C_i(w')$ differ in at least d_i components.

Furthermore α_i defines one position in each code word R_i. Given α_i, let $h_i\colon R_i \to [q_i]$, be the projection of a code word w of C_i to its α_ith component, i.e., $h_i(w) = w_{\alpha_i}$.

Finally, we observe that $|D_{i+1}| = q_{i+1}^{K_{i+1}} \geq 5^{K_{i+1}} \geq (2^{K_{i+1}})^2 + 1 = q_i$. So there is an injective mapping $g_i\colon [q_i] \to D_{i+1}$, mapping alphabet symbols of C_i to messages of C_{i+1}.

Overall, this defines a function $f = h_3 \circ C_3 \circ g_2 \circ h_2 \circ C_2 \circ g_1 \circ h_1 \circ C_1$, mapping values of D_1 to $[q_3]$.

Let $f_i = g_i \circ h_i \circ C_i \circ f_{i-1}$.

Now let $w, w' \in D_1$, $w \neq w'$. Observe that all functions except for the h_i are injective. Therefore the event $f(w) = f(w')$ can only occur if $h_i(C_i(f_{i-1}(w))) = h_i(C_i(f_{i-1}(w')))$ for some i. As C_i is a Reed-Solomon code with distance d_i, $C_i(f_{i-1}(w))$ and $C_i(f_{i-1}(w'))$ differ in at least d_i components. Therefore, the probability that $h_i(C_i(f_{i-1}(w))) \neq h_i(C_i(f_{i-1}(w')))$ is at least $\frac{d_i}{N_i}$.

By union bound, the combined probability that this does not hold for one i is bounded by

$$\mathbf{Pr}\left[\bigwedge_{i=1}^{3} h_i(C_i(f_{i-1}(w))) = h_i(C_i(f_{i-1}(w')))\right] \leq \sum_{i=1}^{3}\left(1 - \frac{d_i}{N_i}\right) \leq 3\left(1 - \frac{d_3}{N_3}\right) \leq \frac{3}{K_3} \quad .$$

□

Theorem 18. *There is a distribution over permutations that has entropy $O(\log\log n)$ and fulfills the (k, δ)-uniform-induced-ordering property where $\delta = O(\frac{k^2}{\log\log\log n})$.*

Proof. By the above lemma, there are constants c_1, c_2, c_3 such that the following condition is fulfilled. For some $\ell = c_1 \log^2 \log n$, there is a distribution over functions $f\colon \mathcal{U} \to [\ell]$ with entropy $c_2 \log\log n$ such that for any $x, x' \in \mathcal{U}$, $x \neq x'$, we have $\mathbf{Pr}\left[f(x) = f(x')\right] \leq \frac{c_3}{\log\log\log n}$.

Draw a permutation $\pi'\colon [\ell] \to [\ell]$ uniformly at random and define the permutation $\pi\colon \mathcal{U} \to [n]$ by using $\pi' \circ f$ and extending it to a full permutation arbitrarily.

Let $x_1, \ldots, x_k$ be distinct items from $\mathcal{U}$. Conditioned on $f(x_i) \neq f(x_j)$ for all $i \neq j$, we have $\pi(x_1) < \pi(x_2) < \ldots < \pi(x_k)$ with probability $\frac{1}{k!}$. Furthermore, applying a union bound in combination with the above lemma, the probability that there is some pair $i \neq j$, with $f(x_i) = f(x_j)$ is at most $k^2 \frac{c_2}{\log\log\log n}$. Therefore, the overall probability that $\pi(x_1) < \pi(x_2) < \ldots < \pi(x_k)$ is at least $(1 - \frac{c_2 k^2}{\log\log\log n})\frac{1}{k!}$.

The entropy of the distribution that determines π is $c_2 \log\log n + \log(\ell!) = O(\log\log n)$. □

B.2 Proofs deferred from §3

In this section we restate some of the results from §3 and provide complete proofs.

To show Theorem 7, we first give a bound in terms of the support size of the distribution. Later on, we will show how this transfers to a bound on the entropy.

Lemma 8. *If $\pi\colon \mathcal{U} \to [n]$ is chosen from a distribution of support size at most k, then any algorithm's probability of success against a worst-case adversary is at most $\frac{k+1}{\log n}$.*

Recall from §3 the notion of a *semitone* sequence and Lemma 1, which shows that any k permutations of $[n]$ have a sequence of length $\frac{\log n}{k+1}$ that is semitone with respect to all k permutations simultaneously.

We now turn to showing that an adversary can exploit a semitone sequence and force any algorithm to only have $\frac{1}{s}$ probability of success. To show this we look at the performance of the best deterministic algorithm against a particular distribution over assignment of values to items.

Lemma 9. *Let $\mathcal{V} = \{1, 2, \ldots, s\}$. Assign values from $\mathcal{V}$ to items $(x_1, \ldots, x_s)$ at random by*

$$\textit{value}(x_s) = \begin{cases} \max(\mathcal{V}) & \textit{with probability } 1/s \\ \min(\mathcal{V}) & \textit{with probability } 1 - 1/s \end{cases} \tag{10}$$

and then assigning values from $\mathcal{V} \setminus \{\textit{value}(x_s)\}$ to items $(x_1, \ldots, x_{s-1})$ recursively. Assign a value 0 to all other items.

Consider an arbitrary algorithm following permutation π such that $(x_1, \ldots, x_s)$ is semitone with respect to π. This algorithm selects the best item with probability at most $\frac{1}{s}$.

Proof. Fixing some (deterministic) algorithm and permutation π, let $\mathcal{A}_t$ be the event that the algorithm selects any item among $x_1, \ldots, x_t$ and let $\mathcal{B}_t$ be the event that the algorithm selects the best item among $x_1, \ldots, x_t$. We will show by induction that $\mathbf{Pr}[\mathcal{B}_t] = \frac{1}{t}\mathbf{Pr}[\mathcal{A}_t]$. This will imply $\mathbf{Pr}[\mathcal{B}_s] = \frac{1}{s}\mathbf{Pr}[\mathcal{A}_s] \leq \frac{1}{s}$.

For $t = 1$ this statement trivially holds. Therefore, let us consider some $t > 1$. By induction hypothesis, we have $\mathbf{Pr}[\mathcal{B}_{t-1}] = \frac{1}{t-1}\mathbf{Pr}[\mathcal{A}_{t-1}]$. As $(x_1, \ldots, x_t)$ is semitone with respect to π, x_t either comes before or after all $x_1, \ldots, x_{t-1}$. We distinguish these two cases.

Case 1: x_t comes before all $x_1, \ldots, x_{t-1}$. The algorithm can decide to accept x_t (without seeing the items $x_1, \ldots, x_{t-1}$). In this case, we have $\mathcal{A}_t$ for sure. We only have $\mathcal{B}_t$ if x_t gets a higher value than $x_1, \ldots, x_{t-1}$. By definition this happens with probability $\frac{1}{t}$. So, we have $\mathbf{Pr}[\mathcal{B}_t] = \frac{1}{t}\mathbf{Pr}[\mathcal{A}_t]$. The algorithm can also decide to reject x_t. Then $\mathcal{A}_t$ if and only if $\mathcal{A}_{t-1}$. Furthermore, $\mathcal{B}_t$ if and only if $\mathcal{B}_{t-1}$ and x_t does not get the highest value among $x_1, \ldots, x_t$. These events are independent, so $\mathbf{Pr}[\mathcal{B}_t] = (1 - \frac{1}{t})\mathbf{Pr}[\mathcal{B}_{t-1}]$. Applying the induction hypothesis, we get $\mathbf{Pr}[\mathcal{B}_t] = (1 - \frac{1}{t})\mathbf{Pr}[\mathcal{B}_{t-1}] = \frac{t-1}{t}\frac{1}{t-1}\mathbf{Pr}[\mathcal{A}_{t-1}] = \frac{1}{t}\mathbf{Pr}[\mathcal{A}_t]$.

Case 2: x_t comes after all $x_1, \ldots, x_{t-1}$. When the algorithm comes to x_t, it may or may not have selected an item so far. If it has already selected an item ($\mathcal{A}_{t-1}$), then this element is the best among $x_1, \ldots, x_t$ with probability $\mathbf{Pr}[\mathcal{B}_{t-1} \mid \mathcal{A}_{t-1}] = \frac{1}{t-1}$ by induction hypothesis. Independent of these events, x_t is worse than the best items among $x_1, \ldots, x_{t-1}$ with probability $1 - \frac{1}{t}$. Therefore, we get $\mathbf{Pr}[\mathcal{B}_t \mid \mathcal{A}_{t-1}] = \frac{1}{t-1}\frac{t-1}{t} = \frac{1}{t}$. It remains the case that the algorithm selects item x_t ($\mathcal{A}_t \setminus \mathcal{A}_{t-1}$). This item is the better than $x_1, \ldots, x_{t-1}$ with probability $\frac{1}{t}$. That is, $\mathbf{Pr}[\mathcal{B}_t \mid \mathcal{A}_t \setminus \mathcal{A}_{t-1}] = \frac{1}{t}$. In combination, we have $\mathbf{Pr}[\mathcal{B}_t] = \mathbf{Pr}[\mathcal{A}_{t-1}]\mathbf{Pr}[\mathcal{B}_t \mid \mathcal{A}_{t-1}] + \mathbf{Pr}[\mathcal{A}_t \setminus \mathcal{A}_{t-1}]\mathbf{Pr}[\mathcal{B}_t \mid \mathcal{A}_t \setminus \mathcal{A}_{t-1}] = \mathbf{Pr}[\mathcal{A}_{t-1}]\frac{1}{t} + \mathbf{Pr}[\mathcal{A}_t \setminus \mathcal{A}_{t-1}]\frac{1}{t} = \frac{1}{t}\mathbf{Pr}[\mathcal{A}_t]$. □

Lemmas 1 and 9 now imply Lemma 8.

Proof of Lemma 8. Let Π, $|\Pi| \leq k$, be the support of the distribution π is drawn from. Lemma 1 shows that there is a sequence $(x_1, \ldots, x_s)$ of length $s = \frac{\log n}{k+1}$ that is semitone with respect to any permutation in π.

It only remains to apply Yao's principle: Instead of considering the performance of a random π against a deterministic adversary, we consider the performance of a fixed π against a randomized adversary. Lemma 9 shows that there is a distribution over instances such that no $\pi \in \Pi$ has success probability better than $\frac{1}{s} = \frac{\log n}{k+1}$. □

To get a bound on the entropy, we show that for a low-entropy distribution there is a small subset of the support that is selected with high probability.

Lemma 10. *Let a be drawn from a finite set $\mathcal{D}$ by a distribution of entropy H. Then for any $k \geq 4$ there is a set $T \subseteq \mathcal{D}$, $|T| \leq k$, such that $\mathbf{Pr}[a \in T] \geq 1 - \frac{8H}{\log(k-3)}$.*

Proof. Set $\alpha = \frac{H}{\log(k-3)}$ and $\beta = \frac{\alpha}{k-3}$. Note that for $\alpha \geq \frac{1}{8}$, the statement becomes trivial. Therefore, we can assume without loss of generality that $\alpha < \frac{1}{8}$. This implies $\log(\alpha) < 0$. Therefore, we get

$$\frac{H}{-\log \beta} = \frac{\alpha \log(k-3)}{\log(k-3) - \log(\alpha)} \leq \alpha \ .$$

Let $a_1, \ldots, a_k$ be the elements of $\mathcal{D}$ such that $p_{a_i} \geq \beta$ for all i and $p_{a_1} \geq p_{a_2} \geq \ldots \geq p_{a_k}$. Furthermore, partition $\mathcal{D} \setminus \{a_1, \ldots, a_k\}$ into $S_1, \ldots, S_\ell$ such that $p_{S_i} \in [\beta, 2\beta)$ for $i < \ell$, $p_{S_\ell} < 2\beta$

Observe that $p_{a_3} \leq \frac{1}{\mathrm{e}}$ because probabilities sum up to at most 1. Therefore, for $i \geq 3$, we have $-p_{a_i} \log(p_{a_i}) \geq -\beta \log \beta$ by monotonicity. Furthermore, for all $j < \ell$, we have $-p_{S_j} \log(p_{S_j}) \geq -\beta \log \beta$. In combination, this gives us

$$H \geq \sum_{i=3}^{k} -p_{a_i} \log(p_{a_i}) + \sum_{j=1}^{\ell-1} -p_{S_i} \log(p_{S_i}) \geq (k + \ell - 3)(-\beta \log \beta) \ .$$

For k and ℓ, this implies

$$k \leq \frac{H}{-\beta \log \beta} + 3 \leq \frac{\alpha}{\beta} + 3 \leq k \quad \text{and} \quad \ell \leq \frac{H}{-\beta \log \beta} + 3 \leq \frac{\alpha}{\beta} + 3 \ .$$

In conclusion, we have

$$\sum_{j=1}^{\ell} p_{S_j} \leq 2\beta\ell \leq 2\alpha + 6\beta \leq 8\alpha \ .$$

□

Now Theorem 7 is proven as a combination of Lemma 8 and Lemma 10.

Proof of Theorem 7. Set $k = \sqrt{\log n}$. Lemma 10 shows that there is a set of permutations Π of size at least k that is chosen with probability at least $1 - \frac{8H}{\log(k-3)}$. The distribution conditioned on π being in Π has support size only k. Lemma 8 shows that if π is chosen by a distribution of support size k, then the probability of success of any algorithm against a worst-case adversary is at most $\frac{k+1}{\log n}$. Therefore, we get

$$\begin{aligned}
\mathbf{Pr}[\text{success}] &= \mathbf{Pr}[\pi \in \Pi]\, \mathbf{Pr}[\text{success} \mid \pi \in \Pi] + \mathbf{Pr}[\pi \notin \Pi]\, \mathbf{Pr}[\text{success} \mid \pi \notin \Pi] \\
&\leq \mathbf{Pr}[\text{success} \mid \pi \in \Pi] + \mathbf{Pr}[\pi \notin \Pi] \\
&\leq \frac{k+1}{\log n} + \frac{8H}{\log(k-3)} \\
&= o(1) \ .
\end{aligned}$$

□

B.3 Proofs deferred from §4

In this section we restate some of the results from §4 and provide complete proofs.

Lemma 11. *If $g : \{0,1\}^n \to [k]$ is a random function, then with high probability there is no circuit of size $s(n) = 2^n/(8kn)$ that outputs the function value correctly on more than $\frac{2}{k}$ fraction of inputs.*

Proof. The proof closely parallels the proof of the corresponding statement for worst-case hardness rather than hardness-on-average, which is presented, for example, in the textbook by Arora and Barak [3]. The number of Boolean circuits of size s is bounded by s^{3s}. For any one of these circuits, C, the expected number of inputs x such that $C(x) = g(x)$ is $\frac{1}{k} \cdot 2^n$. Since the events $\{C(x) = g(x)\}$ are mutually independent as x varies over $\{0,1\}^n$, the Chernoff bound (e.g., [33]) implies that the probability of more than $\frac{2}{k} \cdot 2^n$ of these events taking place is less than $\exp\left(-\frac{1}{3k} \cdot 2^n\right)$. The union bound now implies that the probability there exists a circuit C of size s that correctly computes g on more than $\frac{2}{k}$ fraction of inputs is bounded above by $\exp\left(3s\ln(s) - \frac{1}{3k} \cdot 2^n\right)$. When $s = 2^n/(8kn)$ this yields the stated high-probability bound. □

In the sequel we will need a version of the lemma above in which the circuit, rather than being constructed from the usual Boolean gates, is constructed from t different types of gates, each having m binary inputs and one binary output.

Lemma 12. *Suppose we are given t types of gates, each computing a specific function from $\{0,1\}^m$ to $\{0,1\}$. If $g : \{0,1\}^n \to [k]$ is a random function, then with high probability there is no circuit of size $s(n) \le 2^n/(8k \cdot \max\{mn, \ln(t)\})$ that outputs the function value correctly on more than $\frac{2}{k}$ fraction of inputs.*

Proof. The proof is the same except that the number of circuits, rather than being bounded by s^{3s}, is now bounded by $(ts^m)^s = \exp(s\ln t + ms\ln s)$. The stated high-probability bound continues to hold if $ms\ln s < \frac{1}{8k} \cdot 2^n$ and $s\ln t < \frac{1}{8k} \cdot 2^n$. The assumption $s(n) \le 2^n/(8k \cdot \max\{mn, \ln(t)\})$ justifies these two inequalities and completes the proof. □

B.3.1 Proof of Theorem 9

Similar to the proof of Theorem 8, our plan for proving Theorem 9 is to construct a distribution over arrival orderings, $\underline{\pi}$, in which the first half of the input sequence attempts to encode the position where the maximum-value item occurs in the second half of the permutation. What makes the proof more difficult is that the adversary chooses the ordering of items by value (as represented by a permutation $\sigma \in S_n$), and this ordering could potentially be chosen to thwart the decoding process. In our construction we will make a distinction between *decodable* orderings—whose properties will guarantee that our decoding algorithm succeeds in finding the maximum-value item—and *non-decodable* orderings, which may lead the decoding algorithm to make an error. We will then design a separate algorithm that succeeds with constant probability when the adversary's ordering is non-decodable.

We will assume throughout the proof that n is divisible by 8, for convenience. Recall, also, that the theorem statement declares $\kappa(n)$ to be any function of n such that $\lim_{n\to\infty} \kappa(n) = 0$ while $\lim_{n\to\infty} \frac{n\cdot\kappa(n)}{\log n} = \infty$. For convenience we adopt the notation $[a,b]$ to denote the subset of $[n]$ consisting of integers in the range from a to b, inclusive; analogously, we may refer to subsets of $[n]$ using open or half-open interval notation.

Definition 6. A total ordering of $[n]$ is *decodable* if it satisfies the following properties.

1. The maximal element of the ordering is n.
2. When the elements of the set $(\frac{n}{4}, \frac{n}{2}]$ are written in decreasing order with respect to the total ordering, the of the first $n \cdot \kappa(n)$ elements that belong to $(\frac{3n}{8}, \frac{n}{2}]$ is at least $\frac{39}{40}$.

Our proof will involve the construction of three distributions over permutations, and three corresponding algorithms.

- an "encrypting" distribution $\underline{\pi}_e$ that hides item n in the second half of the permutation while arranging the first half of the permutation to form a "clue" that reveals the location of item n, but does so in a way that cannot be decrypted by small circuits;
- a first "adversary-coercing" distribution $\underline{\pi}_{c,1}$ that forces the adversary to make item n the most valuable item;
- a second "adversary-coercing" distribution $\underline{\pi}_{c,2}$ that forces the adversary to satisfy the second property in the definition of a decodable ordering.

Corresponding to these three distributions we will define algorithms $\text{ALG}_e, \text{ALG}_{c,1}, \text{ALG}_{c,2}$ such that:

- ALG_e has constant probability of correct selection when the adversary chooses a decodable ordering;
- $\text{ALG}_{c,1}$ has constant probability of correct selection when the adversary chooses an ordering that violates the first property of decodable orderings;
- $\text{ALG}_{c,2}$ has constant probability of correct selection when the adversary chooses an ordering that satisfies the first property of decodable orderings but violates the second.

Combining these three statements, one can easily conclude that when nature samples the arrival order using the permutation distribution $\underline{\pi} = \frac{1}{3}(\underline{\pi}_e + \underline{\pi}_{c,1} + \underline{\pi}_{c,2})$, and when the algorithm ALG is the one that randomizes among the three algorithms $\{\text{ALG}_e, \text{ALG}_{c,1}, \text{ALG}_{c,2}\}$ with equal probability, then ALG has constant probability of correct selection no matter how the adversary orders items by value.

We begin with the construction of the distribution $\underline{\pi}_{c,1}$ and algorithm $\text{ALG}_{c,1}$. The following describes the procedure of drawing a random sample from $\underline{\pi}_{c,1}$.

Algorithm 1 Sampling procedure for $\underline{\pi}_{c,1}$

1: Sample an $\left(\frac{n}{2}\right)$-element set $L \subset [n-1]$ uniformly at random.
2: Let $R = [n-1] \setminus L$.
3: Let π' by the permutation that lists the elements of L in increasing order, followed by the elements of R increasing order, followed by n.
4: Choose a uniformly random $i \in \left[1, \frac{n}{2}\right]$ and let τ_i be the transposition that swaps elements n and i. (If $i = n$ then τ_i is the identity permutation.)
5: Let $\pi = \tau_i \circ \pi'$.

Define $\text{ALG}_{c,1}$ to be an algorithm that observes the first $\frac{n}{2}$ elements, sets a threshold equal to the maximum of the observed elements, and selects the next element whose value exceeds this threshold. In the following lemma and for the remainder of this section, ρ denotes the permutation that lists the items in order of increasing value, i.e., $\rho(i) = n - i$.

Lemma 13. *If the adversary's ordering σ assigns the maximum value to any item other than n, then $V^{\underline{\pi}_{c,1}}(\text{ALG}_{c,1}, \sigma) > \frac{1}{4}$. On the other hand, $V^{\underline{\pi}_{c,1}}(*, \rho) = \frac{2}{n}$.*

Proof. Suppose that σ assigns the maximum value to item $i \neq n$, and suppose that item j receives the second-largest value among items in $[n-1]$. In the sampling procedure for $\underline{\pi}_{c,1}$, the event that $j \in L$ and $i \in R$ has probability

$$\frac{n/2}{n-1} \cdot \frac{(n/2)-1}{n-2} > \frac{1}{4},$$

and when this event happens the algorithm $\text{ALG}_{c,1}$ is guaranteed to select item i.

To prove the second part of the lemma, suppose the adversary assigns values to items in increasing order and observe that this guarantees that the first $n/2$ items in the permutation π' (defined in Step 3 of the sampling procedure) are listed in increasing order of value, and that the first $n/2$ items in π are the same except that the value at index i is replaced by the maximum value. Now consider any algorithm and let t denote the time when it makes its selection when facing a monotonically increasing sequence of n values. If $t > n/2$, then the algorithm assuredly makes an incorrect selection when facing the input sequence $\pi\rho$. If $t \leq n/2$, then the algorithm makes a correct selection if and only if t matches the random index i chosen in the sampling procedure for π, an event with probability $2/n$. □

We next present the construction of $\underline{\pi}_{c,2}$.

Algorithm 2 Sampling procedure for $\underline{\pi}_{c,2}$

```
With probability 1/2, reverse the order of the first n/4 items in the list.
Initialize I = ∅.
for i = 1, ..., n/4 do
    With probability 1/(n κ(n)):
        Swap the items in positions i and i + n/4.
        Add i to the set I.
        If i > n/8 then add i into the set I⁺.
end for
Let π′ denote the permutation of items defined at this point in the procedure.
if I⁺ is non-empty then
    Choose a uniformly random index i ∈ I⁺
else
    Choose a uniformly random index i ∈ (n/8, n/4].
end if
Let τ_i be the transposition that swaps elements n and i.
Let π = τ_i ∘ π′.
```

Define $\text{ALG}_{c,2}$ to be an algorithm that observes the first $\frac{n}{8}$ elements, sets a threshold equal to the maximum of the observed elements, and selects the next element whose value exceeds this threshold.

Lemma 14. *If the adversary's ordering σ assigns the maximum value to item n but violates Property 2 in the definition of a decodable ordering, then $V^{\underline{\pi}_{c,2}}(\text{ALG}_{c,2}, \sigma) > \frac{1}{250}$. On the other hand, $V^{\underline{\pi}_{c,1}}(*, \rho) = O(\kappa(n))$.*

Proof. First suppose that σ assigns the maximum value to item n but violates Property 2 in the definition of a decodable ordering. To prove that $V^{\underline{\pi}_{c,2}}(\text{ALG}_{c,2}, \sigma) > \frac{1}{250}$, note first that $\text{ALG}_{c,2}$ is guaranteed to make a correct selection if the permutation π' (defined in step 9 of the sampling procedure) has the property that the highest-value item found among the first $n/4$ positions in π' belongs to one of the first $\frac{n}{8}$ positions. Recalling the set I defined in the sampling procedure, let $J = \left[1, \frac{n}{4}\right] \setminus I$ and let $K = \left\{i + \frac{n}{4} \mid i \in I\right\}$. Note that $J \cup K$ is the set of items found among the first $n/4$ positions in π'. Let j, k denote the highest-value elements of J and K, respectively. (If K is empty then k is undefined.) Step 1 of the sampling procedure ensures that with probability $\frac{1}{2}$, item j belongs to one of the first $\frac{n}{8}$ positions, and this event is independent of the event that K is non-empty and k belongs to one of the first $\frac{n}{8}$ positions. To complete the proof, we now bound the probability of that event from below by $\frac{1}{100}$.

Let $i_1, i_2, \ldots, i_{n/4}$ denote a listing of the elements of the set $\left(\frac{n}{4}, \frac{n}{2}\right]$ in decreasing order of value. For $1 \leq \ell \leq \frac{n}{4}$, the probability that $k = i_\ell$ is $\left(1 - \frac{1}{n\,\kappa(n)}\right)^{\ell-1} \frac{1}{n\,\kappa(n)}$. Let L denote the set of $\ell \leq n\,\kappa(n)$ such that $i_\ell \leq \frac{3n}{8}$. By our hypothesis that σ violates Property 2 in the definition of a decodable ordering, we know that $\frac{|L|}{n\,\kappa(n)} > \frac{1}{40}$. If $k = i_\ell$ for any $\ell \in L$, then k belongs to one of the first $\frac{n}{8}$ positions in π'. The probability of this event is

$$\sum_{\ell \in L} \mathbf{Pr}\left[k = i_\ell\right] = \sum_{\ell \in L} \left(1 - \frac{1}{n\,\kappa(n)}\right)^{\ell-1} \frac{1}{n\,\kappa(n)} \geq \frac{1}{n\,\kappa(n)} \sum_{\ell \in L} \left(1 - \frac{1}{n\,\kappa(n)}\right)^{n\,\kappa(n)-1} \geq \frac{|L|}{n\,\kappa(n)} \cdot \frac{1}{e} > \frac{1}{100},$$

as desired.

The second half of the lemma asserts that $V^{\underline{\pi}_{c,1}}(*, \rho) = O(\kappa(n))$, where ρ denotes the permutation that lists the items in order of increasing value. To prove this, first recall the set I^+ defined in the sampling procedure; note that $|I^+|$ is equal to the number of successes in $\frac{n}{8}$ i.i.d. Bernoulli trials with success probability $\frac{1}{n\,\kappa(n)}$. Hence $\mathbf{E}\left[|I^+|\right] = \frac{1}{8\kappa(n)}$ and, by the Chernoff Bound,

$$\mathbf{Pr}\left[|I^+| < \tfrac{1}{16\kappa(n)}\right] < \exp\left(-\frac{1}{128\kappa(n)}\right) < 128\kappa(n).$$

Conditional on the event that $|I^+| \geq \frac{1}{16\kappa(n)}$, the conclusion of the proof is similar to the conclusion of the proof of Lemma 13. Consider any algorithm and let t denote the time when it makes its selection when the items are presented in the order π'. Also, let s denote the time when item n is presented in the order π. The first s items in π and π' have exactly the same relative ordering by value since, by construction, $s \in I^+$ and hence the element that arrives at time s in π' has the maximum value observed so far. Hence, the algorithm makes a correct selection only when $t = s$. However, if $t \notin I^+$ then this event does not happen, and if $t \in I^+$ the event $t = s$ happens only if s is the random index i selected in Step 11, an event whose probability is $1/|I^+|$, which is at most $16\kappa(n)$ since we are conditioning on $|I^+| \geq \frac{1}{16\kappa(n)}$. Combining our bounds for the cases $|I^+| < \frac{1}{16\kappa(n)}$ and $|I^+| \geq \frac{1}{16\kappa(n)}$, we find that for any algorithm ALG,

$$\begin{aligned}
\mathrm{PCS}(\textsc{alg}, \pi\rho) &\leq \mathbf{Pr}\left[|I^+| < \tfrac{1}{16\kappa(n)}\right] + \mathbf{Pr}\left[|I^+| \geq \tfrac{1}{16\kappa(n)}\right] \cdot \mathbf{Pr}\left[\text{correct selection} \;\middle|\; |I^+| \geq \tfrac{1}{16\kappa(n)}\right] \\
&\leq \mathbf{Pr}\left[|I^+| < \tfrac{1}{16\kappa(n)}\right] + \mathbf{Pr}\left[\text{correct selection} \;\middle|\; |I^+| \geq \tfrac{1}{16\kappa(n)}\right] \\
&< 128\kappa(n) + 16\kappa(n),
\end{aligned}$$

as desired. □

Finally, we present the construction of the permutation distribution $\underline{\pi}_e$. A crucial ingredient is a coding-theoretic construction that may be of independent interest.

Definition 7. We say that a function $A : \{0,1\}^k \to \{0,1\}^n$ has half-unique-decoding radius r if at least half of the inputs $x \in \{0,1\}^k$ satisfy the property that for all $x' \neq x$, the Hamming distance from $A(x)$ to $A(x')$ is greater than $2r$.

Codes with half-unique-decoding radius r are useful because they allow a receiver to decode messages with probability at least $\frac{1}{2}$, in a model with random messages and adversarial noise. The following easy lemma substantiates this interpretation. Here and subsequently, we use $\|y - y'\|$ to denote the Hamming distance between strings y, y'.

Lemma 15. *Suppose that:*

- *a uniformly random string $x \in \{0,1\}^k$ is encoded using a function A whose half-unique-decoding radius is r,*
- *an adversary is allowed to corrupt any r bits of the resulting codeword, and*
- *a decoding algorithm receives the corrupted string $\hat{y}$, finds the nearest codeword (breaking ties arbitrarily), and applies the function A^{-1} to produce an estimate $\hat{x}$ of the original message.*

The $\mathbf{Pr}[\hat{x} = x] \geq \frac{1}{2}$ *regardless of the adversary's policy for corrupting the transmitted codeword.*

Proof. Let $y = A(x)$. The constraint on the adversary implies that $\|y - \hat{y}\| \leq r$. The definition of half-unique-decoding radius implies that, with probability at least $\frac{1}{2}$ over the random choice of x, the nearest codeword to y is at Hamming distance greater than $2r$. By the triangle inequality, this event implies that y is the unique nearest codeword to $\hat{y}$, in which case the decoder succeeds. □

The particular coding construction that our proof requires is a code with the property that, roughly speaking, all of its low-dimensional projections have large half-unique-decoding radius. The following definition and lemma make this notion precise.

Definition 8. For $S \subseteq [n]$, let $\text{proj}_S : \{0,1\}^n \to \{0,1\}^{|S|}$ denote the function that projects a vector onto the coordinates indexed by S. In other words, letting $(i_1, i_2, \ldots, i_s)$ denote a sequence containing each element of S once, we define $\text{proj}_S(y) = (y_{i_1}, y_{i_2}, \ldots, y_{i_s})$. For any function $A : \{0,1\}^k \to \{0,1\}^n$, we introduce the notation A_S to denote the composition $\text{proj}_S \cdot A : X \to \{0,1\}^{|S|}$.

Lemma 16. *For all sufficiently large m, if $2m \leq n < \frac{m}{2} \cdot 2^{2^{m-4}}$, there exists a function $A : \{0,1\}^m \to \{0,1\}^n$ such that for every set $S \subseteq [n]$ of cardinality $2m$, the function A_S has half-unique-decoding radius $\frac{m}{10}$.*

Proof. We prove existence of A using the probabilistic method, by showing that a uniformly random function $A : \{0,1\}^m \to \{0,1\}^n$ has the property with positive probability. To do so, we need to estimate the probability, for a given set S, that A_S fails to have half-unique-decoding radius $\frac{m}{10}$.

Define a graph G_S with vertex set $\{0,1\}^m$ by drawing an edge between every two vertices x, x' such that $\|A_S(x) - A_S(x')\| \leq \frac{m}{5}$. The event that A_S has half-unique-decoding radius $\frac{m}{10}$ corresponds precisely to the event that G_S has at least 2^{m-1} isolated vertices. When this event does not happen, the number of connected components in G_S is at most $2^m - 2^{m-2}$, so a spanning forest of G_S has at least 2^{m-2} edges.

Our plan is to bound—for every set $S \subseteq [n]$ of size $2m$ and every forest F with 2^{m-2} edges—the probability that G_S contains all the edges of F. Summing over S and F we will find the sum is less than 1, which implies, by the union bound, that with positive probability over the random choice of A no such pair (S, F) exists. By the arguments in the preceding paragraph, it follows that when no such pair (S, F) exists the half-unique-decoding radius of A_S is $\frac{m}{10}$ for every S of size $2m$, yielding the lemma.

To begin, let us fix $x, x' \in \{0,1\}^m$ and $S \subseteq [n]$ with $|S| = 2m$, and let us estimate the probability that $\|A_S(x) - A_S(x')\| \leq \frac{m}{5}$. The strings $A_S(x)$ and $A_S(x')$ are independent uniformly-random binary strings of length $2m$. The number of binary strings within Hamming distance $\frac{m}{5}$ of $A_S(x)$ is bounded above by $2^{(1+o(1)) \cdot H(1/10) \cdot 2m}$, where $H(p)$ denotes the binary entropy function $-p\log_2(p) - (1-p)\log_2(1-p)$. Using the fact that $2H(\frac{1}{10}) < 0.95$ we can conclude that for large enough m, fewer than $2^{(0.95)m}$ binary strings belong to the Hamming ball of radius $\frac{m}{5}$ around $A_S(x)$. Hence the probability that $A_S(x')$ is one of these strings is less than $2^{-m/20}$. If F is the edge set of a forest on vertex set $\{0,1\}^m$, then the random variables $A_S(x) - A_S(x')$ are mutually independent as (x, x') ranges over the edges of F. Consequently the probability that all the edges of F are contained in G_S is less than $\left(2^{-m/20}\right)^{|F|}$.

Let $N = 2^m$. The number of spanning trees of an N-element vertex set is N^{N-2} [8, 11] and the number of forests with $N/4$ edges contained in any one such tree is $\binom{N}{N/4}$. Thus, the number of pairs (S, F) where $S \subseteq [n]$ has $2m$ elements and F is the edge set of a forest with vertex set $\{0,1\}^m$ is bounded above by $\binom{n}{2m}\binom{N}{N/4}N^{N-2}$. Applying the union bound, we conclude that the probability of failure for our construction is bounded above by

$$\binom{n}{2m}\binom{N}{N/4}N^{N-2}\left(2^{-m/20}\right)^{N/4} < \left(\tfrac{2n}{m}\right)^{2m} 2^N N^N \left(2^{-m/20}\right)^{N/4},$$

where we have used the inequalities $\binom{n}{k} \leq \left(\frac{4n}{k}\right)^k$ (valid for all $0 \leq k \leq n$) and $\binom{N}{N/4} \leq 2^N$ (valid for all N). The base-2 logarithm of the probability of failure is bounded above by

$$2m[1 + \log(n) - \log(m)] + N\left[1 + \log(N) - \tfrac{m}{80}\right].$$

(All logs are base 2.) Substituting $N = 2^{m-2}$ and rearranging terms, we find that this expression is negative (i.e., the probability of failure is strictly less than 1) when

$$\log(n) < \log(m) - 1 + \frac{2^{m-3}}{m}\left[\frac{79m}{80} - 1\right] < \log(m) - 1 + 2^{m-4},$$

provided $m > 2$. This inequality is satisfied when $n < \frac{m}{2} \cdot 2^{2^{m-4}}$, which completes the proof. □

We now continue with the construction of the permutation distribution $\underline{\pi}_e$. Let $m = \left\lfloor \frac{1}{2} n\,\kappa(n) \right\rfloor$. By Lemma 16 there exists a function $A : \{0,1\}^m \to \{0,1\}^{n/8}$ such that for all $S \subseteq [n]$ with $|S| = 2m$, the half-unique-decoding radius of A_S is $\frac{m}{10}$. Let us choose one such function A for the remainder of the construction. Define an A-*augmented circuit* to be a circuit constructed from the usual AND, OR, NOT gates along with $n/8$ additional types of gates that take an m-bit input x and output one of the bits of $A(x)$. By Lemma 12 there exists a function $g : \{0,1\}^m \to \left[\frac{n}{2}\right]$ such that no A-augmented circuit of size $s(n) < 2^m/(4m^2 n)$ computes the value of g correctly on more than $\frac{4}{n}$ fraction of inputs. Let us choose one such function and denote it by g for the remainder of the construction. (To justify the application of Lemma 16, note that our assumption that $n\,\kappa(n)/\log(n) \to \infty$ implies $\frac{n}{8} < 2^{2^{m-4}}$ for all sufficiently large n.) Armed with the functions g and A we are ready to present the construction of $\underline{\pi}_e$.

Algorithm 3 Sampling procedure for $\underline{\pi}_e$

Sample $x \in \{0,1\}^m$ uniformly at random.
Let $y = A(x) \in \{0,1\}^{n/8}$.
for $i = 1, \ldots, \frac{n}{8}$ **do**
    if $y_i = 1$ **then**
        Swap the items in positions $\frac{3n}{8} + i$ and $\frac{n}{4} + i$.
    else
        Leave the permutation unchanged.
    end if
end for
Swap the items in positions n and $\frac{n}{2} + g(x)$.

The corresponding algorithm ALG_e works as follows.

Algorithm 4 Algorithm ALG_e

Observe the first $\frac{n}{2}$ elements of the input sequence.
Let J denote the set of items with arrival times in the interval $\left(\frac{n}{4}, \frac{n}{2}\right]$, i.e. $J = \pi^{-1}\left(\frac{n}{4}, \frac{n}{2}\right]$.
Let $j_1, \ldots, j_{n/4}$ denote a listing of the elements of J in order of decreasing value.
for $\ell = 1, \ldots, 2m$ **do**
    if $\pi(j_\ell) \leq \frac{3n}{8}$ **then**
        Set $\hat{y}_\ell = 1$ and $i_\ell = \pi(j_\ell) - \frac{n}{4}$.
    else
        Set $\hat{y}_\ell = 0$ and $i_\ell = \pi(j_\ell) - \frac{3n}{8}$.
    end if
end for
Set $S = (i_1, i_2, \ldots, i_{2m})$.
Find the $x \in \{0,1\}^m$ that minimizes $\|A_S(x) - \hat{y}\|$, breaking ties arbitrarily.
Select the item that arrives at time $t = \frac{n}{2} + g(x)$.

Lemma 17. *If the adversary's ordering σ is a decodable ordering, then $V^{\underline{\pi}_e}(\text{ALG}_e, \sigma) \geq \frac{1}{2}$. On the other hand, for any algorithm ALG_p whose stopping rule can be computed by circuits of size $s(n) = 2^{n\,\kappa(n)/4}$, we have $V^{\underline{\pi}_{c,1}}(*, \rho) \leq 4/n$.*

Proof. Note that a permutation π sampled from $\underline{\pi}_e$ always maps the set $\left(\frac{n}{4}, \frac{n}{2}\right]$ to itself, though it may permute the elements of that set. Consequently, when one runs ALG_e on an input sequence ordered using π in the support of $\underline{\pi}_e$, it sets $J = \left(\frac{n}{4}, \frac{n}{2}\right]$. The definition of a decodable permutation now implies that the fraction of items in $\{j_1, \ldots, j_{2m}\}$ that belong to $\left(\frac{3n}{8}, \frac{n}{2}\right]$ is at least $\frac{39}{40}$; let us call the remaining items in $\{j_1, \ldots, j_{2m}\}$ "misplaced". For each j_ℓ that is not misplaced, ALG_e correctly deduces the corresponding value y_ℓ *unless* item $j_\ell - \frac{n}{8}$ also belongs to $\{j_1, \ldots, j_{2m}\}$ (in which case it is a misplaced item). Hence each misplaced item contributes to potentially two errors, meaning that at most $\frac{1}{20}$ fraction of the bits in $\hat{y}$ differ from the corresponding bit in $A(x)$. These strings have length $2m$, so we have shown their Hamming distance is at most $\frac{m}{10}$. Lemma 15 now ensures that with probability at least $\frac{1}{2}$, ALG_e decodes the appropriate value of x. When this happens, it correctly selects item n from the second half of the input sequence. Our assumption that σ is decodable means that item n is the item with maximum value, which completes the proof that $V^{\underline{\pi}_e}(\text{ALG}_e, \sigma) \geq \frac{1}{2}$.

To prove the second statement in the lemma, we can use ALG_p to guess the value of $g(x)$ for any input $x \in \{0,1\}^m$ by the following simulation procedure. First, define a permutation $\pi'(x)$ by running Algorithm 3 with random string x, omitting the final step of swapping the items in positions n and $n/2 + g(x)$; note that this means that $\pi'(x)$, unlike $\pi(x)$, can be constructed from input x by an A-augmented circuit of polynomial size. Now simulate ALG on the input sequence $\pi'(x)$, observe the time t when it selects an item, and output $t - \frac{n}{2}$. The A-augmented circuit complexity of this simulation procedure is at most poly(n) times the A-augmented circuit complexity of the stopping rule implemented by ALG, and the fraction of inputs x on which it guesses $V^{\underline{\pi}}(x)$ correctly is precisely $V^{\underline{\pi}}(\text{ALG}, \iota)$. (To verify this last statement, note that ALG makes its selection at time $t = \frac{n}{2} + g(x)$ when observing input sequence $\pi(x)$ *if and only if* if also makes its selection at time t when observing input sequence $\pi'(x)$, because the two input sequences are indistinguishable to comparison-based algorithms at that time.) Hence, if $V^{\underline{\pi}}(\text{ALG}, \iota) > \frac{4}{n}$ then the stopping rule of ALG cannot be implemented by circuits of size $2^{m/2} = 2^{n\,\kappa(n)/4}$. □

B.4 Proofs deferred from §5

B.4.1 Full proof of Theorem 10

We start by proving the following lemmas which turn out to be critical for the analysis of $\mathsf{ALG}(\mathcal{U}, k, q)$ under non-uniform permutation distributions. In fact, these lemmas capture the fact that if membership random variables of different items for the random set S (and S^{c}) are almost pairwise independent (rather than mutually independent), then we still preserve enough of probabilistic properties that are needed in the analysis of algorithm proposed by [25].

Lemma 18. *Suppose π is drawn from a permutation distribution satisfying (p, q, δ)-BIP for $p \geq 2$ and $S \triangleq \{x \in \mathcal{U} : \rho(x) \leq \tau(q)\}$ where $\tau(q)$ is independently drawn from Binom$(q, 1/2)$. Then for any $T \subseteq \mathcal{U}$ such that $\delta \leq \frac{1}{\sqrt{|T|}}$ we have*

1. $\mathbf{E}\left[|T \cap S|\right] \in [(1-\delta)|T|/2, (1+\delta)|T|/2]$
2. $\mathbf{E}\left[val(T \cap S)\right] \in [(1-\delta)val(T)/2, (1+\delta)val(T)/2]$
3. $\mathbf{Pr}\left[|T \cap S| \geq |T|/2 + \alpha\right] \leq \frac{|T|}{2\alpha^2}$

Proof. For $x \in \mathcal{U}$, let Y_x be a 0/1 variable indicating if $x \in S$. We have

$$\begin{aligned}
\mathbf{E}\left[Y_x\right] &= \sum_{i=1} q\mathbf{Pr}\left[x \text{ is in block } i\right]\mathbf{Pr}\left[\tau_b \geq i\right] \\
&\geq (1-\delta)\frac{1}{q}\sum_{i=1} q\mathbf{Pr}\left[\tau_b \geq i\right] \\
&= (1-\delta)\frac{1}{q}\mathbf{E}\left[\tau_b\right] \\
&= (1-\delta)\frac{1}{q}\frac{q}{2} \\
&= \frac{1-\delta}{2}.
\end{aligned}$$

Analogously, we get $\mathbf{E}\left[Y_x\right] \geq \frac{1+\delta}{2}$.

Claims 1 and 2 now follow from linearity of expectation, e.g., $\mathbf{E}\left[|T \cap S|\right] = \mathbf{E}\left[\sum_{x \in T} Y_x\right] \geq \frac{1-\delta}{2}|T|$.

To show Claim 3, we use that for $x \neq x'$, we have $\mathbf{E}\left[Y_x Y_{x'}\right] \leq \frac{1+\delta}{4}$. This implies $\mathbf{E}\left[|T \cap S|^2\right] = \mathbf{E}\left[\sum_x Y_x\right] + \mathbf{E}\left[\sum_{x \neq x'} Y_x Y_{x'}\right] \leq \mathbf{E}\left[|T \cap S|\right] + |T|(|T|-1)\frac{1+\delta}{4} \leq \mathbf{E}\left[|T \cap S|\right]$.

By Markov's inequality, we get

$$\mathbf{Pr}\left[|T \cap S| \geq |T|/2 + \alpha\right] \leq \mathbf{Pr}\left[(|T \cap S| - |T|/2)^2 \geq \alpha^2\right] \leq \frac{1}{\alpha^2}\mathbf{E}\left[(|T \cap S| - |T|/2)^2\right] .$$

Using linearity of expectation and the bounds obtained so far, we get

$$\begin{aligned}\mathbf{E}\left[(|T \cap S| - |T|/2)^2\right] &= \mathbf{E}\left[(|T \cap S|)^2\right] - |T|\mathbf{E}\left[|T \cap S|\right] + \left(\frac{|T|}{2}\right)^2 \\ &\leq \frac{1+\delta}{2}|T|^2 - \frac{1+\delta}{4}|T| - (|T|-1)\mathbf{E}\left[|T \cap S|\right] \\ &\leq \frac{1+\delta}{2}|T|^2 - \frac{1+\delta}{4}|T| - (|T|-1)\frac{1-\delta}{2}|T| \\ &\leq \delta|T|^2 + \frac{1-3\delta}{4}|T| \leq \frac{|T|}{2} .\end{aligned}$$

where the last inequality is true because of $\delta \leq \frac{1}{\sqrt{|T|}}$. □

Lemma 19. *Suppose π is drawn from a permutation distribution satisfying (p, q, δ)-BIP for some $p \geq 2$ and S is as defined in Lemma 18. Let Y_1 be the (possibly negative) random variable such that $(k/2)^{th}$ item in the sorted-by-value list of items in S is the $(k + Y_1)^{th}$ in the sorted-by-value list of items in $\mathcal{U}$. Then $\mathbf{E}\left[|Y_1|\right] = O(\sqrt{k})$.*

Proof. We have $\mathbf{E}\left[|Y_1|\right] = \sum_{i=1}^{\infty} \mathbf{Pr}\left[|Y_1| \geq i\right] = \sum_{i=1}^{\infty} \mathbf{Pr}\left[Y_1 \geq i\right] + \sum_{i=1}^{\infty} \mathbf{Pr}\left[Y_1 \leq -i\right]$. Now we bound each of the terms separately. For a fixed i, look at the event $Y_1 \leq -i$. This event is equivalent to the event that the number of items in S among $k - i$ highest-valued items is at least $k/2$. Let us define $r \triangleq k - i$. Furthermore, let T_r be the set of the r-highest valued items. Using Lemma 18 we have:

$$\mathbf{Pr}\left[Y_1 \leq -i\right] = \mathbf{Pr}\left[|T_r \cap S| \geq r/2 + i/2\right] \leq \frac{2(k-i)}{i^2} \tag{11}$$

So we have

$$\begin{aligned}\sum_{i=1}^{\infty} \mathbf{Pr}\left[Y_1 \leq -i\right] &\leq \sum_{i=1}^{\lceil\sqrt{k}\rceil} 1 + \sum_{i=\lceil\sqrt{k}\rceil+1}^{k} \frac{2(k-i)}{i^2} \leq 1 + \sqrt{k} + 2k \sum_{i=\lceil\sqrt{k}\rceil+1}^{\infty} \frac{1}{i^2} \\ &\leq 1 + \sqrt{k} + 2k \int_{\sqrt{k}}^{\infty} \frac{1}{x^2}dx = 3\sqrt{k} + 1 = O(\sqrt{k}) \end{aligned} \tag{12}$$

Now, let's consider the event $Y_1 \geq i$. This event implies that number of items of S^{c} among the k highest valued items is at least $k/2 + i$. Again, using Lemma 18 we have:

$$\mathbf{Pr}\left[Y_1 \geq i\right] \leq \mathbf{Pr}\left[|T_k \cap S^{\mathsf{c}}| \geq k/2 + i\right] \leq \frac{k}{2i^2} \tag{13}$$

and hence we have

$$\begin{aligned}\sum_{i=1}^{\infty} \mathbf{Pr}\left[Y_1 \geq i\right] &\leq \sum_{i=1}^{\lceil\sqrt{k}\rceil} 1 + \sum_{i=\lceil\sqrt{k}\rceil+1}^{\infty} \frac{k}{2i^2} \leq 1 + \sqrt{k} + \frac{k}{2} \sum_{i=\lceil\sqrt{k}\rceil+1}^{\infty} \frac{1}{i^2} \\ &\leq 1 + \sqrt{k} + \frac{k}{2} \int_{\sqrt{k}}^{\infty} \frac{1}{x^2}dx = \frac{3}{2}\sqrt{k} + 1 = O(\sqrt{k}) \end{aligned} \tag{14}$$

which completes the proof. □

Now we start proving the theorem. Basically, we prove for any fixed k there exists a function $\epsilon(k,q,\delta)$, non-increasing w.r.t. q, such that $\mathsf{ALG}(\mathcal{U},k,q)$ is $\left(1-O(\frac{1}{k^{\frac{1}{3}}})-\epsilon(k,q,\delta)\right)$-competitive, and ϵ goes to 0 as $q\to\infty$ and $\delta\to 0$ for a fixed k. First without loss of generality we modify values so that if the value is among k highest it remains the same, otherwise it is set to 0. This doesn't change sum of the values of the k highest items, and just weakly decreases the values of items picked by any algorithm. Now, run the algorithm with modified values. Let $\mathcal{A}$ be the set of items picked by $\mathsf{ALG}(\mathcal{U},k,q)$ and $\mathcal{O}$ be the subset of k highest value items under σ. Define $S\triangleq\{x\in\mathcal{U}:\pi(x)\leq\frac{\tau(q)}{q}n\}$ to be the set sampled before threshold and $S^{\mathsf{c}}\triangleq\mathcal{U}\backslash S$ be its complement. Suppose v_0 is the value of the $\frac{k}{2}^{\text{th}}$ highest valued item in S (if $|S|<\frac{k}{2}$, set $v_0=0$). Moreover, define the value function val(.) to be the sum of values of the input set of items under σ.

Fixing σ, we prove the claim by induction over k. The case $k=1$ is exactly the case of a single secretary, which is analyzed in §2.1. For general k, we first run $\mathsf{ALG}(\mathcal{U}\cap S,k/2,\tau(q))$ to give us $\mathcal{A}\cap S$. Note that the ordering of arrivals of items in S satisfies $(p,\tau(q),\delta)$-BIP. So, by induction hypothesis and conditioned on set S we have

$$\mathbf{E}\left[\text{val}(\mathcal{A}\cap S)|S\right]\geq\mathbf{E}\left[\text{val}([\mathcal{O}\cap S]_{k/2})\left(1-O(\frac{1}{k^{\frac{1}{3}}})-\epsilon(k/2,\tau,\delta)\right)|S\right]\tag{15}$$

We can lower-bound the right-hand side further as follows.

$$\begin{aligned}&\mathbf{E}\left[\text{val}([\mathcal{O}\cap S]_{k/2})\left(1-O(\frac{1}{k^{\frac{1}{3}}})-\epsilon(k/2,\tau,\delta)\right)|S\right]\geq\\&\mathbf{E}\left[\text{val}([\mathcal{O}\cap S]_{k/2})|S\right]-\text{val}(\mathcal{O})\left(O(\frac{1}{k^{\frac{1}{3}}})+\mathbf{E}\left[\epsilon(k/2,\tau,\delta)|S\right]\right),\end{aligned}\tag{16}$$

and by taking expectation with respect to S we have

$$\mathbf{E}\left[\text{val}(\mathcal{A}\cap S)\right]\geq\mathbf{E}\left[\text{val}([\mathcal{O}\cap S]_{k/2})\right]-\text{val}(\mathcal{O})\left(O(\frac{1}{k^{\frac{1}{3}}})+\mathbf{E}\left[\epsilon(k/2,\tau,\delta)\right]\right)\tag{17}$$

Now suppose $\mathbb{I}\{.\}$ is the indicator function. One can easily decompose $\epsilon(k/2,\tau,\delta)$ as follows.

$$\epsilon(k/2,\tau,\delta)=\epsilon(k/2,\tau,\delta)\mathbb{I}\{\tau\geq q/4\}+\epsilon(k/2,\tau,\delta)\mathbb{I}\{\tau<q/4\}\leq\epsilon(k/2,q/4,\delta)+\mathbb{I}\{\tau<q/4\}\tag{18}$$

where the last inequality is true because $\epsilon(k/2,\tau,\delta)$ is non-increasing w.r.t. τ. Now by taking expectations from both hand sides of (18), we have:

$$\begin{aligned}\mathbf{E}\left[\epsilon(k/2,\tau,\delta)\right]&\leq\epsilon(k/2,q/4,\delta)+\mathbf{Pr}\left[\tau<q/4\right]\\&=\epsilon(k/2,q/4,\delta)+\mathbf{Pr}\left[\tau<q/2(1-1/2)\right]\leq\epsilon(k/2,q/4,\delta)+e^{-\frac{q}{16}}\end{aligned}\tag{19}$$

where in the last inequality we used Chernoff bound, as τ is drawn from $\text{Binom}(q,1/2)$. Now, fix ε'. For a given ε' we have

$$\begin{aligned}&\mathbf{E}\left[\text{val}([\mathcal{O}\cap S]_{k/2})\right]=\mathbf{E}\left[\text{val}(\mathcal{O}\cap S)\mathbb{I}\{|\mathcal{O}\cap S|<k/2\}\right]+\mathbf{E}\left[\text{val}([\mathcal{O}\cap S]_{k/2})\mathbb{I}\{|\mathcal{O}\cap S|\geq k/2\}\right]\\&\geq\mathbf{E}\left[\text{val}(\mathcal{O}\cap S)\mathbb{I}\{|\mathcal{O}\cap S|<k/2\}\right]+\mathbf{E}\left[\frac{k/2}{|\mathcal{O}\cap S|}\text{val}(\mathcal{O}\cap S)\mathbb{I}\{|\mathcal{O}\cap S|\geq k/2\}\right]\\&\geq\mathbf{E}\left[\text{val}(\mathcal{O}\cap S)\mathbb{I}\{|\mathcal{O}\cap S|<k/2\}\right]+\frac{1}{1+\varepsilon'}\mathbf{E}\left[\text{val}(\mathcal{O}\cap S)\mathbb{I}\left\{\frac{k}{2}(1+\varepsilon')\geq|\mathcal{O}\cap S|\geq k/2\right\}\right]\end{aligned}\tag{20}$$

Also, we have:

$$
\begin{aligned}
&\frac{1}{1+\varepsilon'}\mathbf{E}\left[\mathrm{val}(\mathcal{O}\cap S)\mathbb{I}\left\{\frac{k}{2}(1+\varepsilon') \geq |\mathcal{O}\cap S| \geq k/2\right\}\right] \\
&= \frac{1}{1+\varepsilon'}\mathbf{E}\left[\mathrm{val}(\mathcal{O}\cap S)\mathbb{I}\left\{|\mathcal{O}\cap S| \geq k/2\right\}\right] - \frac{1}{1+\varepsilon'}\mathbf{E}\left[\mathrm{val}(\mathcal{O}\cap S)\mathbb{I}\left\{|\mathcal{O}\cap S| \geq \frac{k}{2}(1+\varepsilon')\right\}\right] \\
&\geq \frac{1}{1+\varepsilon'}\mathbf{E}\left[\mathrm{val}(\mathcal{O}\cap S)\mathbb{I}\left\{|\mathcal{O}\cap S| \geq k/2\right\}\right] - \mathrm{val}(\mathcal{O})\mathbf{Pr}\left[|\mathcal{O}\cap S| \geq \frac{k}{2}(1+\varepsilon')\right] \\
&\overset{(1)}{\geq} \frac{1}{1+\varepsilon'}\mathbf{E}\left[\mathrm{val}(\mathcal{O}\cap S)\mathbb{I}\left\{|\mathcal{O}\cap S| \geq k/2\right\}\right] - \frac{1}{k\varepsilon'^2}\mathrm{val}(\mathcal{O}) \qquad (21)
\end{aligned}
$$

in which (1) is true because of Lemma 18. Combining (20) with (21) we have:

$$
\mathbf{E}\left[\mathrm{val}([\mathcal{O}\cap S]_{k/2})\right] \geq \frac{1}{1+\varepsilon'}\mathbf{E}\left[\mathrm{val}(\mathcal{O}\cap S)\right] - \frac{1}{k\varepsilon'^2}\mathrm{val}(\mathcal{O}) \geq \mathbf{E}\left[\mathrm{val}(\mathcal{O}\cap S)\right] - (\varepsilon' + \frac{1}{k\varepsilon'^2})\mathrm{val}(\mathcal{O}). \quad (22)
$$

Finally, by combining (17), (19) and (22) and setting $\varepsilon' = \frac{1}{k^{\frac{1}{3}}}$, we have

$$
\mathbf{E}\left[\mathrm{val}(\mathcal{A}\cap S)\right] \geq \mathbf{E}\left[\mathrm{val}(\mathcal{O}\cap S)\right] - \mathrm{val}(\mathcal{O})\left(O(\frac{1}{k^{\frac{1}{3}}}) + e^{-\frac{q}{16}} + \epsilon(k/2, q/4, \delta)\right) \qquad (23)
$$

Next, we try to lower-bound $\mathbf{E}\left[\mathrm{val}(\mathcal{A}\cap S^{\mathsf{c}})\right]$ by $\mathbf{E}\left[\mathrm{val}(\mathcal{O}\cap S^{\mathsf{c}})\right]$. Lets define random variable $Q \triangleq |\mathcal{A}\cap S^{\mathsf{c}}|$ to be number of items algorithm picked from S^{c}. We have $\mathbf{E}\left[\mathrm{val}(\mathcal{A}\cap S^{\mathsf{c}})\right] = \sum_{x=0}^{k/2}\mathbf{E}\left[\mathrm{val}(\mathcal{A}\cap S^{\mathsf{c}})\mathbb{I}\left\{Q=x\right\}\right]$. Now we look at each term $\mathbf{E}\left[\mathrm{val}(\mathcal{A}\cap S^{\mathsf{c}})\mathbb{I}\left\{Q=x\right\}\right]$, and we try to lower-bound it with $\mathbf{E}\left[\mathrm{val}(\mathcal{O}\cap S^{\mathsf{c}})\mathbb{I}\left\{Q=x\right\}\right]$ for different values of x. Consider two cases:

Case 1, when $x < \frac{k}{2}$: In this case $v_0 > 0$ and all items in S with value more than v_0 are in $\mathcal{O}$. We know the number of items in S^{c} that have value at least v_0 is x. If we look at items in $\mathcal{O}\cap S^{\mathsf{c}}$, all items in $\mathcal{A}\cap S^{\mathsf{c}}$ are also in $\mathcal{O}\cap S^{\mathsf{c}}$ and in addition we have at most $k-(k/2+x) = k/2-x$ items in $\mathcal{O}\cap S^{\mathsf{c}}$, all of which have value at most v_0. Hence, as the value of any item in $\mathcal{A}\cap S^{\mathsf{c}}$ is at least v_0, the followings hold deterministically :

$$
\begin{aligned}
&\mathrm{val}(\mathcal{O}\cap S^{\mathsf{c}}) - \mathrm{val}(\mathcal{A}\cap S^{\mathsf{c}}) = \mathrm{val}(\{x \in \mathcal{O}\cap S^{\mathsf{c}} : v_{\sigma^{-1}(x)} \leq v_0\}) \\
&\leq \frac{|\mathcal{O}\cap S^{\mathsf{c}}| - |\mathcal{A}\cap S^{\mathsf{c}}|}{|\mathcal{A}\cap S^{\mathsf{c}}|}\mathrm{val}(\mathcal{A}\cap S^{\mathsf{c}}) \leq (\frac{k}{2x} - 1)\mathrm{val}(\mathcal{A}\cap S^{\mathsf{c}}) \qquad (24)
\end{aligned}
$$

which implies $\mathrm{val}(\mathcal{A}\cap S^{\mathsf{c}}) \geq \frac{2x}{k}\mathrm{val}(\mathcal{O}\cap S^{\mathsf{c}})$ when $x < k/2$. So for $x < k/2$,

$$
\mathbf{E}\left[\mathrm{val}(\mathcal{A}\cap S^{\mathsf{c}})\mathbb{I}\left\{Q=x\right\}\right] \geq \mathbf{E}\left[\frac{2Q}{k}\mathrm{val}(\mathcal{O}\cap S)\mathbb{I}\left\{Q=x\right\}\right] \qquad (25)
$$

Case 2, when $x = \frac{k}{2}$: In this case either $v_0 > 0$, which implies at least there are $k/2$ items in $\mathcal{O}\cap S$. As algorithm also picks $k/2$ items and so $\mathcal{A}\cap S^{\mathsf{c}} = \mathcal{O}\cap S^{\mathsf{c}}$ for which we are done. Otherwise, suppose $v_0 = 0$. We know the permutation distribution generating π satisfies the (p, q, δ)-BIP some $p \geq k$, and hence it satisfies (k, q, δ)-BIP. So, based on Theorem 13 it also satisfies $(k, \delta + \frac{k^2}{q})$-UIOP. Roughly speaking, if you look at any subset of elements with cardinality at most k, their induced ordering is almost uniformly distributed (within an error of $\delta + \frac{k^2}{q}$). We know in this case algorithm picks $\frac{k}{2}$ items (all of them in $\mathcal{O} \cap S^{\mathsf{c}}$), and in fact it picks the first $\frac{k}{2}$ elements of $\mathcal{O} \cap S^{\mathsf{c}}$ in the ordering of elements in $\mathcal{O} \cap S^{\mathsf{c}}$ induced by the permutation π. Suppose $X \triangleq |\mathcal{O} \cap S^{\mathsf{c}}| - k/2$. Then

$$\begin{aligned}
\mathbf{E}\left[\text{val}(\mathcal{A} \cap S^{\mathsf{c}})\mathbb{I}\left\{Q = \frac{k}{2}\right\}\right] &\geq \mathbf{E}\left[\text{val}(\text{first } k/2 \text{ elements of } \mathcal{O} \cap S^{\mathsf{c}} \text{ in the ordering } \pi)\mathbb{I}\left\{Q = \frac{k}{2}\right\}\right] \\
&= \mathbf{E}\left[\text{val}(\mathcal{O} \cap S^{\mathsf{c}})\mathbb{I}\left\{Q = \frac{k}{2}\right\}\right] \\
&\quad - \mathbf{E}\left[\text{val}(\text{last } X \text{ elements of } \mathcal{O} \cap S^{\mathsf{c}} \text{ in the ordering } \pi)\mathbb{I}\left\{Q = \frac{k}{2}\right\}\right] \\
&\geq \mathbf{E}\left[\text{val}(\mathcal{O} \cap S^{\mathsf{c}})\mathbb{I}\left\{Q = \frac{k}{2}\right\}\right] \\
&\quad - \mathbf{E}\left[\text{val}(\text{last } X \text{ elements of } \mathcal{O} \cap S^{\mathsf{c}} \text{ in the ordering } \pi)\right] \qquad (26)
\end{aligned}$$

For a fixed set S, we have $(k, \delta + \frac{k^2}{q})$-UIOP for elements in $\mathcal{O} \cap S^{\mathsf{c}}$ (this is an order oblivious fact), and hence the induced ordering of the elements in $\mathcal{O} \cap S^{\mathsf{c}}$ is almost uniform. So, we have

$$\begin{aligned}
&\mathbf{E}\left[\text{val}(\text{last } X \text{ elements of } \mathcal{O} \cap S^{\mathsf{c}} \text{ in the ordering } \pi)|S\right] \leq (1 + \delta + \frac{k^2}{q})\mathbf{E}\left[\text{val}(\mathcal{O} \cap S^{\mathsf{c}})\frac{X}{|\mathcal{O} \cap S^{\mathsf{c}}|}|S\right] \\
&\leq (1 + \delta + \frac{k^2}{q})\text{val}(\mathcal{O})\mathbf{E}\left[\frac{|\mathcal{O} \cap S^{\mathsf{c}}| - k/2}{|\mathcal{O} \cap S^{\mathsf{c}}|}|S\right] \qquad (27)
\end{aligned}$$

Now by taking expectation w.r.t. S and combining it with (26) we have

$$\mathbf{E}\left[\text{val}(\mathcal{A} \cap S^{\mathsf{c}})\mathbb{I}\left\{Q = \frac{k}{2}\right\}\right] \geq \mathbf{E}\left[\text{val}(\mathcal{O} \cap S^{\mathsf{c}})\mathbb{I}\left\{Q = \frac{k}{2}\right\}\right] - (1 + \delta + \frac{k^2}{q})\text{val}(\mathcal{O})\mathbf{E}\left[\frac{|\mathcal{O} \cap S^{\mathsf{c}}| - k/2}{|\mathcal{O} \cap S^{\mathsf{c}}|}\right] \qquad (28)$$

Moreover, one can use Lemma 18 to find an upper-bound on the error term in (27). Fix any ε', Now we have

$$\frac{|\mathcal{O} \cap S^{\mathsf{c}}| - k/2}{|\mathcal{O} \cap S^{\mathsf{c}}|} = \frac{|\mathcal{O} \cap S^{\mathsf{c}}| - k/2}{|\mathcal{O} \cap S^{\mathsf{c}}|}\mathbb{I}\left\{|\mathcal{O} \cap S^{\mathsf{c}}| < k/2 + \varepsilon'\right\} + \frac{|\mathcal{O} \cap S^{\mathsf{c}}| - k/2}{|\mathcal{O} \cap S^{\mathsf{c}}|}\mathbb{I}\left\{|\mathcal{O} \cap S^{\mathsf{c}}| \geq k/2 + \varepsilon'\right\} \qquad (29)$$

By taking expectation from both sides of (29), setting $\varepsilon' = k^{\frac{1}{3}}$ and using Lemma 18 we have

$$\mathbf{E}\left[\frac{|\mathcal{O} \cap S^{\mathsf{c}}| - k/2}{|\mathcal{O} \cap S^{\mathsf{c}}|}\right] \leq \frac{\varepsilon'}{k/2 + \varepsilon'} + \mathbf{Pr}\left[|\mathcal{O} \cap S^{\mathsf{c}}| \geq k/2 + \varepsilon'\right] \leq \frac{\varepsilon'}{k/2} + \frac{k}{2\varepsilon'} \leq \frac{3}{k^{\frac{1}{3}}}. \qquad (30)$$

By combining (30) and (27) we have (note that $\delta \leq 1$)

$$\mathbf{E}\left[\text{val}(\mathcal{A} \cap S^{\mathsf{c}})\mathbb{I}\left\{Q = k/2\right\}\right] \geq \mathbf{E}\left[\frac{2Q}{k}\text{val}(\mathcal{O} \cap S^{\mathsf{c}})\mathbb{I}\left\{Q = k/2\right\}\right] - \text{val}(\mathcal{O})(\frac{6}{k^{\frac{1}{3}}} + \frac{3k^{\frac{5}{3}}}{q}) \qquad (31)$$

As we desired.

Now, by combining the above cases with each other we have

$$\begin{aligned}
\mathbf{E}\left[\text{val}(\mathcal{A} \cap S^{\mathsf{c}})\right] &\geq \mathbf{E}\left[\frac{2Q}{k}\text{val}(\mathcal{O} \cap S^{\mathsf{c}})\right] - \text{val}(\mathcal{O})(\frac{6}{k^{\frac{1}{3}}} + \frac{3k^{\frac{5}{3}}}{q}) \\
&\geq \mathbf{E}\left[\text{val}(\mathcal{O} \cap S^{\mathsf{c}})\right] - \text{val}(\mathcal{O})\left(\frac{6}{k^{\frac{1}{3}}} + \frac{3k^{\frac{5}{3}}}{q} + \mathbf{E}\left[\frac{k/2 - Q}{k/2}\right]\right) \qquad (32)
\end{aligned}$$

Finally, by combining equations (23) and (32) we have

$$\mathbf{E}\left[\mathrm{val}(\mathcal{A})\right] \geq \mathrm{val}(\mathcal{O}) \left(1 - \mathbf{E}\left[\frac{k/2 - Q}{k/2}\right] - O(\frac{1}{k^{\frac{1}{3}}}) - \frac{3k^{\frac{5}{3}}}{q} - e^{\frac{-q}{16}} + \epsilon(k/2, q/4, \delta)\right) \tag{33}$$

As it can be seen, the question of finding the competitive ratio of $1 - o(1)$ boils down to upper-bounding $\mathbf{E}\left[k/2 - Q\right]$. To do so, we define random variable Y_1 such that $(k/2)^{\text{th}}$ item in sorted-by-value list of items in S will be the $(k + Y_1)^{\text{th}}$ item in $\mathcal{U}$. Now we claim that $Q \geq k/2 - |Y_1|$. The proof is easy. If $v_0 = 0$ then algorithm picks $k/2$ items from S^{c} and we are done. Otherwise, there are $k + Y_1 - k/2 = k/2 + Y_1$ items in S^{c} such that their values is at least v_0. By a simple case analysis, if $Y_1 \leq 0$ then algorithm picks all of those, and hence $Q \geq k/2 + Y_1 = k/2 - |Y_1|$. If $Y_1 \geq 0$ then algorithm picks $k/2$ items which is $\geq k/2 - |Y_1|$ we are again down. So $\mathbf{E}\left[k/2 - Q\right] \leq \mathbf{E}\left[|Y_1|\right]$. Lemma 19 shows that $\mathbf{E}\left[|Y_1|\right] = O(\sqrt{k})$, and hence $\mathbf{E}\left[\frac{k/2-Q}{k/2}\right] \leq O(1/\sqrt{k})$. Hence, we have

$$\begin{aligned}
\mathbf{E}\left[\mathrm{val}(\mathcal{A})\right] &\geq \mathrm{val}(\mathcal{O}) \left(1 - O(\frac{1}{k^{\frac{1}{3}}}) - \frac{3k^{\frac{5}{3}}}{q} - e^{\frac{-q}{16}} - \epsilon(k/2, q/4, \delta)\right) \\
&\geq \mathrm{val}(\mathcal{O}) \left(1 - O(\frac{1}{k^{\frac{1}{3}}}) - \epsilon(k, q, \delta)\right)
\end{aligned} \tag{34}$$

which completes the proof, as ϵ can be arbitrary small for large enough q and small enough δ .

B.4.2 Full proof of Theorem 11

We define a randomized construction that defines an input and a probability distribution simultaneously. By the probabilistic method this implies the statement.

Our bipartite graph has n vertices on the online and the offline side each. For each pair (j, i), we add the connecting edge with probability $\frac{1}{2} - 8\sqrt{\frac{\ln n}{n}}$ independently. In case j and i are connected, the edge weight is set to $w(j, i) = 1 - \epsilon(j + i)$ for $\epsilon = \frac{1}{n^2}$. This way, the expected weight of the optimal solution is $\Omega(n)$.

To define the distribution over permutations $\pi_i \colon \mathcal{U} \to [n]$, we draw for the first $\xi = \frac{2(k+1)!}{\delta^2} \log n$ offline vertices $i \in R$ one permutation uniformly at random from the set of all permutations in which the neighbors of node i come last. Afterwards, we draw one of these permutations $\pi_1, \ldots, \pi_\xi$ at random. We claim that this way, the probability distribution fulfills the (k, δ)-uniform-induced-ordering property.

Fix k distinct items $x_1, \ldots, x_k \in \mathcal{U}$. Note that we can ignore the fact that in any permutation neighbors come last as all $x_1, \ldots, x_k$ have the same probability of corresponding to a neighbor. Therefore, we can steadily follow the argument from Theorem 5. Let $Y_i = 1$ if $\pi_i(x_1) < \pi_i(x_2) < \ldots < \pi_i(x_k)$. As π_i is drawn uniformly from the set of all permutations, we have $\mathbf{Pr}\left[Y_i = 1\right] = \frac{1}{k!}$. That is, we have $\mathbf{E}\left[\sum_{i=1}^{\xi} Y_i\right] = \frac{\xi}{k!}$. As the random variables Y_i are independent, we can apply a Chernoff bound. This gives us

$$\mathbf{Pr}\left[\sum_{i=1}^{\xi} Y_i \leq (1 - \delta)\frac{\xi}{k!}\right] \leq \exp\left(-\frac{\delta^2}{2}\frac{\xi}{k!}\right) = n^{k+1} \quad .$$

Note that if $\sum_{i=1}^{\xi} Y_i \leq (1 - \delta)\frac{\xi}{k!}$ then the respective sequence $x_1, \ldots, x_k \in \mathcal{U}$ has probability at least $(1 - \delta)\frac{1}{k!}$ when drawing one permutation from $\pi_1, \ldots, \pi_\xi$.

There are fewer than n^k possible sequences. Therefore, applying a union bound, with probability at least $1-\frac{1}{n}$ the bound is fulfilled for all sequences simultaneously and therefore S fulfills the stated condition.

It now remains to show that the Korula-Pál algorithm has a poor performance on this type of instance. The algorithm draws a transition point $\tau \sim \mathrm{Binom}(n, \frac{1}{2})$, before which it only observes the input and after which it starts a greedy allocation based on the vertices seen until round τ and the current vertex. It is important to remark that for the tentative allocation the other vertices seen between round τ and the current round are ignored. Only after a tentative edge has been selected, their allocation is taken into consideration in order to check whether the matching would still be feasible.

Let π_i be the chosen permutation. That is, the neighbors of i come last. Let i have $n - A$ neighbors. We now claim that with high probability no neighbor of i comes before τ, i.e., $A < \tau$. Furthermore, after τ essentially only neighbors of i arrive. This has the consequence that almost all vertices are tentatively matched to i. However, only the first such edge is feasible.

Using Chernoff bounds, we get

$$\mathbf{Pr}\left[\tau < \frac{n}{2} - 2\sqrt{n \ln n}\right] = \mathbf{Pr}\left[n - \tau > \frac{n}{2}\left(1 + 4\sqrt{\frac{\ln n}{n}}\right)\right] \leq \exp\left(-\frac{1}{3}16\frac{\ln n}{n}\frac{n}{2}\right) < \frac{1}{n} \ .$$

$$\mathbf{Pr}\left[A > \frac{n}{2} - 2\sqrt{n \ln n}\right] \leq \mathbf{Pr}\left[R > \left(1 + 4\sqrt{\frac{\ln n}{n}}\right)\left(\frac{n}{2} - 8\sqrt{n \ln n}\right)\right] < \frac{1}{n} \ .$$

In case $A < \tau$, the value of the solution is upper-bounded by j because every node is tentatively matched to a vertex of index at most i. As $i \leq \tau$, this gives a value bounded by $\xi = \frac{2(k+1)!}{\delta^2} \ln n$.