Sample Efficient Estimation and Recovery in Sparse FFT via Isolation on Average

Michael Kapralov[*]

August 17, 2017

Abstract

The problem of computing the Fourier Transform of a signal whose spectrum is dominated by a small number k of frequencies quickly and using a small number of samples of the signal in time domain (the Sparse FFT problem) has received significant attention recently. It is known how to approximately compute the k-sparse Fourier transform in $\approx k\log^2 n$ time [Hassanieh et al'STOC'12], or using the optimal number $O(k\log n)$ of samples [Indyk et al'FOCS'14] in time domain, or come within $(\log\log n)^{O(1)}$ factors of both these bounds simultaneously, but no algorithm achieving the optimal $O(k\log n)$ bound in sublinear time is known.

At a high level, sublinear time Sparse FFT algorithms operate by 'hashing' the spectrum of the input signal into $\approx k$ 'buckets', identifying frequencies that are 'isolated' in their buckets, subtracting them from the signal and repeating until the entire signal is recovered. The notion of 'isolation' in a 'bucket', inspired by applications of hashing in sparse recovery with arbitrary linear measurements, has been the main tool in the analysis of Fourier hashing schemes in the literature. However, Fourier hashing schemes, which are implemented via filtering, tend to be 'noisy' in the sense that a frequency that hashes into a bucket contributes a non-negligible amount to neighboring buckets. This leakage to neighboring buckets makes identification and estimation challenging, and the standard analysis based on isolation becomes difficult to use without losing $\omega(1)$ factors in sample complexity.

In this paper we propose a new technique for analysing noisy hashing schemes that arise in Sparse FFT, which we refer to as *isolation on average*. We apply this technique to two problems in Sparse FFT: estimating the values of a list of frequencies using few samples and computing Sparse FFT itself, achieving sample-optimal results in $k\log^{O(1)} n$ time for both. We feel that our approach will likely be of interest in designing Fourier sampling schemes for more general settings (e.g. model based Sparse FFT).

[*] School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland. Email: michael.kapralov@epfl.ch

1 Introduction

The Discrete Fourier Transform (DFT) is a fundamental computational primitive with numerous applications in areas such as digital signal processing, medical imaging and data analysis as a whole. The fastest known algorithm for computing the Discrete Fourier Transform of a signal of length n is the FFT algorithm, designed by Cooley and Tukey in 1965. The efficiency of FFT, which runs in time $O(n \log n)$ on any signal of length n, has contributed significantly to its popularity as a computational primitive, making FFT one of the top 10 most important algorithms of the 20th century [Cip00]. However, computational efficiency of FFT is not the only reason why the Fourier transform emerges in many applications: in signal processing the Fourier basis is often a convenient way of representing signals since it concentrates their energy on a few components, allowing compression (which is the rationale behind image and video compression schemes such as JPEG and MPEG), and in medical imaging applications such as MRI the Fourier transform captures the physics of the measurement process (the problem of reconstructing an image from MRI data is exactly the problem of reconstructing a signal x from Fourier measurements of x). While FFT works for worst case signals, signals arising in practice often exhibit structure that can be exploited to speed up the computation of the Fourier transform. For example, it is often the case that most of the energy of these signals is concentrated on a small number of components in Fourier domain. In other words, the signals that arise in applications are often *sparse* (have a small number of nonzeros) or *approximately sparse* (can be well approximated by a small number of dominant coefficients) in the Fourier domain. This motivates the question of (approximately) computing the Fourier transform of a signal that is (approximately) sparse in Fourier domain using *few samples of the signal in time domain (i.e. with small sample complexity)* and *small runtime*. We note that while runtime is a natural parameter to optimize, sample complexity is at least as important in applications such as medical imaging, where sample complexity governs the *measurement* complexity of the imaging process.

In this paper we consider the problem of computing a sparse approximation to a signal $x \in \mathbb{C}^n$ given access to its Fourier transform $\widehat{x} \in \mathbb{C}^n$, which is equivalent to the problem above (since the inverse Fourier transform only differs from the Fourier transform by a conjugation), but leads to somewhat more compact notation. This problem has been studied extensively. The seminal work of [CT06, RV08] in *compressed sensing* first showed that length n signals with at most k Fourier coefficients can be recovered using only $k \log^{O(1)} n$ samples in time domain. The recovery algorithms are based on linear programming and run in time polynomial in n. A different line of research on the *Sparse Fourier Transform* (Sparse FFT), originating from computational complexity and learning theory, has resulted in algorithms that use $k \log^{O(1)} n$ samples and $k \log^{O(1)} n$ runtime (i.e. the runtime is *sublinear* in the length of the input signal). Many such algorithms have been proposed in the literature, including [GL89, KM91, Man92, GGI$^+$02, AGS03, GMS05, Iwe10, Aka10, HIKP12b, HIKP12a, LWC12, BCG$^+$12, HAKI12, PR13, HKPV13, IKP14, IK14, Kap16, PS15, CKPS16]. Nevertheless, despite significant progress that has recently been achieved, important gaps in our understanding of sample and time efficient recovery from Fourier measurements remain. We address some of these gaps in this work.

The main contribution of this work is a new technique for designing and analyzing sample efficient sublinear time Sparse FFT algorithms. We refer to this technique as *isolation on average*. We apply our technique to two problems in the area of Sparse Fourier Transform computation, namely estimation and recovery with Fourier measurements.

Our results: estimation In the first problem we are given a subset $S \subseteq [n]$ of locations in the time domain and are asked to estimate x_S from a few values of $\widehat{x}$. Formally, we would like the algorithm to output a signal x' with $\text{supp}(x') \subseteq S$ such that

$$\|x - x'\|_2^2 \leq (1+\epsilon)\|x_{[n]\setminus S}\|_2^2 \tag{1}$$

In other words, we would like to output an estimate x' of x that is correct up to the 'noise', i.e. elements outside of S (to achieve (1), it suffices to ensure that $\|(x - x')_S\|_2^2 \leq \epsilon \|x_{[n]\setminus S}\|_2^2$). Note that in some of the applications described above one often has a good prior on which coefficients of x are the dominant ones, and a natural question is whether one can recover the values of x_S quickly, using few samples, and in a noise robust manner, i.e. solve (1). Our main result on estimation is Algorithm 2 (presented in Section 4) together with

Theorem 1.1. *For every $\epsilon \in (1/n, 1), \delta \in (0, 1/2)$, $x \in \mathbb{C}^n$ and every integer $k \geq 1$, any $S \subseteq [n]$, $|S| = k$, if $||x||_\infty \leq R^* \cdot ||x_{[n]\setminus S}||_2/\sqrt{k}, R^* = n^{O(1)}$, an invocation of* $\textsc{Estimate}(\hat{x}, S, k, \epsilon, R^*)$ *(Algorithm 2) returns $\chi^* \in \mathbb{C}^n$ such that*

$$||(x - \chi^*)_S||_2^2 \leq \epsilon \cdot ||x_{[n]\setminus S}||_2^2$$

using $O_\delta(\frac{1}{\epsilon}k)$ samples and $O_\delta(\frac{1}{\epsilon}k \log^{3+\delta} n)$ time with at least $4/5$ success probability.

A linear sketch with $O(k)$ measurements and $O(k)$ recovery time that provides the guarantee in (1) was presented in [Pri11], but this solution uses general linear measurements as opposed to the more restrictive Fourier measurements. To the best of our knowledge, the estimation problem with guarantees (1) has not been studied explicitly in the setting of Fourier measurements. We now describe 'folklore' results and give a comparison with Theorem 1.1.

Estimation from Fourier measurements: least squares Recall that the problem is as follows: given a set $S \subseteq [n]$, estimate x_S from a small number of Fourier measurements of x, i.e. from a small number of accesses to $\widehat{x}$. A popular approach is to select a subset $T \subseteq [n]$ of frequencies and solve the least squares problem

$$\min_{y \in \mathbb{C}^n, \operatorname{supp} y \subseteq S} ||\widehat{y}_T - \widehat{x}_T||_2^2. \tag{2}$$

A natural choice is to let T be a (multi)set of frequencies selected uniformly at random with replacement from $[n]$. The solution to (2) is then provided by the normal equations $y_{OPT} = (F^*_{T,S} F_{T,S})^{-1} F^*_{T,S} x$, where $Fx \in \mathbb{C}^n$ is the Fourier transform of x, and $F_{T,S}$ is the $T \times S$ submatrix of F scaled by $\sqrt{n/|T|}$. Writing $x = x_S + x_{[n]\setminus S}$, so that $\widehat{x}_T = F_{T,S} x_S + F_{T,[n]\setminus S} x_{[n]\setminus S}$, we get $y_{OPT} = (F^*_{T,S} F_{T,S})^{-1} F^*_{T,S} \widehat{x}_T = x_S + (F^*_{T,S} F_{T,S})^{-1} F^*_{T,S} F_{T,[n]\setminus S} x_{[n]\setminus S}$, where the second term corresponds to the estimation error due to tail noise. Thus, if T is such that $\frac{1}{2} I_S \preceq F^*_{T,S} F_{T,S} \preceq 2 I_S$, then $||y_{OPT} - x||_2 = O(1) \cdot ||x_{[n]\setminus S}||_2$ with constant probability. A simple application of matrix Chernoff bounds shows that the spectral bound $\frac{1}{2} I_S \preceq F^*_{T,S} F_{T,S} \preceq 2 I_S$ is satisfied when $|T| \geq C|S| \log |S|$ for an absolute constant C. Note that the analysis above is tight, as for certain choices of $S \subseteq [n]$ at least $\Omega(|S| \log |S|)$ samples are needed even to ensure that $F^*_{S,T} F_{S,T}$ is invertible. For example, suppose that $S = (n/k) \cdot [k]$, where k divides n, so that the signal $\widehat{x}$ is k-periodic. In this case x only becomes recoverable from $\widehat{x}_T$ as long as T contains at least one element of every conjugacy class of $\mathbb{Z}_n$ modulo k, and by a Coupon Collection argument $\Omega(k \log k)$ samples are needed to ensure that this is the case. To summarize, the sample complexity of least squares with a random T is at least $\Omega(k \log k)$. Another significant disadvantage of this approach is that solving the least squares problem requires at least $\Omega(k^2)$ runtime using current techniques. Of course, given the knowledge of S one may be able to design a better than random set T, but no such construction is known for general supports S. As Theorem 1.1 shows, there exists a distribution over sampling patterns T that is *oblivious to* S and allows decoding from $O(k)$ samples in $k \log^{O(1)} n$ time.

Estimation from Fourier measurements: Fourier hashing Estimation of a subset S of coefficients of x using Fourier measurements can be performed using the idea of Fourier hashing (via filtering) commonly used in the Sparse FFT literature. In this approach one round of hashing allows one to compute estimates w_i for $x_i, i \in S$ such that

$$|w_i - x_i| \leq \alpha ||x||_2^2/k$$

using $O(k/\alpha)$ samples in Fourier domain and $O((k/\alpha) \log(k/\alpha))$ runtime. Here $\alpha \in (0, 1)$ is the oversampling parameter, which is normally set to a small constant, as it directly affects runtime and sample complexity. The approach is similar to standard hashing techniques such as CountSketch [CCFC02], but the crucial difference is that the error bound depends on the *energy of the entire signal* as opposed to energy of the tail[1]. Indeed,

[1] One way to improve the error bound is to use strong filters [HIKP12b], but that requires a $\Omega(k \log n)$ samples – see Lemma E.1

in general one can have $||x||_2^2 \gg n^{\Omega(1)} \cdot ||x_{[n]\setminus S}||_2^2$, meaning that one round of hashing gives results that are very far from estimating x_S up to the energy of the 'noise', i.e. elements outside S. This can be fixed by iterating the estimation process on the residual signal. A naive implementation and analysis results in $\Theta(k \log n)$ measurements (due to $\log n$ iterations of refinement) and $k \log^{O(1)} n$ time. Recent works on saving samples by reusing measurements [IK14, Kap16] can lead to improvements over the factor $\log n$ blow up in sample complexity, but all prior approaches inherently lead to $\omega(1)$ factor loss in the number of samples, as we argue below.

Our results: recovery The second version of the problem is the Sparse FFT (recovery) problem with ℓ_2/ℓ_2 guarantees: we are given access to $\widehat{x}$, a precision parameter $\epsilon > 0$ and a sparsity parameter k, and would like to output x' such that

$$\|x - x'\|_2 \leq (1+\epsilon) \min_{k\text{-sparse } y} \|x - y\|_2, \tag{3}$$

Note that here we are not provided with any information about the 'heavy' coefficients of x, and the hardest and most sample intensive part of the problem is to recover the identities of the 'heavy' elements.

It is known that any (randomized, non-adaptive) algorithm whose output satisfies (3) with at least constant probability must use $m = \Omega(k \log(n/k))$ samples [DIPW10]. An algorithm that matches this bound for every $k \leq n^{1-\delta}$ was recently proposed by [IK14]. The algorithm of [IK14] required $\Omega(n)$ runtime, however, leaving open the problem of achieving sample-optimality in $k \log^{O(1)} n$, or even just *sublinear time*. Sublinear time algorithms that come close to the optimal sample complexity (within an $O(\log \log n)$ factor) have been proposed [IKP14, Kap16], but no algorithm was able to match the lower bound to within constant factors using sublinear runtime[2]. As we argue below, achieving the $O(k \log n)$ bound in sublinear time appears to require a fundamentally different approach to Fourier hashing, which we provide in this work. Our new technique results in an algorithm that matches the lower bound of [DIPW10] up to constant factors for every k polynomially bounded away from n (i.e. $k \leq n^{1-c}$ for a constant $c > 0$) in sublinear time:

Theorem 1.2. *For any $\epsilon \in (1/n, 1)$, $\delta \in (0, 1/2)$, $x \in \mathbb{C}^n$ and any integer $k \geq 1$, if $R^* \geq ||x||_\infty/\mu$, $R^* = n^{O(1)}$, $\mu^2 \geq ||x_{[n]\setminus[k]}||_2^2/k$, $\mu^2 = O(||x_{[n]\setminus[k]}||_2^2/k)$ and $\alpha > 0$ is smaller than a function of δ,* $\textsc{SparseFFT}(\hat{x}, k, \epsilon, R^*, \mu)$ *(Algorithm 3) solves the ℓ_2/ℓ_2 sparse recovery problem using $O_\delta(k \log n) + O(\frac{1}{\epsilon} k \log n)$ samples and $O_\delta(\frac{1}{\epsilon} k \log^{4+\delta} n)$ time with at least $4/5$ success probability.*

We now discuss the technical difficulties that our approach overcomes. In this discussion we concentrate mainly on the estimation problem, as it is easier than sparse recovery, but at the same time exhibits all the relevant technical challenges. We first describe known sample optimal and efficient solutions that use arbitrary linear measurements, and then outline the difficulties that one faces when working with Fourier measurements.

Estimation and sparse recovery with arbitrary linear measurements. If arbitrary linear measurements are allowed, one takes, multiple times, a set of $B = O(k)$ linear measurements of the form $\tilde{u}_j = \sum_{i:h(i)=j} s_i x_i$ for a random hash function $h : [n] \to [B]$ and random signs $s_i \in \{-1, +1\}$. Since we are hashing in a number of buckets a constant factor (say, 100) larger than the sparsity of the signal, a large fraction (say, $\approx 90\%$) of the top k components are likely to be isolated in a bucket, and not have too much noise (i.e. elements other than the top k) hash into the same bucket. For such isolated elements we can approximate their value *up to the noise that hashes into the same bucket* (in the case of sparse recovery, we perform $O(\log(n/k))$ specially crafted linear measurements using the same hash function h to recover the identity of the isolated element). This lets us estimate (resp. recover) $\approx 90\%$ of the top k elements of the signal, we subtract them off and recurse on the remaining $\approx 10\%$ of the top k elements, hashing into $k/2$ buckets this time. In general, for $t = 1, 2, \ldots, O(\log k)$ we choose

[2] In this paper we are only interested in algorithms that work for worst case signals. If probabilistic assumptions on the signal are made, better results are possible in some settings (see, e.g. [GHI+13]).

a random hash function $h_t : [n] \to [B_t]$, where $B_t = 100k/2^{t-1}$, say (in the case of sparse recovery we take $O(\log(n/k))$ measurements using each of these hash functions). One can show [GLPS10] that after $O(\log k)$ iterations of the hashing, recovery and subtraction process we recover an approximation to x that satisfies (1) (resp. (3) in case of recovery). The sample complexity of this process is dominated by the sample complexity of the first iteration, where we use $B_1 = 100k$ buckets, resulting in a $O(k)$ (resp. $O(k \log(n/k))$) bound on the sample complexity overall. Note that the recovery process only uses every hash function h_t once, at step t: those elements that are isolated under this hashing are perfectly recovered and essentially 'disappear' from the system, so h_t can be discarded!

A natural approach to estimation and recovery with Fourier measurements and why it fails. In order to achieve $O(k)$ (resp. $O(k \log n)$) sample complexity using Fourier measurements (i.e. in Sparse FFT) it seems natural to revisit the original idea used in recovery from arbitrary linear measurements that we outlined above. More precisely, we could follow the strategy of choosing, for $t = 1, 2, \ldots, O(\log k)$, a random hash function $h_t : [n] \to [B_t]$, where $B_t = 100k/2^{t-1}$. The problem is that in order to ensure that we hash into B buckets at the cost of $O(B)$ samples, we need to commit to working with rather low quality buckets implemented using crude filters (see section 2) and this causes 'leakage' between hash buckets. Given this complication, it is not clear at all if estimation (resp. recovery) can be made to work: while with 'ideal' hashing each element isolated in a hashing was identified and estimated *up to amount of noise in its bucket*, here due to the leakage of our simple filters identification of nominally isolated elements can be precluded by interference from other head elements! This means that the elements that were isolated in the first hashing do not 'disappear' from the system (as they essentially do with 'ideal' hashing described above in the context of arbitrary linear measurements), but are reduced in value by only about a constant factor, and will influence the recovery process using the second hashing etc. To put this in perspective, note that when for each $t > 10$, say, we hash into $B_t = 100k/2^{t-1}$ buckets, we generally get $\Omega(k)$ original elements hashing to $\ll k$ buckets! These elements have of course been reduced in value somewhat, but not to the extent that their contribution to $B_t \approx k/2^{t-1}$ buckets is negligible.

The discussion above implies that two difficulties must be overcome to achieve $O(k)$ (resp. $O(k \log n)$) sample complexity. First, since one round of hashing can at most reduce the 'isolated' elements in the residual by a constant factor, $\Omega(\log n)$ iterations are necessary. Furthermore, the process must be set up in such a way that the $\Omega(\log n)$ iterations operate on the same hash functions, and at the same time no adversarial correlations arise to hinder the estimation process. The second difficulty is more subtle, but the harder one to deal with – this is exactly where our main contribution comes in. Note that if several levels of hashing are used, as above there could be elements whose total contribution to estimation error *over all levels* $t > 1$ *is* $\omega(1)$. Indeed, it is easy to see that some of the top k elements will participate in repeated collisions for many values of $t > 1$. Such elements could pose significant difficulties, as they introduce large errors to the identification and estimation process. This issue arises because we reuse hashings that hash $\Omega(k)$ elements into $\ll k$ buckets. Thus, we cannot hope to rely on isolation properties that all prior work is based on, since there are more elements to be estimated than buckets.

Our techniques: a new hashing scheme and isolation on average. To overcome the difficulties outlined above, we use the following approach. As above, we choose a sequence of hash functions h_t that hash the signal into a geometrically decreasing number of buckets. However, a crucial modification is that for each t we repeat the hashing process independently R_t times for an increasing sequence R_t (we use a geometrically increasing sequence; our hashings are denoted by $h_{t,s}$, $s = 1, \ldots, R_t$ for each t). As we show below in Section 3, the independent repetitions ensure, at a high level, that despite the fact that most elements collide in multiple hashings, the fraction of such collisions is small, ensuring that estimation errors do not propagate – see Lemma 3.1 and Remark 3.2 after the lemma.

We give a formal analysis of our scheme in the rest of the paper, and provide intuition as to why our scheme fixes the problem outlined above now. Specifically, we would like to see that the head elements do not contribute a large fraction of their weight as estimation error in hashings $h_{t,s}$ for $t > 1$. The reason is that, as we show below, given the hash functions $\{h_{t,s}\}$ the set S of head elements can be partitioned into sets $S = S_1 \cup S_2 \cup \ldots \cup S_T, |S_1| \gg |S_2| \gg \ldots \gg |S_T|$ so that for every $t > 1$ every element of S collides with at least one element of S_t in *no more than* $R_t^{1-\delta}$ *out of the* R_t *hashings* $h_{t,s}$ *at iteration* t, for some constant $\delta > 0$ (choosing δ small improves runtime, at the expense of sample complexity; any small constant $\delta > 0$ leads to asymptotically sample optimal results). Thus, even though there are many collisions, **on average over** $s \in [1 : R_t]$ every element in S collides with at most $\approx R_t^{-\delta}$ elements of S_t – we refer to this property as 'isolation on average'. Since we choose the number of hashings R_t to increase geometrically, the error contributed by an element of S *over all hashings* is no more than $\sum_{t\geq 1} R_t^{-\delta} = O_\delta(R_t^{-\delta}) \ll 1$. This fact allows us to argue that iterative decoding converges (see section 3). Achieving small runtime with such a scheme requires a delicate balance of parameters, which we exhibit in Section 4.

Our techniques: majorizing sequences for controlling residual signals. Lastly, one should note that the discussion above rests heavily on our ability to control the sequence of residual signals that arise throughout the update process (both in estimation and recovery). We achieve this by showing that residual signals arising during the update process are *majorized* by short (polylogarithmic length) sequence of signals (referred to as a *majorizing sequence*). See Section 4.2 for the application in estimation and Section 5.3 for the application in recovery.

Significance for future work. We feel that the idea of 'isolation on average' may prove useful in further developments in the area. For example, it would be interesting to see if measurement reuse using our techniques can improve sample complexity of to sublinear algorithms for *model based sparse recovery* from Fourier measurements, i.e. to Sparse FFT algorithms that *exploit structure of input signals beyond the sparsity assumption* (the a sublinear time algorithm for model based Sparse FFT for the block-sparse model was recently presented in [CKSZ17]). A strong step in this direction would consist of removing the reliance of our techniques on the ℓ_1 norm of the residual signal as the measure of progress, and introducing an approach to measurement reuse while provably reducing the ℓ_2 norm of the residual during the iterative process.

Organization. The proofs of Theorem 1.1 and Theorem 1.2 rely on a shared set of lemmas that enable analysis via 'isolation on average', with the main technical lemma being Lemma 3.1 (see also Remark 3.2 after the lemma). We present these lemmas first (Sections 2 and 3), then prove Theorem 1.1 (Section 4) as it is less notationally heavy but still uses all the main technical ideas, and then prove Theorem 1.2 (Section 5). Proofs omitted from the main body of the paper are given in the Appendices.

2 Preliminaries and basic notation

For a positive even integer a we will use the notation $[a] = \{-\frac{a}{2}, -\frac{a}{2}+1, \ldots, -1, 0, 1, \ldots, \frac{a}{2}-1\}$. We will consider signals of length n, where n is a power of 2. We use the notation $\omega = e^{2\pi i/n}$ for the root of unity of order n. The forward and inverse Fourier transforms are given by

$$\hat{x}_f = \frac{1}{\sqrt{n}} \sum_{i\in[n]} \omega^{-if} x_i \text{ and } x_j = \frac{1}{\sqrt{n}} \sum_{f\in[n]} \omega^{jf} \hat{x}_f \tag{4}$$

respectively, where $f, j \in [n]$. We will denote the forward Fourier transform by $\mathcal{F}$. Note that we use the orthonormal version of the Fourier transform. Thus, we have $||\hat{x}||_2 = ||x||_2$ for all $x \in \mathbb{C}^n$ (Parseval's identity). We assume that entries of x are integers bounded by a polynomial in n.

2.1 Filters, hashing and pseudorandom permutations

We will use pseudorandom spectrum permutations, which we now define. We write $\mathcal{M}_{odd}$ for the set of odd numbers between 1 and n. For $\sigma \in \mathcal{M}_{odd}, q \in [n]$ and $i \in [n]$ let $\pi_{\sigma,q}(i) = \sigma(i-q) \mod n$. Since $\sigma \in \mathcal{M}_{odd}$, this is a permutation. Our algorithm will use π to hash heavy hitters into B buckets, where we will choose $B \approx k$. We will often omit the subscript σ, q and simply write $\pi(i)$ when σ, q is fixed or clear from context. For $i \in [n]$ we let $h(i) := \text{round}((B/n)\pi(i))$ be a hash function that maps $[n]$ to $[B]$, and for $i, j \in [n]$ we let $o_i(j) = \pi(j) - (n/B)h(i)$ be the "offset" of $j \in [n]$ relative to $i \in [n]$. We always have B a power of two.

Definition 2.1. *Suppose that σ^{-1} exists $\mod n$. For $a, q \in [n]$ we define the permutation $P_{\sigma,a,q}$ by $(P_{\sigma,a,q}\hat{x})_i = \hat{x}_{\sigma(i-a)}\omega^{i\sigma q}$.*

Lemma 2.2. $\mathcal{F}^{-1}(P_{\sigma,a,q}\hat{x})_{\pi_{\sigma,q}(i)} = x_i\omega^{a\sigma i}$

The proof is given in [IK14] and we do not repeat it here. Define

$$\text{Err}_k(x) = \min_{k-\text{sparse } y} ||x-y||_2 \text{ and } \mu^2 = \text{Err}_k^2(x)/k. \tag{5}$$

In this paper, we assume knowledge of μ (a constant factor upper bound on μ suffices). We also assume that the signal to noise ratio is bounded by a polynomial in the length n of the signal, namely that $R^* := ||x||_\infty/\mu \le n^C$ for a constant $C > 0$. It will be convenient to use the notation $\mathbb{B}_\infty(x, r)$ to denote the interval of radius r around x: $\mathbb{B}_\infty(x, r) = \{y \in [n] : |x-y|_\circ \le r\}$, where $|x-y|_\circ$ is the circular distance on $\mathbb{Z}_n$. For a real number a we write $|a|_+$ to denote the positive part of a, i.e. $|a|_+ = a$ if $a \ge 0$ and $|a|_+ = 0$ otherwise.

We will use the following

Definition 2.3 (Flat filter with B buckets and sharpness F)**.** *A sequence $G \in \mathbb{R}^n$ symmetric about zero with Fourier transform $\widehat{G} \in \mathbb{R}^n$ is called a* flat filter with B buckets and sharpness F *if* **(1)** $G_j \in [0,1]$ *for all $j \in [n]$;* **(2)** $G_j \ge 1 - \left(\frac{1}{4}\right)^{F-1}$ *for all $j \in [n]$ such that $|j| \le \frac{n}{2B}$; and* **(3)** $G_f \le \left(\frac{1}{4}\right)^{F-1}\left(\frac{n}{B|j|}\right)^{F-1}$ *for all $j \in [n]$ such that $|j| \ge \frac{n}{B}$.*

We use a construction of such filters from [CKSZ17]:

Lemma 2.4 ([CKSZ17], Lemma 2.1)**.** (Compactly supported flat filter with B buckets and sharpness F) *Fix the integers (n, B, F) with n a power of two, $B < n$, and $F \ge 2$ an even number. There exists an (n, B, F)-flat filter $G \in \mathbb{R}^n$, whose Fourier transform $\widehat{G}$ is supported on a length-$O(FB)$ window centered at zero in time domain.*

Note that most of the mass of the filter is concentrated in an interval of side $O(n/B)$, approximating the "ideal" filter (whose value would be equal to 1 for entries within the square and equal to 0 outside of it). Note that for each $i \in [n]$ one has $G^{-1}_{o_i(i)} \le 2$. We refer to the parameter F as the *sharpness* of the filter. Our hash functions are not pairwise independent, but possess a property that still makes hashing using our filters efficient:

Lemma 2.5 (Lemma 3.2 in [IK14])**.** *Let $i, j \in [n]$. Let σ be uniformly random odd number between* 1 *and n. Then for all $t \ge 0$ one has* $\Pr[|\sigma(i-j)|_\circ \le t] \le 2(2t/n)$.

2.2 Measurements of the signal, notation for estimation error and basic bounds

Pseudorandom spectrum permutations combined with a filter G give us the ability to 'hash' the elements of the input signal into a number of buckets (denoted by B). We formalize this using the notion of a *hashing*. A hashing is a tuple consisting of a pseudorandom spectrum permutation π, target number of buckets B and a sharpness parameter F of our filter, denoted by $H = (\pi, B, F)$. Formally, H is a function that maps a signal x to B signals, each corresponding to a hash bucket, allowing us to solve the k-sparse recovery problem on input x by reducing it to 1-sparse recovery problems on the bucketed signals. We give the formal definition below.

Definition 2.6 (Hashing $H = (\pi, B, F)$)**.** *For a permutation $\pi = (\sigma, q)$, parameters $B > 1$ and F, a* hashing *$H := (\pi, B, F)$ is a function mapping a signal $x \in \mathbb{C}^n$ to B signals $H(x) = (u_s)_{s\in[B]}$, where $u_s \in \mathbb{C}^n$ for each $s \in [B]$, such that for each $i \in [n]$ $u_{s,i} = \sum_{j\in[n]} G_{\pi(j)-(n/B)\cdot s} x_j \omega^{i\sigma j} \in \mathbb{C}$, where G is a filter with B buckets and sharpness F constructed in Lemma 2.4.*

For a hashing $H = (\pi, B, F), \pi = (\sigma, q)$ we sometimes write $P_{H,a}, a \in [n]$ to denote $P_{\sigma,a,q}$.

Definition 2.7 (Measurement $m = m(x, H, a)$)**.** *For a signal $x \in \mathbb{C}^n$, a hashing $H = (\pi, B, F)$ and a parameter $a \in [n]$, a* measurement *$m = m(x, H, a) \in \mathbb{C}^B$ is the B-dimensional complex valued vector of evaluations of a hashing $H(x)$ at a point $a \in [n]$, i.e. for $s \in [B]$ $m_s = \sum_{j\in[n]} G_{\pi(j)-(n/B)\cdot s} x_j \omega^{a\sigma j}$, where G is a filter with B buckets and sharpness F constructed in Lemma 2.4.*

We access the signal x in Fourier domain via the function $\textsc{HashToBins}(\hat{x}, \chi, (H, a))$, which evaluates the hashing H of residual signal $x - \chi$ at point $a \in [n]$, i.e. computes the measurement $m(x, H, a)$ (the computation is done with polynomial precision). We will use the following lemma, which is rather standard (the proof is given in Appendix B.1 for completeness):

Lemma 2.8. $\textsc{HashToBins}$*$(\widehat{x}, \chi, (H, a))$, where $H = (\pi, B, F)$, computes $u \in \mathbb{C}^B$ such that for any $i \in [n]$, $u_{h(i)} = \Delta_{h(i)} + \sum_j G_{o_i(j)}(x - \chi)_j \omega^{a\sigma j}$, where G is the filter defined in section 2, and for all $i \in [n]$ we have that $\Delta^2_{h(i)} \le \|\chi\|_2^2 \cdot n^{-c}$ is a negligible error term (and $c > 0$ is an absolute constant that governs the precision that semi-equispaced FFT, i.e. Lemma E.1, is invoked with). It takes $O(BF)$ samples, and $O(F \cdot B \log B + \|\chi\|_0 \log n)$ time.*

We now introduce relevant notation for bounding the error induced by our measurements in locating or estimating an element $i \in [n]$. For a hashing $H = (\pi, B, F)$ and an evaluation point $z \in [n]$, we have by Definition 2.7

$$m_{h(i)}(x, H, z) = \sum_{j\in[n]} G_{o_i(j)} x_j \omega^{z\sigma j},$$

where the filter $G_{o_i(j)}$ is the filter corresponding to hashing H (note that $o_i(j)$ implicitly depends on π). In particular, one has:

$$G^{-1}_{o_i(i)} m_{h(i)} \omega^{-z\sigma i} = x_i + \underbrace{G^{-1}_{o_i(i)} \sum_{j\in[n]\setminus\{i\}} G_{o_i(j)} x_j \omega^{z\sigma(j-i)}}_{\text{noise term}}$$

A common idea underlying our analysis of estimation and recovery is to split the estimation/recovery error induced on an element i into the contribution from the carefully defined 'head' of the signal and the contribution from the 'tail'. The 'head' of the signal is denoted by a set $S \subseteq [n]$ throughout the paper. For each $i \in [n]$ we write

$$G^{-1}_{o_i(i)} m_{h(i)} \omega^{-z\sigma i} = x_i + \underbrace{G^{-1}_{o_i(i)} \cdot \sum_{j\in S\setminus\{i\}} G_{o_i(j)} x_j \omega^{z\sigma(j-i)}}_{\text{noise from 'heavy' elements}} + \underbrace{G^{-1}_{o_i(i)} \cdot \sum_{j\in[n]\setminus(S\cup\{i\})} G_{o_i(j)} x_j \omega^{z\sigma(j-i)}}_{\text{'tail' noise}} \tag{6}$$

We now define special notation for the two noise terms in (6). These two noise terms will be handled very differently in our analysis.

Noise from heavy hitters. The first term in (6) corresponds to noise from $x_{S\setminus\{i\}}$, i.e. noise from 'head' of the signal. For every $i \in S$, hashing H we let

$$e_i^{head}(H, x) := G^{-1}_{o_i(i)} \cdot \sum_{j\in S\setminus\{i\}} G_{o_i(j)} |x_j|. \tag{7}$$

Remark 2.9. *Note that e^{head} depends implicitly on the set S. We do not make this dependence explicit to avoid complicated notation, but state which set S the quantity e^{head} is defined with respect to whenever this quantity is used. We also note that e^{head} provides a bound on the error induced by the tail of the signal that (for a random hashing) depends on the ℓ_1 norm of the head of the signal.*

We thus get that $e_i^{head}(H, x)$ upper bounds the absolute value of the first error term in (6) *for every value of evaluation point z* (note that $e^{head}(H, x)$ only depends on the hashing H and x). Note that $G \geq 0$ by Lemma 2.4 and Definition 2.3 as long as F is even, which is the setting that we are in. We will often use several hashings to estimate or locate an element i. It is thus convenient to define, for a sequence of hashings $H_1, \ldots, H_r$

$$e_i^{head}(\{H_r\}, x) := \text{quant}_r^{1/5} e_i^{head}(H_r, x), \tag{8}$$

where for a list of reals $u_1, \ldots, u_s$ and a number $f \in (0, 1)$ we let $\text{quant}^f(u_1, \ldots, u_s)$ denote the $\lceil f \cdot s \rceil$-th largest element of $u_1, \ldots, u_s$.

Tail noise. To capture the second term in (6) (corresponding to tail noise), we define, for any $i \in [n], z \in [n]$, permutation $\pi = (\sigma, q)$ and hashing $H = (\pi, B, F)$

$$e_i^{tail}(H, z, x) := \left| G_{o_i(i)}^{-1} \cdot \sum_{j \in [n] \setminus (S \cup \{i\})} G_{o_i(j)} x_j \omega^{z\sigma(j-i)} \right|. \tag{9}$$

Remark 2.10. *Note that e^{tail} depends implicitly on the set S. We do not make this dependence explicit to avoid complicated notation, but state which set S the quantity e^{tail} is defined with respect to whenever this quantity is used. We also note that e^{tail} provides a bound on the error induced by the tail of the signal that (for a random hashing and a random evaluation point z) depends on the ℓ_2 norm of the tail (as opposed to the ℓ_1 norm used by e^{head}).*

With this definition in place $e_i^{tail}(H, z, x)$ upper bounds the absolute value of second term in (6). We will sometimes use several hashings and values z to obtain better estimates. For a sequence $\{(H_r, z_r)\}_{r=1}^{r_{max}}$ for some $r_{max} \geq 1$ we let

$$e_i^{tail}(\{H_r, z_r\}, x) := \text{quant}_r^{1/5} \left| G_{o_i(i)}^{-1} \cdot \sum_{j \in [n] \setminus (S \cup \{i\})} G_{o_i(j)} x_j \omega^{z\sigma(j-i)} \right|, \tag{10}$$

where $o_i(j)$ on the rhs implicitly depends on the hashing H.

The definitions above are sufficient for our analysis of the signal estimation procedure in Section 4. The analysis of the sparse recovery procedure requires several further specialized definitions, which are presented in Appendix C. With the definitions above we can state the following simple guarantees on the performance of a basic estimation procedure that is the main building block of our analysis of the more powerful ESTIMATE primitive in Section 4.

Lemma 2.11 (Bounds on estimation quality for Algorithm 4)**.** *For every $x, \chi \in \mathbb{C}^n$, every $L \subseteq [n]$, every set $S \subseteq [n]$ the following conditions hold for functions e^{head} and e^{tail} defined with respect to S (see* (7) *and* (9)*). If r_{max} is larger than an absolute constant, then for every sequence $H_r = (\pi_r, B, F), r = 1, \ldots, r_{max}$ of hashings and every sequence $a_1, \ldots, r_{max}$ of evaluation points the output w of*

$$\textsc{EstimateValues}(\chi, L, \{(H_r, a_r, m(x, H_r, a_r))\}_{r=1}^{r_{max}})$$

satisfies, for each $i \in L$

$$|w_i - (x - \chi)_i| \leq 2 \cdot quant_r^{1/5} e_i^{head}(H_r, x - \chi) + 2 \cdot quant_r^{1/5} e_i^{tail}(H_r, a_r, x - \chi) + n^{-\Omega(c)},$$

where $c \geq 2$ is an absolute constant that governs the precision of our approximate semi-equispaced FFT computations (see HASHTOBINS*, Lemma 2.8). The sample complexity is bounded by $O(FBr_{max})$, and the runtime by $O((F \cdot B \cdot \log n + ||\chi||_0 \log n + |L|) \cdot r_{max})$.*

The proof of the lemma is given in Appendix B. The proof is rather standard modulo our definitions of e^{head} and e^{tail}, as well as the fact that the statement of the lemma is entirely deterministic. We will later apply this lemma to random hashings and evaluation points, but the deterministic nature of the claim will be crucial in analyzing measurement reuse.

As both our ESTIMATE and SPARSEFFT algorithms (Section 4 and Section 5) respectively iteratively update the signal, we will need to analyse the performance of ESTIMATEVALUES on various residual signals derived from the original input signal x. The notion of a *majorant* is central to this part of our analysis:

Definition 2.12 (Majorant)**.** *For any $S \subseteq [n]$ and any $x, y \in \mathbb{C}^n$ we say that y is an majorant for x with respect to S if $|x_i| \leq |y_i|$ for all $i \in S$.*

With this definition and definition of e^{head} above the following crucial lemma follows immediately:

Lemma 2.13. *For every hashing H, every set $S \subseteq [n]$ one has for every pair $x, y \in \mathbb{C}^n$ that if $x \prec_S y$, then for every $i \in [n]$ $e_i^{head}(H, x) \leq e_i^{head}(H, y)$.*

Proof. Recall that by (7) one has $e_i^{head}(H, x) = G_{o_i(i)}^{-1} \cdot \sum_{j \in S \setminus \{i\}} G_{o_i(j)} |x_j|$. Since $G \geq 0$ by Definition 2.3 and Lemma 2.4, we have by using $x \prec_S y$ that $e_i^{head}(H, x) = G_{o_i(i)}^{-1} \cdot \sum_{j \in S \setminus \{i\}} G_{o_i(j)} |x_j| \leq G_{o_i(i)}^{-1} \cdot \sum_{j \in S \setminus \{i\}} G_{o_i(j)} |y_j| = e_i^{head}(H, y)$ as required.

□

3 Isolating partitions

In this section we prove the main lemmas that allow us to reason about performance of the SNR reduction process in our algorithms (Algorithm 2 and Algorithm 3). Both algorithms perform SNR reduction (see lines 16 to 22 in Algorithm 2 and lines 25-36 in Algorithm 3) using two loops. The first loop (over r), controls the ℓ_1 norm of the signal, with the (upper bound on the) norm being reduced by a factor of 4 in each iteration. This reduction is achieved via a sequence of calls to ESTIMATEVALUES (in ESTIMATE, Algorithm 2) or LOCATESIGNAL (in Section C) using a separate collection of hashings for each $t = 1, \ldots, T$. Our correctness analysis for this process proceeds by showing that, with high constant probability over the choice of the hashings $\{\{H_{t,s}\}_{s=1}^{R_t}\}_{t=1}^{T}$ any set S of size $\approx k$ the hashings induce a partition of S into at most T sets $S_1 \cup \ldots \cup S_T$ such that hashings used in the t-th round allow the algorithm to make progress on elements in S_t. The main result of this section is a formalization of this claim, achieved by Lemma 3.1.

Lemma 3.1. *For every integer $k \geq 1$, every $S \subseteq [n], |S| \leq k$, every $\delta \in (0, 1/2)$, if the parameters B_t, R_t are selected to satisfy* **(p1)** *$R_t = C_1 \cdot 2^t$ and* **(p2)** *$B_t \geq C_2 \cdot k/R_t^2$ for every $t \in [0 : T]$, where C_1 is a sufficiently large constant and C_2 is sufficiently large as a function of C_1 and δ, then the following conditions hold.*

For every collection of hashings $\{\{H_{t,s}\}_{s=1}^{R_t}\}_{t=1}^{T}$, if the filters used in hashings $H_{t,s}$ are at F-sharp for even $F \geq 6$ for every $t \in [1 : T]$, and S admits a δ-isolating partition (as per Definition 3.7) $S = S_1 \cup \ldots \cup S_T$ with respect to $\{H_{t,s}\}$, then for every $x, \chi \in \mathbb{C}^n$, $x' = x - \chi$, for every $t \in [1 : T]$ one has $||e_{S_t}^{head}(\{H_{t,s}\}_{s \in [1:R_t]}, x')||_1 \leq 20 R_t^{-\delta} ||x'_S||_1$.

Remark 3.2. *Note that the result of Lemma 3.1 implies that the cumulative error induced by the entire set S of 'heavy' coefficients on S_t is only a $\approx R_t^{-\delta}$ fraction of the ℓ_1 norm of x'_S, despite the fact that when estimating S_t we hash into $B_t \ll k$ buckets, and in general each bucket will contain many elements of S. Furthermore, if we choose R_t to increase fast enough so that $\sum_{t \geq 1} R_t^{-\delta} \ll 1$, we get that the cumulative contribution of elements in*

S to estimation/location error over all $t \geq 1$ is less than 1, meaning that errors do not accumulate much. This is exactly what we achieve by setting $R_t = C_1 2^t$ for a large constant $C_1 > 1$ – see proof of Lemma 4.1 in Section 4.

In the rest of the section we first introduce relevant notation and in particular define the central notion of an *isolating partition* of the set S of head elements (in section 3.1), then prove that any fixed set S of size about k admits an isolating partition with at least high constant probability over the choice of hashings $\{\{H_{t,s}\}_{s=1}^{R_t}\}_{t=1}^{T}$ (section 3.2), and show how to construct the partition efficiently (Lemma 3.9) if the set S is given explicitly (used in Algorithm 2, line 13). Using this result we then give a proof of Lemma 3.1.

3.1 Main definitions

Let S be any subset of $[n]$ (we will later instantiate S to the set of 'large' elements of x). We now define a decomposition of the set S into $T = \frac{1}{1-\delta}\log_2 \log(k+1) + O(1)$ disjoint sets $S_1, S_2, \ldots, S_T$ with respect to a sequence $1 \leq R_0 \leq R_1 \leq R_2 \leq \ldots \leq R_T$ and hashings $\{\{H_{t,s}\}_{s=1}^{R_t}\}_{t=1}^{T}$. We start with several auxiliary definitions.

Definition 3.3 (t-Collision)**.** *We say that an element $a \in [n]$ participates in a t-collision with another element $b \in [n]$ under hashing $H = (\pi, B, F)$ if a hashes within at most t buckets of b under H, i.e. if $|\pi(a) - \pi(b)| \leq \frac{n}{B}(t-1)$.*

Definition 3.4 (δ-bad element)**.** *For $\delta \in (0,1)$, for each $t \in [1:T]$, sequence $1 \leq R_0 \leq R_1 \leq \ldots \leq R_T$, where $T \geq 1$ is an integer, we say that an element a of S is* δ-bad for *S_t with respect to a partition $S = S_1 \cup S_2 \cup \ldots \cup S_T$ and hashings $\{\{H_{t,s}\}_{s=1}^{R_t}\}_{t=1}^{T}$ if a participates in an R_t-collision with at least one element of S_t under more than a $R_t^{-\delta}$ fraction of hashings $H_{t,1}, \ldots, H_{t,R_t}$.*

Definition 3.5 (λ-crowded element)**.** *For a hashing $H = (\pi, B, F)$ and a real number $\lambda \in (0,1)$, an element $a \in [n]$ is λ-crowded at scale $q \geq 0$ by a set $Q \subseteq [n]$ if $\left|\mathbb{B}\left(\pi(a), \frac{n}{B} \cdot 2^q\right) \cap \pi(Q \setminus \{a\})\right| \geq \lambda 2^{2q}$. We say that an element a is simply λ-crowded if it is λ-crowded at least at one scale $q \geq 0$.*

Remark 3.6. *The intuition for the definition above is that if the permutation π was pairwise independent, then for every $a \in [n]$ the expectation of $\left|\mathbb{B}\left(\pi(a), \frac{n}{B} \cdot 2^q\right) \cap \pi(Q \setminus \{a\})\right|$ would be about 2^q if $|Q| \leq B$, i.e. about the number of buckets that fall into the interval. We say that an element is λ-crowded when the number of elements in its vicinity exceeds $\lambda 2^{2q}$ for at least one scale q, i.e. exceeds expectation by a $\lambda 2^q$ factor. Our choice of $\lambda = R_t^{-3}$ serves the purpose of enforcing that no element of S_t is hashed too close to another element of S_t, and no (large) neighborhood of any element of S_t it too crowded. These two parameter regimes have somewhat distinct applications in the proof of Lemma 3.1 – see footnotes 3 and 4 in the proof of the lemma on pages 13 and 14 respectively.*

Definition 3.7 (δ-isolating partition)**.** *For $\delta \in (0,1)$, for any $k \geq 1$ and any $S \subseteq [n], |S| \leq k$, a partition $S = S_1 \cup S_2 \cup \ldots \cup S_T$ of $S \subseteq [n]$ into disjoint subsets is* δ-isolating *with respect to a sequence of integers $1 \leq R_0 \leq R_1 \leq \ldots \leq R_T$ and hashings $\{H_{t,s}\}_{s=1}^{R_t}$ for $t = [1:T]$ if the following conditions are satisfied for each $t \in [1:T]$:*

(1) $|S_t| \leq k \cdot \frac{R_0}{R_{t-1}} 2^{-2^{(1-\delta)\cdot(t-1)}+1}$ *(the sizes of S_t's decay doubly exponentially);*

(2) *no element of S_t is R_t^{-3}-crowded by S_t under any of $\{H_{t,s}\}_{s=1}^{R_t}$ (elements of S_t are rather uniformly spread under hashings $H_{t,s}$);*

(3) *no element of S_t R_t-collides with an element of S that is δ-bad for S_t under any of $\{H_{t,s}\}_{s=1}^{R_t}$ (collisions between S and S_t are rare).*

Note that the bound on the size of S_1 is at most k, which will be trivial for our instantiation of S. Nontrivial decay of the S_t starts at $t = 2$. Also note that property **(3)** is exactly why we call our approach 'isolation on average': as the proof of Lemma 3.1 below shows, the fact that no element of S that is δ-bad for S_t collides with S_t under any of the hashings implies that every element of S has a limited contribution to estimation error, and errors do not propagate.

3.2 Existence of isolating partitions of S

In this section we prove that any set S of size at most k admits an isolating partition with respect to a random set of hashings as in Algorithm 3 with at least high constant probability. We prove this claim by giving an algorithm that we show constructs such a partition successfully with high constant probability. The algorithm is Algorithm 1, presented below. We prove that Algorithm 1 terminates correctly in Lemma 3.8 below, assuming that the number of hashings at each step t and their parameters are chosen appropriately.

If the set S is known explicitly, the partition can be constructed efficiently by running Algorithm 1 – the details are given in Lemma 3.9. An efficient construction is needed in our sample efficient primitive (see Algorithm 2). Our sample-optimal Sparse FFT algorithm is oblivious to the actual partition, as its task is to identify the set S. However, as we show in Section 5, the existence of an isolating partition of S is sufficient for the algorithm to work.

Algorithm 1 Construction of an isolating partition $\{S_j\}$

```
procedure ConstructPartition($\{\{H_{t,s}\}_{s\in[1:R_t]}\}_{t=1}^{T}$)        ▷ Hashings sampled uniformly
    $S_1^1 \leftarrow S, t \leftarrow 1$
    while $S_t^t \neq \emptyset$ do
        $\text{Bad}_t \leftarrow$ {elements of $S$ that are $\delta$-bad wrt $S_t^t$ under $\{H_{t,s}\}_{s\in[1:R_t]}$}
        $U_t \leftarrow$ {elements $a \in S_t^t$ that $R_t$-collide with $\text{Bad}_t$ under at least one of $\{H_{t,s}\}_{s\in[1:R_t]}$}
        $V_t \leftarrow$ {elements $a \in S_t^t$ that are $R_t^{-3}$-crowded by $S_t^t$ under at least one of $\{H_{t,s}\}_{s\in[1:R_t]}$}
        Set $S_{t+1}^{t+1} \leftarrow \text{Bad}_t \cup U_t \cup V_t$ and $S_j^{t+1} \leftarrow S_j^t \setminus S_{t+1}^{t+1}$ for $j = 1, \ldots, t$
        $t \leftarrow t + 1$
    end while
    return the partition $\{S_j^t\}_{j=1}^t$
end procedure
```

We now argue that Algorithm 1 constructs an isolating partition of any set $S \subseteq [n]$ that satisifies $|S| \leq k$ with at least high constant probability:

Lemma 3.8. *For every integer $k \geq 1$, every $S \subseteq [n], |S| \leq k$, every $\delta \in (0, 1/2)$, if the parameters B_t, R_t are selected to satisfy* **(p1)** *$R_t = C_1 \cdot 2^t$ and* **(p2)** *$B_t \geq C_2 \cdot k/R_t^2$ for every $t \in [0 : T]$, where C_1 is a sufficiently large constant and C_2 is sufficiently large as a function of C_1 and δ, then the following conditions hold.*

With probability at least $1 - 1/25$ over the choice of hashings $\{\{H_{t,s}\}_{s\in[1:R_t]}\}_{t=1}^{T}$ Algorithm 1 terminates in $T = \frac{1}{1-\delta}\log_2 \log(k+1) + O(1)$ steps. When the algorithm terminates, the output partition $\{S_j\}_{j=1}^{T}$ is isolating as per Definition 3.7.

Proof Outline: The proof consists of two parts: showing that once the algorithm terminates, the resulting partition is δ-isolating, and then showing that the algorithm terminates with large constant probability as long as parameters are set appropriately (to satisfy **p1** and **p2**). The first part is rather direct from definitions, and is presented in Appendix A together with the full proof of Lemma 3.8. We now outline the proof of the second part, i.e. that the algorithm terminates.

The crux of the proof consists of bounding the sizes of the sets V_t, U_t, Bad_t obtained at time step $t = 1, \ldots, T$. It is easiest to start with the intuition for V_t, i.e. elements in S_t^t that are crowded by other elements of S_t^t. An element a is crowded by S_t^t under permutation π if there are too many elements of S_t^t in a neighborhood of a certain size of a. The definition of being crowded (Definition 3.5 above) involves the notion of being crowded at a given scale, and one can see that an element a is more likely to fail because of small scales as opposed to large scales. This means that for every $a \in S_t^t$ one has

$$\mathbf{Pr}[a \in V^t] \approx \text{poly}(R_t) \cdot k_t/B_t = \text{poly}(R_t) \cdot k_t/B, \quad \text{(since } B_t = \Theta(B/R_t^2))$$

i.e., up to terms polynomial in R_t, the probability of being crowded is about the probability of $O(1)$-colliding with one other element of S_t^t (see below for a formalization of this claim). Since decay of the size of S_t^t that we will exhibit is doubly exponential in t, factors polynomial in R_t are negligible, as they are only singly exponential in t. Given the expression for $\mathbf{Pr}[a \in V^t]$ above, we get that

$$\mathbf{E}[|V_t|] = \sum_{a \in S_t^t} \text{poly}(R_t) \cdot k_t/B = \text{poly}(R_t) \cdot (k_t/B)^2 \cdot B.$$

This means that if V_t were the only contribution to S_{t+1}^{t+1}, then, discounting the poly(R_t) terms, we would get the recurrence $k_{t+1}/B = (k_t/B)^2$, which implies that $k_t \approx B \cdot 2^{-2^t}$. The poly$(R_t)$ terms do not affect the decay substantially, and in the actual proof below we get that $k_t \leq O(k \cdot 2^{-2^{(1-\delta)t}})$ for a small constant $\delta > 0$.

The more interesting term is the contribution from U_t and Bad$_t$. We first describe the intuition behind the asymptotic growth of Bad$_t$. For an element $a \in S$ we have

$$\mathbf{Pr}[a \;\; R_t\text{-collides with an element of } S_t^t] \approx k_t/B_t = \text{poly}(R_t) \cdot k_t/B \ll 1/(10R_t),$$

since $k_t \approx B \cdot 2^{-2^t}$ and R_t are only singly exponential in t. Note that this means that the expected number of collisions **over all R_t hashings $H_{t,s}$, $s = 1, \ldots, R_t$, is less than $1/10$!** At the same time recall that a is δ-bad (as per Definition 3.4 above) if a collides with at least one element of S_t^t under at least $R_t^{1-\delta}$ hashings $\{H_{t,s}\}_{s=1}^{R_t}$. Since the hashings are independent, we have by Chernoff bounds that the probability of the latter event is bounded by about $e^{-R_t^{1-\delta}} = e^{-C_1^{1-\delta}2^{(1-\delta)t}}$, so we again get doubly exponential decay, but for a different reason this time: by our choice of parameters R_t to grow singly exponentially, and Chernoff bounds show that the probability of getting at least $R_t^{1-\delta}$ collisions while the expected number of collisions is less than $1/10$ decays exponentially in $R_t^{1-\delta}$. This is exactly the point at which we say that most elements of S are 'isolated on average'. A formal version of this argument lets us argue that the size of Bad$_t$ is about $ke^{-C_1^{1-\delta}2^{(1-\delta)t}}$. It remains to bound the size of U_t, i.e. elements that collide with a bad element in at least one of the hashings. Since the number of bad elements is doubly exponentially small and the number of hashings R_t is only single exponential, a union bound essentially shows that this quantity is small as well. Some care is needed in arguing this due to a dependency issue, but the intuition is the same as the one described for Bad$_t$ above. The formal proof is given in Appendix A.
□

If the set S is known explicitly, Algorithm 1 admits a simple efficient implementation (the proof of the lemma is given in Appendix A):

Lemma 3.9. *For any integer $k \geq 1$, any $S \subseteq [n], |S| \leq k$, if the hashings $\{\{H_{t,s}\}_{s \in [1:R_t]}\}_{t=1}^T$ are such that the partition $\{S_j\}_{j=1}^T$ defined by Algorithm 1 is isolating as per Definition 3.7, this partition can be constructed explicitly in time $O\left(\left(\sum_{t=1}^T R_t\right) \cdot |S| \log |S|\right)$.*

3.3 Proof of main technical lemma (Lemma 3.1)

We now prove the main result of this section, Lemma 3.1. This lemma then forms the basis of our sample-efficient estimation primitive, as well as the location primitive. The proof crucially relies on the existence of an isolating partition of the set S of 'head elements' (which is guaranteed by Lemma 3.8 from the previous section with at least high constant probability).

Lemma 3.1 (restated) *For every integer $k \geq 1$, every $S \subseteq [n], |S| \leq k$, every $\delta \in (0, 1/2)$, if the parameters B_t, R_t are selected to satisfy* **(p1)** $R_t = C_1 \cdot 2^t$ *and* **(p2)** $B_t \geq C_2 \cdot k/R_t^2$ *for every $t \in [0 : T]$, where C_1 is a sufficiently large constant and C_2 is sufficiently large as a function of C_1 and δ, then the following conditions hold.*

For every collection of hashings $\{\{H_{t,s}\}_{s=1}^{R_t}\}_{t=1}^{T}$, if the filters used in hashings $H_{t,s}$ are at F-sharp for even $F \geq 6$ for every $t \in [1 : T]$, and S admits a δ-isolating partition (as per Definition 3.7) $S = S_1 \cup \ldots \cup S_T$ with respect to $\{H_{t,s}\}$, then for every $x, \chi \in \mathbb{C}^n$, $x' = x - \chi$, for every $t \in [1 : T]$ one has $||e_{S_t}^{head}(\{H_{t,s}\}_{s \in [1:R_t]}, x')||_1 \leq 20 R_t^{-\delta} ||x'_S||_1$.

Proof. Fix $t \in [1 : T]$. For every $s \in [1 : R_t]$ by (7) for all $i \in S$ we have

$$e_i^{head}(H_{t,s}, x') = G_{o_i(i)}^{-1} \cdot \sum_{j \in S \setminus \{i\}} G_{o_i(j)} \cdot |x'_j| \tag{11}$$

where $o = o_{H_{t,s}}$ implicitly depends on the hashing H. We omit the dependence on H when H is fixed to simplify notation. By summing (11) over $i \in S_t$ we get

$$\begin{aligned} e_{S_t}^{head}(H_{t,s}) = \sum_{i \in S_t} G_{o_i(i)}^{-1} \cdot \sum_{j \in S \setminus \{i\}} G_{o_i(j)} \cdot |x'_j| &\leq 2 \sum_{j \in S} |x'_j| \cdot \sum_{i \in S_t \setminus \{j\}} G_{o_i(j)} \\ &= 2 \sum_{j \in S} |x'_j| \cdot D_j^s, \end{aligned} \tag{12}$$

where for all $j \in S$ and $s \in [1 : R_t]$ we let $D_j^s := \sum_{i \in S_t \setminus \{j\}} G_{o_i(j)}$, and used the fact that $G_{o_i(i)} \geq 1/2$ by Definition 2.3 and assumption that $F \geq 6$. Note that D_j^s depends on t, but we omit t to simplify notation. This will not cause confusion since t is fixed for the entire proof. We now bound D_j^s for $j \in S$. We have for $j \in S$

$$\begin{aligned} D_j^s = \sum_{i \in S_t \setminus \{j\}} G_{o_i(j)} &\leq \left| \left\{ i \in S_t \setminus \{j\} : |\pi_{s,t}(i) - \pi_{s,t}(j)|_\circ < \frac{n}{B_t} \cdot R_t \right\} \right| + \sum_{\substack{i \in S_t \setminus \{j\}: \\ |\pi_{s,t}(i) - \pi_{s,t}(j)|_\circ \geq \frac{n}{B_t} \cdot R_t}} G_{o_i(j)} \\ &=: Z_j^s + X_j^s. \end{aligned} \tag{13}$$

We used the fact that $||G||_\infty \leq 1$ by Definition 2.3, **(1)** and assumption that F is even, to go from the first line to the second.

Bounding Z_j^s. We start by showing that $Z_j^s \in \{0, 1\}$ for all s, j. We have by property **(2)** of an isolating partition (see Definition 3.7) that no element of S_t is R_t^{-3}-crowded with respect to S_t[3]. This in particular means that no $i \in S_t$ is R_t^{-3}-crowded at scale $q = 1 + \log_2 R_t$ by S_t, i.e.

$$\left| \mathbb{B}_\infty \left(\pi_{s,t}(i), \frac{n}{B_t} \cdot (2R_t) \right) \cap \pi(S_t \setminus \{i\}) \right| \leq R_t^{-3} 4 \cdot 2^{2q} \leq 4 R_t^{-3} R_t^2 \leq 4/R_t < 1$$

for all $i \in S_t$, as long as $C_1 > 4$ (recall that $R_t = C_1 2^t$ and C_1 is larger than an absolute constant). For every $j \in S$ (as opposed to S_t) and every $a, b \in S_t$ we have by triangle inequality $|\pi_{s,t}(a) - \pi_{s,t}(b)|_\circ \leq |\pi_{s,t}(a) - \pi_{s,t}(j)|_\circ + |\pi_{s,t}(b) - \pi_{s,t}(j)|_\circ$. This means that if for some $j \in S$

$$Z_j^s = \left| \left\{ i \in S_t \setminus \{j\} : |\pi_{s,t}(i) - \pi_{s,t}(j)|_\circ < \frac{n}{B_t} \cdot R_t \right\} \right| > 1,$$

[3] Note that the assumption that no element of S_t is λ-crowded with respect to S_t for $\lambda = R_t^{-3}$ is used twice in the proof: first to show that $Z_j^s \in \{0, 1\}$, since no two elements of S_t can hash too close to each other by the choice of λ, and then later to upper bound the number of neighbors in S_t an element can have. The specific choice of $\lambda = R_t^{-3}$ is only used here, however: for the other application $\lambda = O(1)$ would have been sufficient.

we have $|\pi_{s,t}(a) - \pi_{s,t}(b)|_\circ \le |\pi_{s,t}(a) - \pi_{s,t}(j)|_\circ + |\pi_{s,t}(b) - \pi_{s,t}(j)|_\circ < \frac{n}{B_t} \cdot 2R_t$, a contradiction. Thus, $Z_j^s \in \{0,1\}$ for all $j \in S$, and by property **(3)** of an isolating partition we now conclude that

$$\sum_{s=1}^{R_t} Z_j^s \le R_t^{1-\delta} \tag{14}$$

for all $j \in S$.

Bounding X_j^s. We now turn to bounding X_j^s (i.e. second term on the rhs of (13)). Let

$$N(j,q) := \{i \in S_t \setminus \{j\} \text{ s.t. } |\pi_{s,t}(j) - \pi_{s,t}(i)|_\circ \le \frac{n}{B_t}(2^{q+1}-1) \text{ and } |\pi_{s,t}(j) - \pi_{s,t}(i)|_\circ \ge \frac{n}{B_t} \cdot R_t\} \tag{15}$$

for convenience. We have

$$\begin{aligned} X_j^s = \sum_{i \in S_t \setminus \{j\} : |\pi_{s,t}(i) - \pi_{s,t}(j)|_\circ \ge \frac{n}{B_t} R_t} G_{o_i(j)} &\le \sum_{q \ge \log_2 R_t} \sum_{\substack{i \in S \setminus \{j\} \text{ s.t.} \\ |\pi_{s,t}(j) - \pi_{s,t}(i)|_\circ \in \frac{n}{B_t}[2^q, 2^{q+1}-1)}} G_{o_i(j)} \\ &\le \sum_{q \ge \log_2 R_t} |N(j,q)| \cdot \max_{|\pi_{s,t}(j) - \pi_{s,t}(i)|_\circ \ge \frac{n}{B_t} 2^q} G_{o_i(j)}. \end{aligned} \tag{16}$$

We now upper bound both terms in the last line of the equation above.

To bound the second term it suffices to note that for every $q \ge 0$ every i, j with $|\pi_{t,s}(j) - \pi_{t,s}(i)|_\circ \ge \frac{n}{B_t} 2^q$ satisfy $|o_i(j)|_\circ = |\pi_{t,s}(j) - \frac{n}{B_t} \cdot h_{t,s}(i)|_\circ \ge |\pi_{t,s}(j) - \pi_{t,s}(i)|_\circ - |\pi_{t,s}(i) - \frac{n}{B_t} h_{t,s}(i)|_\circ \ge \frac{n}{B_t} \cdot (2^q - 1)$. Using this bound together with Definition 2.3, **(3)**, and assumption that the filter G is at least 6-sharp we have for $q \ge \log_2 R_t \ge 1$

$$\max_{|\pi(j) - \pi(i)|_\circ \ge \frac{n}{B_t} \cdot 2^q} G_{o_i(j)} \le \left(\frac{1}{4(2^q - 1)}\right)^5 \le (2^q)^{-5} \le 2^{-5q}. \tag{17}$$

Note that we also used the assumption that $q \ge \log_2 R_t \ge 1$ to lower bound $4(2^q - 1)$ by 2^q.

We now upper bound the first term on the last line of (16), i.e. the size of $N(j,q)$ for every $j \in S$. Let $i^* := \text{argmin}_{i \in S_t} |\pi_{s,t}(j) - \pi_{s,t}(i)|_\circ$ denote a point in S_t that is mapped closest to j. Let $L^* := |\pi_{s,t}(j) - \pi_{s,t}(i^*)|_\circ$ denote the distance to this point, and let q^* be the smallest such that $(n/B_t)2^{q^*} \ge L^*$. By triangle inequality we have for all $i \in [n]$

$$|\pi_{t,s}(i) - \pi_{t,s}(i^*)|_\circ \le |\pi_{t,s}(j) - \pi_{t,s}(i)|_\circ + L^*, \tag{18}$$

allowing us to upper bound the size of $N(j,q)$ (points not too far from j) using the fact that S_t is not crowded (property **(2)** of an isolating partition; see Definition 3.7). Specifically, for every $q \ge 0$ we have, combining (18) and (15),

$$\begin{aligned} N(j,q) &\subseteq \{i \in S_t \setminus \{j\} \text{ s.t. } |\pi_{s,t}(i) - \pi_{s,t}(j)|_\circ \le \frac{n}{B_t}(2^{q+1} - 1)\} && \text{(by (15))} \\ &\subseteq \{i \in S_t \setminus \{j\} \text{ s.t. } |\pi_{s,t}(i) - \pi_{s,t}(i^*)|_\circ \le \frac{n}{B_t}(2^{q+1} + 2^{q^*} - 1)\} && \text{(by (18))} \end{aligned}$$

By property **(2)** of an isolating partition (see Definition 3.7) we have for any $q \ge 0$ the number of $i \in S_t$ such that $|\pi_{s,t}(i) - \pi_{s,t}(i^*)|_\circ \le \frac{n}{B_t}(2^{q+1} + 2^{q^*} - 1)$ is bounded by $R_t^{-3}(2^{q+1} + 2^{q^*} - 1)^2 + 1$, where the $+1$ accounts for i^* itself. Since we will only use the bound for $q \ge \log_2 R_t$, the R_t^{-3} factor in first term will not be consequential[4],

[4] Note that this is the second time we are using the assumption that no element of S_t is λ-crowded with respect to S_t, but in this case the choice of $\lambda = R_t^{-3}$ is not important, any constant, even $\lambda > 1$, would have sufficed to this particular application.

and we hence use the simpler form $R_t^{-3}(2^{q+1}+2^{q^*}-1)^2+1 \le (2^{q+1}+2^{q^*}-1)^2+1 \le 2(2^{q+1}+2^{q^*}-1)^2$, where we used the fact that $q \ge \log_2 R_t \ge 0$. We thus have for all $q \ge 0$

$$|N(j,q)| \le \left\{ \begin{array}{cc} 0 & \text{if } q < q^* \\ 2(2^{q+1}+2^{q^*})^2 & \text{o.w.} \end{array} \right. \tag{19}$$

Substituting (17) and (19) into (16), we get

$$\begin{aligned} X_j^s &= \sum_{i \in S_t \setminus \{j\} : |\pi_{s,t}(i)-\pi_{s,t}(j)|_\circ \ge \frac{n}{B_t} R_t} G_{o_i(j)} \le \sum_{q \ge \log_2 R_t} \max_{|\pi_{s,t}(j)-\pi_{s,t}(i)|_\circ \ge \frac{n}{B_t} 2^q} G_{o_i(j)} \cdot |N(j,q)| \\ &\le \sum_{\substack{q \ge \log_2 R_t \\ q \ge q^*}} 2^{-5q} \cdot (2 \cdot (2^{q+1}+2^{q^*})^2) \le \sum_{\substack{q \ge \log_2 R_t \\ q \ge q^*}} 2^{-5q} \cdot (32 \cdot 2^{2q}) \\ &\le 32 \cdot \sum_{\substack{q \ge \log_2 R_t \\ q \ge q^*}} 2^{-3q} \quad \text{(summing the geometric sum)} \\ &\le 64 \cdot R_t^{-3} \le R_t^{-2} \end{aligned}$$

as long as C_1 is larger than 64 (since $R_t = C_1 \cdot 2^t \ge C_1$ by assumption **p1** of the lemma). Substituting this bound into (13), we get $D_j^s \le Z_j^s + X_j^s \le Z_j^s + R_t^{-2}$, which means that

$$\sum_{s=1}^{R_t} D_j^s \le R_t^{1-\delta} + R_t^{-1}. \tag{20}$$

To complete the argument, recall that by (8) one has $e_i^{head}(\{H_{t,s}\}, x') = \text{quant}^{1/5}_{s \in [1:R_t]} e_i^{head}(H_{t,s}, x')$. This means that for each $i \in S_t$ there exist at least $(1/5) r_{max}$ values of $s \in [1 : R_t]$ such that $e_i^{head}(H_{t,s}, x') > e_i^{head}$, and hence

$$||e_{S_t}^{head}(\{H_{t,s}\}, x')||_1 \le \frac{1}{(1/5)R_t} \sum_{s=1}^{R_t} ||e_{S_t}^{head}(H_{t,s}, x')||_1. \tag{21}$$

Substituting the bound from (21) into (12), we get

$$\begin{aligned} ||e_{S_t}^{head}(\{H_{t,s}\}, x')||_1 \le \frac{1}{(1/5)R_t} \sum_{s=1}^{R_t} ||e_{S_t}^{head}(H_{t,s}, x')||_1 &\le \frac{2}{(1/5)R_t} \sum_{s=1}^{R_t} \sum_{j \in S} |x'_j| \cdot D_j^s \\ &\le \sum_{j \in S} |x'_j| \cdot \left(\frac{2}{(1/5)R_t} \sum_{s=1}^{R_t} D_j^s \right). \end{aligned}$$

Substituting the above into (21), we get

$$\begin{aligned} ||e_{S_t}^{head}(\{H_{t,s}\}, x')||_1 \le \sum_{j \in S} |x'_j| \cdot \left(\frac{2}{(1/5)R_t} \sum_{s=1}^{R_t} D_j^s \right) &\le \sum_{j \in S} |x'_j| \cdot \left(\frac{2}{(1/5)R_t} (R_t^{1-\delta} + R_t^{-1}) \right) \\ &\le \sum_{j \in S} |x'_j| \cdot \left(\frac{20}{R_t} R_t^{1-\delta} \right) \quad \text{(since } R_t^{-1} \le R_t^{1-\delta}) \\ &\le 20 R_t^{-\delta} ||x'_S||_1. \end{aligned} \tag{22}$$

Substituting the bound from (22) into (21) we get $||e_{S_t}^{head}(\{H_{t,s}\}, x')||_1 \le 20 \cdot R_t^{-\delta} ||x'_S||_1$, as required. □

4 Sample efficient estimation

In this section we state our sample optimal estimation algorithm (Algorithm 2) and provide its analysis.

4.1 Algorithm and overview of analysis

Our algorithm (Algorithm 2) contains three major components: it starts by taking measurements m of the signal x (accessing the signal in Fourier domain, i.e. accessing $\widehat{x}$), then uses these measurements to perform a sequence of ℓ_1 norm reduction steps, reducing ℓ_1 norm of the residual signal on the target set S of coefficients to about the noise level. Finally, a simple cleanup procedure is run to convert the ℓ_1 norm bounds on the residual to ℓ_2/ℓ_2 guarantees of (1). In this section we will use the functions e^{head}, e^{tail} (see Section 2) defined with respect to the set S.

Measuring $\widehat{x}$. All measurements that the algorithm takes are taken in lines 6-12, and then line 31. The measurements in lines 6-12 are taken over $T = \frac{1}{1-\delta}\log_2\log(k+1) + O(1)$ rounds for small constant $\delta \in (0, 1/2)$, where in round $t = 1, \ldots, T$ we are hashing the signal into $B_t \approx k/R_t^2$ buckets, where R_t grows exponentially with t. For each t we perform R_t independent hashing experiments of this type. This matches the setup of Lemma 3.1, which is our main analysis tool (see proof of Lemma 4.1).

ℓ_1 **norm reduction loop.** Once the samples have been taken, Algorithm 2 proceeds to the ℓ_1 norm reduction loop (lines 16-22). The objective of this loop is to reduce the ℓ_1 norm of the residual signal on the target set S of coefficients that we would like to estimate to about the noise level, namely to $O(||x_{[n]\setminus S}||_2\sqrt{k})$. The formal guarantees are provided by

Lemma 4.1. *For every $\delta \in (0, 1/2)$, if C_1, C_2 (parameters in Algorithm 2) are sufficiently large constants, then the following conditions hold.*

For every $x \in \mathbb{C}^n$, every integer $k \geq 1$, every $S \subseteq [n]$, $|S| = k$, if $||x||_\infty \leq R^* \cdot ||x_{[n]\setminus S}||_2/\sqrt{k}, R^* = n^{O(1)}$, the vector $\tilde{\chi}$ computed in line 23 of an invocation of ESTIMATE$(\hat{x}, S, k, \epsilon, R^*)$ *(Algorithm 2) satisfies*

$$||(x - \tilde{\chi})_S||_1 \leq O(||x_{[n]\setminus S}||_2 \cdot \sqrt{k})$$

conditioned on an event $\mathcal{E}_{maj}$ that occurs with probability at least $1 - 2/25$.

Cleanup phase and final result. Once the ℓ_1 norm of the residual on S has been reduced to $O(||x_{[n]\setminus S}||_2\sqrt{k})$, we run the ESTIMATEVALUES procedure once to convert ℓ_1 norm bounds on the residual into ℓ_2/ℓ_2 guarantees (1). This results in a proof of Theorem 1.1, restated below for convenience of the reader. The theorem establishes correctness of Algorithm 2, as well as its runtime and sample complexity bounds:

Theorem 1.1 (Restated) *For every $\epsilon \in (1/n, 1), \delta \in (0, 1/2)$, $x \in \mathbb{C}^n$ and every integer $k \geq 1$, any $S \subseteq [n]$, $|S| = k$, if $||x||_\infty \leq R^* \cdot ||x_{[n]\setminus S}||_2/\sqrt{k}, R^* = n^{O(1)}$, an invocation of* ESTIMATE$(\hat{x}, S, k, \epsilon, R^*)$ *(Algorithm 2) returns $\chi^* \in \mathbb{C}^n$ such that*

$$||(x - \chi^*)_S||_2^2 \leq \epsilon \cdot ||x_{[n]\setminus S}||_2^2$$

using $O_\delta(\frac{1}{\epsilon}k)$ samples and $O_\delta(\frac{1}{\epsilon}k\log^{3+\delta} n)$ time with at least $4/5$ *success probability.*

4.2 Proof of Lemma 4.1

We now give

Proof of Lemma 4.1: Recall that in this section we use the quantities e^{head} and e^{tail} defined with respect to the set S. We will also use an isolating partition of S, denoted by $S = S_1 \cup S_2 \cup \ldots \cup S_T$. We argue the existence of such a partition with high probability later.

We prove by induction on the pair (r, t) that conditional on a high probability success event $\mathcal{E}_{maj}$ (defined below) the residual signals $x - \chi^{(r,t)}$ are *majorized* on S (in the sense of Definition 2.12) by a fixed sequence

Algorithm 2 ESTIMATE($\hat{x}, S, k, \epsilon, R^*$)

```
 1: procedure ESTIMATE($\hat{x}, S, k, \epsilon, R^*$)                         ▷ List $S$ of size $k$
 2:     $T \leftarrow \frac{1}{1-\delta}\log_2 \log(k+1) + O(1)$ for a small constant $\delta \in (0, 1/2)$
 3:     $R_t \leftarrow C_1 \cdot 2^t$ for $t \in [1:T]$              ▷ $C_1 > 0$ sufficiently large absolute constant
 4:     $B_t \leftarrow C_2 \cdot k/R_t^2$ for $t \in [1:T]$           ▷ $C_2 > 0$ sufficiently large absolute constant
 5:     $G_t \leftarrow$ filter with $B_t$ buckets and sharpness $F = 8$.
 6:     for $t = 1$ to $T$ do                                              ▷ Take samples
 7:         for $s = 1$ to $R_t$ do
 8:             Choose $\sigma \in \mathcal{M}_{odd}$, $q \in [n]$ u.a.r., let $\pi_{t,s} \leftarrow (\sigma, q)$, $H_{t,s} := (\pi_{t,s}, B_t, F)$
 9:             Let $a_{t,s} \leftarrow$ an element of $[n]$ u.a.r.
10:             $m(x, H_{t,s}, a_{t,s}) \leftarrow$ HASHTOBINS($\hat{x}, 0, (H_{t,s}, a_{t,s})$)
11:         end for
12:     end for
13:     Explicitly construct a $\delta$-isolating partition $S = S_1 \cup S_2 \ldots \cup S_T$     ▷ As per Lemma 3.9
14:     $\chi^{(0,0)} \leftarrow 0$
15:     $r' \leftarrow 0, t' \leftarrow 0$
16:     for $r = 0, 1, \ldots, C \log_4 R^*$ do                              ▷ For any constant $C \geq 1$
17:         for $t = 1$ to $T$ do
18:             $\chi' \leftarrow$ ESTIMATEVALUES($\chi^{(r',t')}, S_t, \{(H_{t,s}, a_{t,s}, m(x, H_{t,s}, a_{t,s}))\}_{s=1}^{R_t}$)
19:             $\chi^{(r,t)} \leftarrow \chi^{(r',t')} + \chi'$                    ▷ $(r', t')$ are the previous indices
20:             $r' \leftarrow r, t' \leftarrow t$
21:         end for
22:     end for
23:     $\tilde{\chi} \leftarrow \chi^{(C \log_4 R^*, T)}$                         ▷ $\tilde{\chi}$ is the final residual computed by the loop
24:     $B \leftarrow C_2 \cdot k/\epsilon$
25:     $G \leftarrow$ filter with $B$ buckets and sharpness $F = 8$.
26:     $r_{max} \leftarrow O(1)$                                      ▷ A sufficiently large absolute constant
27:     for $r = 1$ to $O(1)$ do
28:         Choose $\sigma_r \in \mathcal{M}_{odd}$, $q_r, a_r \in [n]$ u.a.r., let $\pi_r \leftarrow (\sigma_r, q_r)$, $H_r := (\pi_r, B, F)$
29:         $m(x, H_r, a_r) \leftarrow$ HASHTOBINS($\hat{x}, \tilde{\chi}, H_r, a_r$)
30:     end for
31:     $\chi'' \leftarrow$ ESTIMATEVALUES($\tilde{\chi}, S, \{(H_r, a_r, m(x, H_r, a_r))\}_{r=1}^{r_{max}}$)
32:     $\chi^* \leftarrow \tilde{\chi} + \chi''$
33:     return $\chi^*$
34: end procedure
```

$y^{(r,t)}$ whose ℓ_1 norm converges to $O(||x_{[n]\setminus S}||_2 \cdot \sqrt{k})$ after $O(\log R^*)$ iterations. Since we only update elements in S, this gives the result. We now give the details of the argument. In what follows we let $\mu^2 := ||x_{[n]\setminus S}||_2^2/k$ for convenience. Note, however, that Algorithm 2 is oblivious to the value of μ: we only need an upper bound on $\log R^*$.

We start by defining the majorizing sequence $y^{(r,t)}$. We first let $y_i^{(0,0)} = R^*\mu$ for all $i \in S$ and $y_i^{(0,0)} = x_i$ otherwise. Note that $y^{(0,0)}$ trivially majorizes x as $||x||_\infty \leq R^* \cdot \mu$ by assumption of the lemma. The construction of $y^{(r,t)}$ proceeds by induction on (t, r). Given $y^{(r',t')}$, as per Algorithm 2 the next signal to be defined is $y^{(r,t)}$ with $(r, t) = (r', t' + 1)$ if $t < T$ and $(r, t) = (r' + 1, 1)$ otherwise (as per lines 15-22 of Algorithm 2). We now define the signal $y^{(r,t)}$ by letting for each $i \in [n]$ (recall that S_t is the t-th set in an isolating partition

$S = S_1 \cup S_2 \cup \ldots \cup S_T$)

$$y_i^{(r,t)} := \begin{cases} 20e_i^{head}(\{H_{t,s}\}_{s\in[1:R_t]}, y^{(r',t')}) + 20e_i^{tail}(\{H_{t,s}, a_{t,s}\}_{s\in[1:R_t]}, x) + n^{-\Omega(c)} & \text{if } i \in S_t \\ y_i^{(r',t')} & \text{o.w.} \end{cases} \tag{23}$$

Here $n^{-\Omega(c)}$ corresponds to the (negligible) error term due to polynomial precision of our computations. Note that there are two contributions to $y^{(r,t)}$: one coming from the previous signal in the majorizing sequence, namely $y^{(r',t')}$, and the other coming from the tail of the signal x.

We now prove by induction on (t, r) that the loop in our estimation primitive reduces the ℓ_1 norm of the residual to $O(\mu \cdot k)$ (recall that $\mu^2 = ||x_{[n]\setminus S}||_2^2/k$). Specifically, we prove that there exists an event $\mathcal{E}_{maj}$ with $\mathbf{Pr}_{\{\{H_{t,s}\}_{s\in[1:R_t]}\}_{t=1}^T}[\mathcal{E}_{maj}] \geq 1 - 2/25$ such that conditioned on $\mathcal{E}_{maj}$ the set S admits an isolating partition $S = S_1 \cup S_2 \cup \ldots \cup S_T$ with respect to $\{\{H_{t,s}\}\}$, and for every $(r, t) \in ([0 : +\infty) \times [1 : T]) \cup \{(0, 0)\}$

(A) for all $q \in [1 : t]$ one has $||y_{S_q}^{(r,t)}||_1 \leq (R^* \cdot (1/4)^{r+1}\mu + 2\mu) \cdot k \cdot (R_0/R_{q-1})^\delta$;

(B) for all $q \in [t + 1 : T]$ one has $||y_{S_q}^{(r,t)}||_1 \leq (R^* \cdot (1/4)^r\mu + 2\mu) \cdot k \cdot (R_0/R_{q-1})^\delta$;

(C) $||y_S^{(r,t)}||_1 \leq (2/\delta) \cdot (R^*(1/4)^r\mu + 2\mu) \cdot k$;

(D) $(x - \chi^{(r,t)}) \prec_S y^{(r,t)}$ and $\operatorname{supp} \chi^{(r,t)} \subseteq S$.

First, note that the set S admits an isolating partition with respect to the hash functions $\{\{H_{t,s}\}\}$ with probability at least $1 - 1/25$ by Lemma 3.8. Denote the success event by $\mathcal{E}_{partition}$. We condition on this event in what follows. We give the inductive argument, and finally define the event $\mathcal{E}_{maj}$ as the intersection of $\mathcal{E}_{partition}$ with several other high probability success events.

The **base** is provided by $r = 0$ and $t = 0$. Indeed, by property **(1)** of an isolating partition (see Definition 3.7) we have for any $q \in [1 : T]$

$$||y||_{S_q} \leq R^*\mu \cdot |S_q| \leq R^*\mu \cdot k \cdot \frac{R_0}{R_{q-1}} 2^{-2^{(1-\delta)(q-1)}+1} \leq R^*\mu \cdot k \cdot (R_0/R_{q-1})^\delta$$

since $2^{-2^{(1-\delta)(q-1)}+1} \leq 1$ for all $q \geq 1$ and $\frac{R_0}{R_{q-1}} \leq (R_0/R_{q-1})^\delta$ (as $\delta < 1$ by assumption of the lemma).

We now prove the **inductive step**. Let $(r', t') := (r, t - 1)$ if $t > 1$, else let $(r', t') := (r - 1, T)$ if $r > 1$ and $t = 1$, and $(r', t') = (0, 0)$ otherwise. Note that (t', r') is the element preceding $y^{(r,t)}$ in the majorizing sequence (as per lines 15-22 of Algorithm 2).

Proving (C). We start with an upper bound on the ℓ_1 norm of $y^{(r',t')}$, i.e. prove **(C)**. Using the inductive hypothesis **(A)** and **(B)** for (t', r'), we get

$$\begin{aligned} ||y_S^{(r',t')}||_1 &\leq \sum_{q=1}^{t'} (R^*(1/4)^{r'+1}\mu + 2\mu) \cdot k \cdot (R_0/R_{q-1})^\delta + \sum_{q=t'+1}^{\infty} (R^*(1/4)^r\mu + 2\mu) \cdot k \cdot (R_0/R_{q-1})^\delta \\ &\leq (R^*(1/4)^{r'}\mu + 2\mu) \cdot k \cdot \sum_{q=1}^{\infty} (R_0/R_{q-1})^\delta \\ &= (R^*(1/4)^{r'}\mu + 2\mu) \cdot k \cdot \sum_{q=1}^{\infty} 2^{-(q-1)\delta} \qquad (\text{since } R_t = C_1 2^t \text{ by } \mathbf{p1}) \\ &\leq \frac{1}{2^\delta - 1} \cdot (R^* k(1/4)^{r'}\mu + 2\mu) \\ &\leq \frac{1}{e^{\delta \ln 2} - 1} \cdot (R^* k(1/4)^{r'}\mu + 2\mu) \\ &\leq (2/\delta) \cdot (R^* k(1/4)^{r'}\mu + 2\mu) \qquad (\text{since } e^x - 1 \geq x, \text{ and } \ln 2 > 1/2) \end{aligned} \tag{24}$$

This establishes **(C)**, and we now turn to **(A)** and **(B)**. By definition the signal $y^{(r,t)}$ is obtained from $y^{(r',t')}$ by modifying the latter on S_t. We need to bound the error introduced by head and tail elements of $y^{(r',t')}$ to $y^{(r,t)}_{S_t}$ (see (23)). We now bound both terms.

Proving (A) and (B): analyzing contribution from the tail e^{tail}. By Lemma B.2, **(2)**, one has for every $i \in [n]$, $t = 1, \dots, T$ and $s = 1, \dots, R_t$

$$\mathbf{E}_{H_{t,s},a_{t,s}}\left[(e_i^{tail}(H_{t,s}, a_{t,s}, x))^2\right] \le \nu^2$$

for some $\nu > 0$ such that $\nu^2 = O(||x_{[n]\setminus S}||_2^2/B_t)$. By Jensen's inequality we thus have

$$\mathbf{E}_{H_{t,s},a_{t,s}}\left[e_i^{tail}(H_{t,s}, a_{t,s}, x)\right] \le \nu.$$

To upper bound $\mathbf{E}_{\{H_{t,s},a_{t,s}\}}\left[||e^{tail}_{S_t}(\{H_{t,s}, a_{t,s}\}_{s\in[1:R_t]}, x)||_1\right]$, we note that by conditioning on $\mathcal{E}_{partition}$ we have $|S_t| \le k\frac{R_0}{R_{t-1}}2^{-2^{(1-\delta)(t-1)}+1}$. Letting $U := k\frac{R_0}{R_{t-1}}2^{-2^{(1-\delta)(t-1)}+1}$ to simplify notation, we get that

$$\mathbf{E}_{\{H_{t,s},a_{t,s}\}}\left[||e^{tail}_{S_t}(\{H_{t,s}, a_{t,s}\}_{s\in[1:R_t]}, x)||_1\right] \le \mathbf{E}_{\{H_{t,s},a_{t,s}\}}\left[\max_{Q\subseteq S, |Q|\le U}||e^{tail}_{Q}(\{H_{t,s}, a_{t,s}\}_{s\in[1:R_t]}, x)||_1\right] \quad (25)$$

We now recall that by (10) one has

$$e_i^{tail}(\{H_{t,s}, a_{t,s}\}, x) := \text{quant}^{1/5}_{s=1,\dots,R_t} e_i^{tail}(H_{t,s}, a_{t,s}, x),$$

and apply Lemma B.6 with $\gamma = 1/5$, $m = |S|, n = R_t$ and

$$X_i^s = e_i^{tail}(H_{t,s}, a_{t,s}, x) \quad \text{for } i \in S \text{ and } s = 1, \dots, R_t,$$

so that $\mathbf{E}_{H_{t,s},a_{t,s}}[X_i^s] \le \nu$ for each $i \in S$, $s = 1, \dots, R_t$. Note that $Y_i := \text{quant}^{1/5}_{s=1,\dots,R_t} X_i^s = e_i^{tail}(\{H_{t,s}, a_{t,s}\}, x)$ is exactly the quantity that we are interested in. We thus have by Lemma B.6

$$\begin{aligned}\mathbf{E}_{\{H_{t,s},a_{t,s}\}}\left[\max_{Q\subseteq S, |Q|\le U}||e^{tail}_{Q}(\{H_{t,s}, a_{t,s}\}_{s\in[1:R_t]}, x)||_1\right] &= \mathbf{E}_{\{H_{t,s},a_{t,s}\}}\left[\max_{Q\subseteq S, |Q|\le U}\sum_{i\in Q} Y_i\right] \\ &\le U \cdot (20e\nu) \cdot (|S|/U)^{10/R_t}\end{aligned} \quad (26)$$

Since $R_{t'} = C_1 2^{t'}$ for every t', $|S| = |S_0| \le k$ and $U = k\frac{R_0}{R_{t-1}}2^{-2^{(1-\delta)(t-1)}+1} = k2^{-2^{(1-\delta)(t-1)}+1-(t-1)}$, we have

$$(|S|/U)^{10/R_t} = 2^{10(2^{(1-\delta)(t-1)}-1+(t-1))/(C_1 2^t)} \le 2^{10(1+(t-1)/2^t)/C_1} \le 2^{20/C_1} \le 2$$

for all $t \ge 1$ as long as $C_1 > 20$. Substituting the above into (26), we get

$$\mathbf{E}_{\{H_{t,s},a_{t,s}\}}\left[\max_{Q\subseteq S, |Q|\le U}||e^{tail}_{Q}(\{H_{t,s}, a_{t,s}\}_{s\in[1:R_t]}, x)||_1\right] \le (40e) \cdot U \cdot \nu,$$

and thus by (25)

$$\begin{aligned}\mathbf{E}_{\{H_{t,s},a_{t,s}\}}\left[||e^{tail}_{S_t}(\{H_{t,s}, a_{t,s}\}_{s\in[1:R_t]}, x)||_1\right] &= O(U \cdot \nu) \\ &= O(2k\frac{R_0}{R_{t-1}}R_t 2^{-2^{(1-\delta)(t-1)}+1} \cdot ||x_{[n]\setminus S}||_2/\sqrt{C_2 k}) \\ &= \mu k \frac{1}{R_{t-1}^2} \cdot O\left(R_t^2 R_0 2^{-2^{(1-\delta)(t-1)}+1}/\sqrt{C_2}\right) \\ &= \mu k \frac{1}{R_{t-1}^2} \cdot \xi_t, \quad \xi_t = O\left(R_t^2 R_0 2^{-2^{(1-\delta)(t-1)}+1}/\sqrt{C_2}\right).\end{aligned}$$

Since $R_t = C_1 2^t$ increases only exponentially, whereas the second multiplier decreases at a doubly exponential rate, as long as C_2 is larger than a constant, we get that $\xi_t \le 1/10000$ for all $t \ge 1$ (formally, this follows by Claim A.2). By Markov's inequality, for each $t \ge 1$ we have

$$||e^{tail}_{S_t}(\{H_{t,s}, a_{t,s}\}_{s\in[1:R_t]}, x)||_1 \le \frac{1}{200}\mu k \frac{1}{R_{t-1}}$$

with probability at least $1 - 1/(50R_{t-1}) \ge 1 - 1/(50 \cdot 2^{t-1})$. Thus, by a union bound over all $t \ge 1$ we have with probability at least $1 - 1/25$

$$||e^{tail}_{S_t}(\{H_{t,s}, a_{t,s}\}_{s\in[1:R_t]}, x)||_1 \le \frac{1}{200}\mu k \frac{1}{R_{t-1}}. \tag{27}$$

Denote the success event above by $\mathcal{E}_{small-noise}$.

Proving (A) and (B): analyzing contribution from the head e^{head}. By Lemma 3.1 we have

$$||e^{head}_{S_t}(\{H_{t,s}\}_{s\in[1:R_t]}, y^{(r',t')})||_1 \le 20R_t^{-\delta}||y^{(r',t')}_S||_1. \tag{28}$$

We now define the event $\mathcal{E}_{maj}$ by letting $\mathcal{E}_{maj} := \mathcal{E}_{small-noise} \cap \mathcal{E}_{partition}$. Note that $\mathbf{Pr}[\mathcal{E}_{maj}] \ge 1 - 2/25$ by a union bound, as required. We condition on $\mathcal{E}_{maj}$ for the rest of the proof.

Proving (A) and (B): putting it together. We now use the bounds above to prove the result. By definition of y above we have

$$y^{(r,t)}_i := 20e^{head}_i(\{H_{t,s}\}_{s\in[1:R_t]}, y^{(r',t')}) + 20e^{tail}_i(\{H_{t,s}, a_{t,s}\}_{s\in[1:R_t]}, x) + n^{-\Omega(c)},$$

so

$$\begin{aligned}
||y^{(r,t)}_{S_t}||_1 &\le \sum_{i\in S_t}\Big(20e^{head}_i(\{H_{t,s}\}_{s\in[1:R_t]}, y^{(r',t')}) + 20e^{tail}_i(\{H_{t,s}, a_{t,s}\}_{s\in[1:R_t]}, x) + n^{-\Omega(c)}\Big)\\
&\le 400 \cdot R_t^{-\delta} \cdot ||y^{(r',t')}_S||_1 + 20||e^{tail}_{S_t}(\{H_{t,s}, a_{t,s}\}, x)||_1 + n^{-\Omega(c)}
\end{aligned}$$

We now substitute (24) together with (27) and (28) into the last line above, and obtain

$$\begin{aligned}
||y^{(r,t)}_{S_t}||_1 &\le 400 \cdot R_t^{-\delta}(2/\delta) \cdot (R^*(1/4)^{r'}\mu + 2\mu)k + \frac{1}{10}\mu k/R_{t-1} + n^{-\Omega(c)}\\
&\le 400 \cdot (2/\delta) \cdot R_0^{-\delta} \cdot (R^*(1/4)^{r'}\mu + 2\mu)k \cdot (R_0/R_{t-1})^{\delta} + \frac{1}{10}\mu k \cdot (R_0/R_{t-1})^{\delta} + n^{-\Omega(c)} \qquad \text{(since } \delta \in (0,1))\\
&\le \left[400 \cdot (2/\delta) \cdot C_1^{-\delta} + \frac{1}{10}\right] \cdot (R^*(1/4)^{r'}\mu + 2\mu)k \cdot (R_0/R_{t-1})^{\delta} + n^{-\Omega(c)}
\end{aligned}$$

where we upper bounded $\frac{1}{10}\mu k/R_{t-1}$ by $\frac{1}{10}\mu k \cdot (R_0/R_t)^{\delta}$ (which is justified since $R_0 \ge 1$ and $\delta \in (0,1)$) and used the bound $R_t^{-\delta} = R_0^{-\delta} \cdot (R_0/R_t)^{\delta}$.

We now conclude that as long as $C_1 \ge (40000/\delta)^{1/\delta}$, we have

$$\begin{aligned}
||y^{(r,t)}_{S_t}||_1 &\le \left[400 \cdot (2/\delta) \cdot C_1^{-\delta} + \frac{1}{10}\right] \cdot (R^*(1/4)^{r'}\mu + 2\mu)k \cdot (R_0/R_{t-1})^{\delta} + n^{-\Omega(c)}\\
&\le (R^*(1/4)^{r'+1}\mu + 2\mu)k \cdot (R_0/R_{t-1})^{\delta} + n^{-\Omega(c)}.
\end{aligned}$$

This completes the proof of the inductive step for **(A)** and **(B)**. It remains to prove **(D)**.

Proving (D). Our main tool in arguing **(D)** is Lemma 2.11, which we invoke with the set S. By that lemma we have for every $i \in S$

$$|x_i - \chi^{(r',t')}_i - \chi'_i| \le 2\text{quant}^{1/5}_s e^{head}_i(H_{t,s}, x - \chi^{(r',t')}) + 2\text{quant}^{1/5} e^{tail}_i(H_{t,s}, a_{t,s}, x) + n^{-\Omega(c)},$$

since supp $\chi^{(r',t')} \subseteq S$ by the inductive hypothesis. This implies by definition of $y^{(r,t)}$ that for every $i \in S_t$

$$\begin{aligned}|x_i - \chi_i^{(r',t')} - \chi'_i| &\leq 2\text{quant}_r^{1/5} e_i^{head}(H_{t,s}, x - \chi^{(r',t')}) + 2\text{quant}_s^{1/5} e_i^{tail}(H_{t,s}, a_{t,s}, x) + n^{-\Omega(c)} \\ &\leq 20\text{quant}_s^{1/5} e_i^{head}(H_{t,s}, x - \chi^{(r',t')}) + 20\text{quant}_s^{1/5} e_i^{tail}(H_{t,s}, a_{t,s}, x) + n^{-\Omega(c)}.\end{aligned} \tag{29}$$

By part **(D)** of the inductive hypothesis we have $x - \chi^{(r',t')} \prec_S y^{(r',t')}$, and thus by Lemma 2.13 together with (29) for every $i \in S_t$

$$\begin{aligned}|x_i - \chi_i^{(r',t')} - \chi'_i| &\leq 20\text{quant}_s^{1/5} e_i^{head}(H_{t,s}, x - \chi^{(r',t')}) + 20\text{quant}_s^{1/5} e_i^{tail}(H_{t,s}, a_{t,s}, x) + n^{-\Omega(c)} \\ &\leq 20\text{quant}_s^{1/5} e_i^{head}(H_{t,s}, y^{(r',t')}) + 20\text{quant}_s^{1/5} e_i^{tail}(H_{t,s}, a_{t,s}, x) + n^{-\Omega(c)} \\ &= y_i^{(r,t)}.\end{aligned}$$

We thus have for every $i \in S_t$

$$|x_i - \chi_i^{(r',t')} - \chi'_i| \leq y_i^{(r,t)}.$$

Since $y_i^{(r,t)} = y_i^{(r',t')}$ for $i \notin S_t$, $\chi_i^{(r,t)} = \chi_i^{(r',t')}$ for $i \notin S_t$ and $x - \chi^{(r',t')} \prec_S y^{(r',t')}$ by the inductive hypothesis, we get

$$x - \chi^{(r',t')} - \chi' \prec_S y^{(r,t)}$$

as required. Since we only update elements of S, we have supp $\chi^{(r,t)} \subseteq$ supp $\chi^{(r',t')} \cup$ supp $\chi' \subseteq S$. This completes the proof of **(D)**, and the proof of the induction.

To obtain the final result of the lemma, we note that by part **(C)** of the inductive claim for every $r \geq C \log_4 R^*$ (for any $C \geq 1$) one has

$$||y_S^{(r,T)}||_1 \leq (2/\delta) \cdot (R^*(1/4)^r \mu + 2\mu) \cdot k \leq (6/\delta) \cdot \mu k.$$

Now recall that by line 23 of Algorithm 2 we have $\tilde{\chi} = \chi^{(C \log_4 R^*, T)}$, which implies by part **(D)** of the inductive claim, since $(x - \chi^{(C \log_4 R^*, T)}) \prec_S y^{(C \log_4 R^*, T)}$, that

$$||(x - \tilde{\chi})_S||_1 = ||(x - \chi^{(C \log_4 R^*, T)})_S||_1 \leq ||y_S^{(C \log_4 R^*, T)}||_1 \leq (6/\delta) \cdot \mu k = O(\mu k),$$

as required.

□

4.3 Proof of Theorem 1.1

We now give

Proof of Theorem 1.1: Recall that in this section we use the quantities e^{head} and e^{tail} defined with respect to the set S. By Lemma 4.1 we have that conditioned on a high probability event $\mathcal{E}_{maj}$ (which occurs with probability at least $1 - 2/25$) the vector $\tilde{\chi}$ computed in line 23 satisfies

$$||(x - \tilde{\chi})_S||_1 = O(||x_{[n] \setminus S}||_2 \sqrt{k}) \tag{30}$$

To complete the proof, we show that the output χ'' of the invocation of ESTIMATEVALUES in line 31, when added to $\tilde{\chi}$, yields guarantee claimed by the lemma. First, by Lemma 2.11 with S one has for each $i \in S$

$$|\chi''_i - (x - \tilde{\chi})_i| \leq 2 \cdot \text{quant}_r^{1/5} e_i^{head}(H_r, x - \tilde{\chi}) + 2 \cdot \text{quant}_r^{1/5} e_i^{tail}(H_r, a_r, x) + n^{-\Omega(c)}, \tag{31}$$

since supp $\tilde{\chi} \subseteq S$.

Squaring both sides of (31), using the bound $(a+b)^2 \leq 2a^2 + 2b^2$ and taking expectations over the randomness in measurements taken in lines 27-30, we get

$$\mathbf{E}[|\chi''_i - (x-\tilde{\chi})_i|^2] \leq 8 \cdot \mathbf{E}\left[(\text{quant}_r^{1/5} e_i^{head}(H_r, x-\tilde{\chi})^2\right] + 8 \cdot \mathbf{E}\left[(\text{quant}_r^{1/5} e_i^{tail}(H_r, a_r, x))^2\right] + n^{-\Omega(c)}. \quad (32)$$

We now upper bound the expectation of (32). By Lemma B.5, **(1)** one has, letting $Z^{head} := \text{quant}_r^{1/5} e_i^{head}(H_r, x-\tilde{\chi})$ to simplify notation,

$$\mathbf{E}\left[(Z^{head})^2\right] = O\left(\left(\frac{1}{B}||(x-\tilde{\chi})_S||_1\right)^2\right) = O\left(\left(\frac{1}{C_2 k/\epsilon}||(x-\tilde{\chi})_S||_1\right)^2\right) = O(\epsilon^2 ||x_{[n]\setminus S}||_2^2/(C_2 k)),$$

where we used that by conditioning on $\mathcal{E}_{maj}$ one has $||(x-\tilde{\chi})_S||_1 = O(||x_{[n]\setminus S}||_2 \sqrt{k})$ (by (30)).

By Lemma B.5, **(2)** with S one has, letting $Z^{tail} := \text{quant}_r^{1/5} e_i^{tail}(H_r, a_r, x)$ to simplify notation,

$$\mathbf{E}\left[(Z^{tail})^2\right] = O(||(x-\tilde{\chi})_{[n]\setminus S}||_2^2/B) = O(\epsilon ||x_{[n]\setminus S}||_2^2/(C_2 k)),$$

where we used the fact that $\text{supp}\, \tilde{\chi} \subseteq S$.

Substituting these bounds into (32) and summing over all $i \in S$, we get

$$\mathbf{E}[||(x-\tilde{\chi}-\chi'')_S||^2] \leq O(\epsilon^2 ||x_{[n]\setminus S}||_2^2/C_2) + O(\epsilon ||x_{[n]\setminus S}||_2^2/C_2) \leq (\epsilon/1000)||x_{[n]\setminus S}||_2^2$$

as long as C_2 is sufficiently large.

An application of Markov's inequality then gives $||(x-\tilde{\chi}-\chi'')_S||^2 \leq \epsilon ||x_{[n]\setminus S}||_2^2$ with probability at least $1 - 1/1000$. By a union bound over this failure event and $\bar{\mathcal{E}}_{maj}$, we conclude that the algorithm outputs the correct answer with probability at least $1 - 3/25 \geq 4/5$.

We now upper bound the sample complexity and runtime.

Sample complexity. The sample complexity of lines 6-11 is bounded by $\sum_{t=1}^T \sum_{s=1}^{R_t} O(F \cdot B_t) = \sum_{t=1}^T R_t \cdot O(F \cdot k/R_t^2) = O(k) \cdot \sum_{t=1}^T 1/R_t = O(k)$ by the choice of R_t as geometrically increasing. The sample complexity of lines 27-30 is upper bounded by $O(F \cdot B) = O(k/\epsilon)$ by Lemma 2.11 and choice of $F = O(1)$.

Runtime. The runtime of HashToBins in line 10 of Algorithm 2 is $O(F \cdot B_t \log B_t) = O(B_t \log B_t)$ by Lemma 2.8, the setting of $F = O(1)$ and the fact that the residual signal passed to the call is zero. Since this line is executed for $t = 1, \ldots, T$ and $s = 1, \ldots, R_t$, the total runtime of the loop is

$$\sum_{t=1}^T \sum_{s=1}^{R_t} O(B_t \log B_t) = O\left(\sum_{t=1}^T R_t \cdot (C_2 k/R_t^2) \log(C_2 k)\right) = O(k \log k) \cdot \sum_{t=1}^T 1/R_t = O(k \log k).$$

The runtime for construction of the partition $S_1 \cup S_2 \cup \ldots \cup S_T$ in line 13 is $O((\sum_{t=1}^T R_t)|S| \log |S|) = O(R_T k \log k)$ by Lemma 3.9 and the fact that $\sum_{t=1}^T R_t = O(R_T)$. We now note that since $T = \frac{1}{1-\delta} \log_2 \log(k+1) + O(1)$, then

$$R_T = C_1 2^T = C_1 2^{\frac{1}{1-\delta} \log_2 \log k + O(1)} = O(\log_2^{1/(1-\delta)}(k+1)) = O(\log_2^{1+2\delta}(k+1)), \quad (33)$$

where we used the fact that $1/(1-\delta) \leq 1 + 2\delta$ for $\delta \in (0, 1/2)$. Thus, the runtime for construction of the partition $S_1 \cup S_2 \cup \ldots \cup S_T$ in line 13 is $O(k \log^{2+2\delta} k)$.

By Lemma 2.11 each invocation of EstimateValues takes time $O((||\chi^{(r,t)}||_0 \log n + F B_t \log n) \cdot R_t) = O(R_t k \log n + R_t B_t \log n)$, as $F = O(1)$ by choice of parameters in line 25 of Algorithm 2. The total runtime per iteration in lines 16-22 is thus

$$\begin{aligned}
&\sum_{t=1}^{T} O(R_t k \log n + R_t B_t \log n)\\
&= O\left(kR_T \log n + \sum_{t=1}^{T} B_t R_t \log n\right) \quad \text{(since } \sum_{t=1}^{T} R_t = O(R_T)\text{, as } R_t \text{ grow geometrically)}\\
&= O(kR_T \log n) + O\left(\sum_{t=1}^{T} k/R_t\right) \log n \quad \text{(since } B_t = C_2 k/R_t^2\text{)}
\end{aligned}$$

We now note that $\sum_{t=1}^{T} k/R_t = O(k)$ since R_t grow geometrically, and thus the expression on the last line above is $O(kR_T \log n) = k \log^{2+2\delta} n$ by (33). Since the loop in lines 16-22 proceeds over $O(\log n)$ iterations, the final runtime bound is $k \log^{3+2\delta} n$, as required (after rescaling δ).

Finally, lines 27-31 take $O(\frac{1}{\epsilon} k \log n)$ time for the invocation of HASHTOBINS by Lemma 2.8 and $O(\frac{1}{\epsilon} k \log n)$ time for ESTIMATEVALUES by Lemma 2.11. Putting the bounds above together, we obtain the runtime of $k \log^{3+2\delta} n + O(\frac{1}{\epsilon} k \log n)$, as required. □

5 Sample efficient recovery

In this section we state our algorithm for sparse recovery from Fourier measurements that achieves $O(k \log n)$ sample complexity in $k \log^{O(1)} n$ runtime, give an outline of the analysis, and then present the formal proof. The proof reuses the core primitives developed in Section 3 together with the idea of majorizing sequences used in Section 4 to argue about correctness of our estimation primitive to analyze the performance of a natural iterative recovery scheme.

5.1 Algorithm and outline of the analysis

Our algorithm (Algorithm 3) contains three major components: it starts by taking measurements m of the signal x (accessing the signal in Fourier domain, i.e. accessing $\widehat{x}$), then uses these measurements to perform a sequence of recovery steps that reduce the ℓ_1 norm of the 'heavy' elements of x down to (essentially) noise level μ. Finally, a simple cleanup procedure (RECOVERATCONSTANTSNR) is run to achieve the ℓ_2/ℓ_2 sparse recovery guarantees (see (3)). We reuse the location primitive from [Kap16] (LOCATESIGNAL, Algorithm 6).

Measuring $\widehat{x}$. All measurements that the algorithm takes are taken in lines 6-22. Two sets of measurements are taken: one for location (LOCATESIGNAL), another for estimation purposes (calls to ESTIMATEVALUES in line 32 of Algorithm 3). Location relies on a very structured set of measurements: the measurements are taken over $T = \frac{1}{1-\delta} \log_2 \log(k+1) + O(1)$ rounds for small constant $\delta \in (0, 1/2)$, where in round t we are hashing the signal into $B_t \approx k/R_t^2$ buckets, where R_t grows exponentially with t. For each t we perform R_t independent hashing experiments of this type. For each hashing $H_{t,s}, t = 1, \ldots, T, s = 1, \ldots, R_t$ we select a random set $\mathcal{A}_{t,s} \subseteq [n] \times [n]$ that encodes the locations that our measurements access. Besides measurements used for location we take a separate set of measurements to use in the call to ESTIMATEVALUES. These measurements are quite unstructured: we simply make measurements using $C \log n$ random hashings and evaluation points for sufficiently large constant $C > 0$. It is crucial that these measurements are independent of the measurements used for location. Intuitively, the first set of measurements allows us to decode dominant coefficients of the residual signal in sublinear time, whereas the second (unstructured) set of measurements allows us to prune false positives, ensuring that no erroneous coefficients are introduced throughout the update process. The latter idea is similar to the approach used in [IK14], but is harder to implement in our setting as the number of possible trajectories along which the decoding process can evolve is larger. We handle this issue by using the notion of majorizing sequences introduced in Section 2 (see Definition 2.12 and Lemma 2.13) and used to analyze Algorithm 2 in Section 4.

Signal to noise ratio (SNR) reduction loop. Once the samples have been taken, Algorithm 3 proceeds to the signal to noise (SNR) reduction loop (lines 25-36). The objective of this loop is to reduce the mass of the top (about k) elements in the residual signal to roughly the noise level $\mu \cdot k$, where $\mu \geq ||x_{[n]\setminus[k]}||_2/\sqrt{k}$. Specifically, we define the set S of 'head elements' in the original signal x as

$$S = \{i \in [n] : |x_i| > \mu\}. \tag{34}$$

Note that we have $|S| \leq 2k$. Indeed, if $|S| > 2k$, more than k elements of S belong to the tail, amounting to more than $\mu^2 \cdot k = \mathrm{Err}_k^2(x)$ tail mass. The quantities e^{head} and e^{tail} (see (8) and (9) in Section 2) used in this section are defined with respect to this set S.

The SNR reduction loop of Algorithm 3 constructs a vector $\tilde{\chi}$ supported only on S such that

$$||(x - \tilde{\chi})_S||_1 = O(\mu k) \quad \text{and} \quad \operatorname{supp} \tilde{\chi} \subseteq S, \tag{35}$$

i.e. the ℓ_1-SNR of the residual signal on the set S of heavy elements is reduced to a constant.

The main technical contribution lies in our SNR reduction loop, and our main technical result in this section is

Theorem 5.1. *For any $\delta \in (0, 1/2)$, for any $x \in \mathbb{C}^n$, any integer $k \geq 1$, if $\mu^2 \geq \mathrm{Err}_k^2(x)/k$ and $R^* \geq ||x||_\infty/\mu$, $R^* = n^{O(1)}$, the following conditions hold for the set $S := \{i \in [n] : |x_i| > \mu\} \subseteq [n]$.*

Then the SNR reduction loop of Algorithm 3 (lines 25-36) returns $\tilde{\chi}$ such that

$$||(x - \tilde{\chi})_S||_1 = O_\delta(\mu k)$$
$$\operatorname{supp} \tilde{\chi} \subseteq S$$

with probability at least $1 - 3/25$ over the internal randomness used by Algorithm 3. The sample complexity is $O_\delta(k \log n)$. The runtime is bounded by $O_\delta(k \log^{4+2\delta} n)$.

Recovery at constant ℓ_1-SNR and final result. Once (35) has been achieved, we run the RECOVERATCONSTANTSNR primitive from [Kap16] on the residual signal. Adding the correction that it outputs to the output of the SNR reduction loop gives the final output of the algorithm. Given Theorem 5.1, the proof of the main result is simple using

Lemma 5.2 (Lemma 3.4 of [Kap16]). *For any $\epsilon > 0$, $\hat{x}, \chi \in \mathbb{C}^n$, $x' = x - \chi$ and any integer $k \geq 1$ if $||x'_{[2k]}||_1 \leq O(||x_{[n]\setminus[k]}||_2\sqrt{k})$ and $||x'_{[n]\setminus[2k]}||_2^2 \leq ||x_{[n]\setminus[k]}||_2^2$, the following conditions hold. If $||x||_\infty/\mu = n^{O(1)}$, then the output χ' of* RECOVERATCONSTANTSNR*$(\hat{x}, \chi, 2k, \epsilon)$ satisfies $||x' - \chi'||_2^2 \leq (1 + O(\epsilon))||x_{[n]\setminus[k]}||_2^2$ with at least $99/100$ probability over its internal randomness. The sample complexity is $O(\frac{1}{\epsilon}k \log n)$, and the runtime complexity is at most $O(\frac{1}{\epsilon}k \log^2 n)$.*

Theorem 1.2 (Restated) *For any $\epsilon \in (1/n, 1), \delta \in (0, 1/2)$, $x \in \mathbb{C}^n$ and any integer $k \geq 1$, if $R^* \geq ||x||_\infty/\mu$, $R^* = n^{O(1)}$, $\mu^2 \geq ||x_{[n]\setminus[k]}||_2^2/k$, $\mu^2 = O(||x_{[n]\setminus[k]}||_2^2/k)$ and $\alpha > 0$ is smaller than a function of δ,* SPARSEFFT*$(\hat{x}, k, \epsilon, R^*, \mu)$ (Algorithm 3) solves the ℓ_2/ℓ_2 sparse recovery problem using $O_\delta(k \log n) + O(\frac{1}{\epsilon}k \log n)$ samples and $O_\delta(\frac{1}{\epsilon}k \log^{4+\delta} n)$ time with at least $4/5$ success probability.*

Proof. Let the set $S \subseteq [n]$ be defined as in Theorem 5.1. By Theorem 5.1 one has that $||(x - \tilde{\chi})_S||_1 = O_\delta(\mu)$ and $\operatorname{supp} \tilde{\chi} \subseteq S$ with probability at least $1 - 3/25$. Thus, the signal $x - \tilde{\chi}$ satisfies preconditions of Lemma 5.2, and we get $||x - \tilde{\chi} - \chi'||_2 \leq (1 + O(\epsilon)) \mathrm{Err}_k(x)$ with probability at least $99/100$, resulting in success probability at least $1 - 3/25 - 1/100 \geq 4/5$ overall.

The sample complexity of the SNR reduction loop is $O(k \log n)$ by Theorem 5.1. The sample complexity of RECOVERATCONSTANTSNR is $O(\frac{1}{\epsilon}k \log n)$. The runtime of the SNR reduction loop is bounded by $k \log^{4+2\delta} n$ by Theorem 5.1, and the runtime of RECOVERATCONSTANTSNR is at most $O(\frac{1}{\epsilon}k \log^2 n)$ by Lemma 5.2, so the final runtime bound follows (after rescaling δ). □

Algorithm 3 SPARSEFFT($\hat{x}, k, \epsilon, R^*, \mu$)

```
1:  procedure SPARSEFFT($\hat{x}, k, \epsilon, R^*, \mu$)
2:      $\mathcal{W} \leftarrow \{\mathbf{0}\}$, $\Delta \leftarrow 2^{\lfloor \frac{1}{2}\log_2 \log_2 n \rfloor}$, $N \leftarrow \Delta^{\lceil \log_\Delta n \rceil}$
3:      for $g = 1$ to $\log_\Delta N$ do
4:          $\mathcal{W} \leftarrow \mathcal{W} \cup \{N\Delta^{-g}\}$
5:      end for
6:      $T \leftarrow \frac{1}{1-\delta}\log_2 \log(k+1) + O(1)$
7:      $R_t \leftarrow C_1 \cdot 2^t$ for $t \in [1:T]$        ▷ $C_1 > 0$ an absolute constant, $\delta \in (0, 1/2)$ small constant
8:      $B_t \leftarrow C_2 \cdot k/R_t^2$ for $t \in [1:T]$        ▷ $C_2$ sufficiently large
9:      $G_t \leftarrow$ filter with $B_t$ buckets and sharpness $F = 8$.
10:     for $t = 1$ to $T$ do        ▷ Take samples to be used for location
11:         for $s = 1$ to $R_t$ do
12:             Choose $\sigma \in \mathcal{M}_{odd}$ u.a.r., let $\pi_{t,s} \leftarrow (\sigma, 0)$, $H_{t,s} := (\pi_{t,s}, B_t, F)$
13:             Let $\mathcal{A}_{t,s} \leftarrow C \log\log n$ elements of $[n] \times [n]$ u.a.r.
14:             $m(x, H_{t,s}, \alpha + \mathbf{w}\cdot\beta) \leftarrow$ HASHTOBINS($\hat{x}, 0, (H_{t,s}, \alpha + \mathbf{w}\cdot\beta)$) for $(\alpha, \beta) \in \mathcal{A}_{t,s}, \mathbf{w} \in \mathcal{W}$
15:         end for
16:     end for
17:     $B \leftarrow k/\alpha^2$, $\alpha \in (0, 1)$ smaller than a constant
18:     for $t = 1$ to $C \log n$ do
19:         Choose $\sigma \in \mathcal{M}_{odd}$, $q, z_t \in [n]$ u.a.r., let $\pi_t^{est} \leftarrow (\sigma, q)$, $H_t^{est} := (\pi_t^{est}, B, F)$
20:         $m(x, H_t^{est}, z_t) \leftarrow$ HASHTOBINS($\hat{x}, 0, (H_t^{est}, z_t)$)
21:     end for
22:     $\mathcal{M}^{est} \leftarrow \{(H_t^{est}, z_t, m(x, H_t^{est}, z_t))\}_{t=1}^{C \log n}$
23:     $\chi^{(0,0)} \leftarrow 0$, $\chi' \leftarrow 0$
24:     $r' \leftarrow 0, t' \leftarrow 0$
25:     for $r = 0, 1, \ldots, \lfloor \log_4 R^* \rfloor - 3$ do
26:         for $t = 1$ to $T$ do
27:             $\chi^{(r,t)} \leftarrow \chi^{(r',t')} + \chi'$
28:             for $s = 1$ to $R_t$ do        ▷ Invocation of LOCATESIGNAL below does not take any fresh samples
29:                 $L_s \leftarrow$ LOCATESIGNAL($\chi^{(r',t')}, H_{t,s}, \{m(x, H_{t,s}, \alpha + \mathbf{w}\cdot\beta)\}_{(\alpha,\beta)\in\mathcal{A}_{t,s}, \mathbf{w}\in\mathcal{W}}$)
30:             end for
31:             $L \leftarrow \bigcup_{s=1}^{R_t} L_s$        ▷ Invocation of ESTIMATEVALUES below does not take any fresh samples
32:             $\chi \leftarrow$ ESTIMATEVALUES($\chi^{(r',t')}, L, \mathcal{M}^{est}$)
33:             For all $j \in \operatorname{supp}\chi$ let $\chi'_j \leftarrow \chi_j$ if $|\chi_j| \geq \frac{1}{16}R^*\mu(1/4)^r$ and $\chi'_j \leftarrow 0$ otherwise
34:             $r' \leftarrow r$, $t' \leftarrow t$
35:         end for
36:     end for
37:     $\tilde{\chi} \leftarrow \chi^{(r',t')} + \chi'$
38:     $\chi'' \leftarrow$ RECOVERATCONSTANTSNR($\hat{x}, \tilde{\chi}, 2k, \epsilon$)
39:     $\chi^* \leftarrow \tilde{\chi} + \chi''$
40:     return $\chi^*$
41: end procedure
```

In the rest of this section we prove performance guarantees for the SNR reduction loop in Algorithm 3 (lines 23-35). These guarantees are formally stated in Theorem 5.1, our main result in the rest of the section. The main tool in our analysis is the notion of a majorizing sequence for the intermediate residual signals that arise in the SNR reduction loop: we show that with high probability over the measurements taken, the intermediate residual signals that arise during the execution of the algorithm are (assuming perfect estimation) majorized by a fixed sequence of signals $y^{(r,t)}$, constructed in section 5.3.

To prove that the residual signal is indeed with high probability majorized by this sequence $y^{(r,t)}$, we use the fact that our estimation primitive uses $C \log n$ random measurements and hence yields precise bounds for all signals $y^{(r,t)}$ in the majorizing sequence. This means that estimates provided by ESTIMATEVALUES essentially provide perfect estimation for our algorithm, and a simple inductive argument shows that $y^{(r,t)}$ majorizes $x-\chi^{(r,t)}$ at each iteration indeed, and no false positives are created. This argument crucially relies on the definition of a majorant (see Definition 2.12) and a monotonicity property of e^{head} (Lemma 2.13). We first state notation relevant to bounding the effect of tail noise on location in section 5.2. Then the construction of the majorizing sequence is given in section 5.3, and then section 5.4 proves Theorem 5.1.

5.2 Notation for bounding tail noise in location

Our location algorithm (presented in Appendix C) uses several values of $(\alpha, \beta) \in \mathcal{A}_r \subseteq [n] \times [n]$ to perform location, a more robust version of $e_i^{tail}(H, z)$ will be useful. To that effect we let for any $\mathcal{Z} \subseteq [n]$

$$e_i^{tail}(H, \mathcal{Z}, x) := \text{quant}^{1/5}_{z \in \mathcal{Z}} \left| G^{-1}_{o_i(i)} \cdot \sum_{j \in [n] \setminus S} G_{o_i(j)} x_j \omega^{z\sigma(j-i)} \right|. \tag{36}$$

Note that our Sparse FFT algorithm (Algorithm 3) at various iterations r, first selects sets $\mathcal{A}_r \subseteq [n] \times [n]$, and then accesses the signal at locations $\mathcal{Z} = \{\alpha + \mathbf{w} \cdot \beta\}_{(\alpha,\beta) \in \mathcal{A}_r}$ for various $\mathbf{w} \in \mathcal{W}$. It should also be noted here that in the definition above the quantile is taken over all values of $z \in \mathcal{Z}$ for a fixed hashing H.

The definition of $e_i^{tail}(H, \{\alpha + \mathbf{w} \cdot \beta\}, x)$ for a fixed $\mathbf{w} \in \mathcal{W}$ above allows us to capture the amount of noise that our measurements that use H suffer from for locating a specific set of bits of σi. Since the algorithm requires all $\mathbf{w} \in \mathcal{W}$ to be not too noisy in order to succeed, the following quantity will be useful in analysis. We define

$$e_i^{tail, \mathcal{W}}(H, \mathcal{A}, x) := 40\mu_{H,i}(x) + \sum_{\mathbf{w} \in \mathcal{W}} \left| e_i^{tail}(H, \{\alpha + \mathbf{w} \cdot \beta\}_{(\alpha,\beta) \in \mathcal{A}}, x) - 40\mu_{H,i}(x) \right|_+ \tag{37}$$

where for any $\eta \in \mathbb{R}$ one has $|\eta|_+ = \eta$ if $\eta > 0$ and $|\eta|_+ = 0$ otherwise.

The following definition is useful for bounding the norm of elements $i \in S$ that are not discovered by several calls to LOCATESIGNAL on a sequence of hashings $\{H_r\}$. For a sequence of measurement patterns $\{H_r, \mathcal{A}_r\}$ we let

$$e^{tail, \mathcal{W}}(\{H_r, \mathcal{A}_r\}, x) := \text{quant}^{1/5}_r e_i^{tail, \mathcal{W}}(H_r, \mathcal{A}_r, x). \tag{38}$$

We will use the following lemma, whose proof is given in Appendix C:

Lemma 5.3. *For any integer $r_{max} \geq 1$, for any sequence of r_{max} hashings $H_r = (\pi_r, B, R), r \in [1 : r_{max}]$ and evaluation points $\mathcal{A}_r \subseteq [n] \times [n]$, for every $S \subseteq [n]$ and for every $x, \chi \in \mathbb{C}^n, x' := x - \chi$, the following conditions hold. If for each $r \in [1 : r_{max}]$ $L_r \subseteq [n]$ denotes the output of* LOCATESIGNAL*($\widehat{x}, \chi, H_r, \{m(x, H_r, \alpha + \mathbf{w} \cdot \beta)\}_{(\alpha,\beta) \in \mathcal{A}_r, \mathbf{w} \in \mathcal{W}}$), $L = \bigcup_{r=1}^{r_{max}} L_r$, and the sets $\{\beta\}_{(\alpha,\beta) \in \mathcal{A}_r}$ are balanced $r \in [1 : r_{max}]$, then*

$$||x'_{S \setminus L}||_1 \leq 20||e_S^{head}(\{H_r\}, x')||_1 + 20||e_S^{tail, \mathcal{W}}(\{H_r, \mathcal{A}_r\}, x)||_1 + |S| \cdot n^{-\Omega(c)}. \tag{*}$$

Furthermore, every element $i \in S$ such that

$$|x'_i| > 20(e_i^{head}(\{H_r\}, x') + e_i^{tail, \mathcal{W}}(\{H_r, \mathcal{A}_r\}, x)) + n^{-\Omega(c)} \tag{**}$$

belongs to L.

We will also use the following lemma, whose proof is given in Appendix D:

Lemma 5.4. *For every C_1 larger than an absolute constant, every integer $k \geq 1$ and every $x \in \mathbb{C}^n$, if the parameter μ satisfies $\mu \geq ||x_{[n]\setminus[k]}||_2/\sqrt{k}$, the following conditions hold. If hashings $\{\{H_{t,s}\}_{s=1}^{R_t}\}_{t=1}^{T}$ and locations $\{\{\mathcal{A}_{t,s}\}_{s=1}^{R_t}\}_{t=1}^{T}$ are selected as in Algorithm 3, lines 6-16, the sequence $R_1, \ldots, R_T$ satisfies*

q1 $R_t = C_1 2^t$ *for all* $t \geq 0$*;*

q2 $B_t = C_2(2k)/R_t^2$*,*

wheret $C_2 > 0$ is sufficiently large (as a function of C_1), then there exists an event $\mathcal{E}_{small-noise}$ (that depends on $H_{t,s}$ and $\mathcal{A}_{t,s}$) with $\mathbf{Pr}[\bar{\mathcal{E}}_{small-noise} \wedge \mathcal{E}_{partition}] \leq 1/1000$ (where $\mathcal{E}_{partition}$ is the success event for Lemma 3.8) such that the following conditions hold conditioned on $\mathcal{E}_{small-noise} \cap \mathcal{E}_{partition}$. For $e^{tail,\mathcal{W}}$ defined with respect to $S := \{i \in [n] : |x_i| > \mu\}$ one has for all $t \in [1:T]$ simultaneously $||e_{S_t}^{tail,\mathcal{W}}(\{H_{t,s}, \mathcal{A}_{t,s}\}_{s\in[1:R_t]}, x)||_1 \leq \frac{1}{200}||x_{[n]\setminus k}||_2\sqrt{k}/R_{t-1}$.

5.3 Construction of a majorizing sequence

We now construct a sequence of vectors $y^{t,r} \in \mathbb{R}_+^{[n]}$, where $t = 1, \ldots, T$ and $r = 0, 1, \ldots, \lfloor \log_2 R^* \rfloor - 3$, which, as we show later, will majorize the actual sequence of residual signals that arise in the execution of our algorithm on the set of head elements S assuming expected behaviour of our estimation primitive. These two properties together will later ensure that the update vectors $\chi^{(r',t')}$ that the SNR reduction loop computes are always supported on S.

To define the majorizing sequence, we first let $y_i^{(0,0)} = R^*\mu$ for all $i \in S$ and $y_i^{(0,0)} = 0$ otherwise. Note that $y^{(0,0)}$ trivially majorizes every x with the property that $||x||_\infty \leq R^* \cdot \mu$ on S. The construction of $y^{(r,t)}$ proceeds by induction on (r,t). Given $y^{(r',t')}$, the next signal to be defined is $y^{(r,t)}$, where $(r,t) = (r', t'+1)$ if $t' < T$ and $(r,t) = (r'+1, 1)$ otherwise (note that this notation matches the notation in lines 23-35) of Algorithm 3, i.e. the SNR reduction loop. We now define the signal $y^{(r,t)}$ by letting for each $i \in S_t$

$$y_i^{(r,t)} := \max\left\{20e_i^{head}(\{H_{t,s}\}_{s\in[1:R_t]}, y^{(r',t')}) + 20e_i^{tail,\mathcal{W}}(\{H_{t,s}, \mathcal{A}_{t,s}\}_{s\in[1:R_t]}, x) + n^{-\Omega(c)}, \frac{1}{8}\cdot(1/4)^r R^*\mu\right\} \tag{39}$$

and letting $y_i^{(r,t)} := y_i^{(r',t')}$ otherwise. Here $n^{-\Omega(c)}$ corresponds to the (negligible) error term due to polynomial precision of our computations. Note that there are two contributions to $y^{(r,t)}$: one coming from the previous signal in the majorizing sequence, namely $y^{(r',t')}$, and the other coming from the tail of the signal x. Also, recall that the quantities e^{head} and e^{tail} (see (8) and (9) in Section 2) used in this section are defined with respect to the set S given by (34).

The ℓ_1 norm of the majorizing sequence satisfies useful decay properties:

Lemma 5.5. *For every $\delta \in (0, 1/2)$, every even $F \geq 6$, every $x \in \mathbb{C}^n$, every integer $k \geq 1$, if $\mu \geq ||x_{[n]\setminus[k]}||_2/\sqrt{k}$, $R^* \geq ||x||_\infty/\mu, R^* = n^{O(1)}$, and $S = \{i \in [n] : |x_i| \geq \mu\}$, then the following conditions hold.*

If $e^{head}, e^{tail,\mathcal{W}}$ are defined with respect to S, hashings $\{H_{t,s}\}$, sets $\{\mathcal{A}_{t,s}\}$ are defined as in Algorithm 3, parameters R_t, B_t satisfy

q1 $R_t = C_1 2^t$ *for all* $t \geq 0$*,* C_1 *larger than a function of* δ*;*

q2 $B_t = C_2(2k)/R_t^2$*, where* C_2 *is larger than a function of* C_1 *and* δ*,*

and the sequence $y^{(r,t)}$ is defined as in (39)*, then there exists an event $\mathcal{E}_{maj}$ with $\mathbf{Pr}_{\{\{H_{t,s}\}_{s\in[1:R_t]}\}_{t=1}^{T}}[\mathcal{E}_{maj}] \geq 1 - 2/25$ such that conditioned on $\mathcal{E}_{maj}$ the set S admits an isolating partition (as per Definition 3.7) $S = S_1 \cup S_2 \cup \ldots \cup S_T$, and the following hold.*

For every $(r,t) \in [1:T] \times [0:\lfloor \log_4 R^* \rfloor] \cup \{(0,0)\}$

(A) *for all* $q \in [1:t]$ *one has* $||y^{(r,t)}_{S_q}||_1 \leq R^*\mu \cdot (1/4)^{r+1} \cdot (2k) \cdot (R_0/R_{q-1})^{\delta}$*;*

(B) *for all* $q \in [t+1:T]$ *one has* $||y^{(r,t)}_{S_q}||_1 \leq R^*\mu \cdot (1/4)^{r} \cdot (2k) \cdot (R_0/R_{q-1})^{\delta}$*;*

(C) $||y^{(r,t)}_{S}||_1 \leq (2/\delta) \cdot R^*\mu(1/4)^{r} \cdot (2k)$

Proof. By Lemma 3.8 applied to the set S (recall that $|S| \leq 2k$) we get that conditioned on an event $\mathcal{E}_{partition}$ that occurs with probability at least $1-1/25$ there exists an isolating partition $S = S_1 \cup \ldots \cup S_T$. We condition on $\mathcal{E}_{partition}$ in what follows, and the event $\mathcal{E}_{maj}$ that we construct later will be a subset of $\mathcal{E}_{partition}$.

We prove the claims by induction on (r,t). The **base** is provided by $r = 0$ and $t = 0$. Indeed, by property **(1)** of an isolating partition (see Definition 3.7) and the fact that $|S| \leq 2k$ we have for any $q \in [1:T]$

$$||y||_{S_q} \leq R^*\mu \cdot |S_q| \leq R^*\mu \cdot (2k) \cdot \frac{R_0}{R_{q-1}} 2^{-2^{(1-\delta)(q-1)}+1} \leq R^*\mu(2k) \cdot (R_0/R_{q-1})^{\delta}$$

since $\delta \in (0,1)$ by assumption of the lemma and $2^{-2^{(1-\delta)(q-1)}+1} \leq 1$ for all $q \geq 1$.

We now prove the **inductive step**. There are two cases, depending on whether $t \in [1:T-1]$ or $t = T$. Let $t' = t-1, r = r'$ if $t > 1$ and $t' = T, r' = r-1$ otherwise. If $t = 1, r = 0$, then let $t' = 0, r' = 0$. Note that (r', t') is the element preceding $y^{(r,t)}$ in the majorizing sequence.

We start with an upper bound on the ℓ_1 norm of $y^{(r',t')}$. Using the inductive hypothesis for (r',t'), we get

$$\begin{aligned}
||y^{(r',t')}_{S}||_1 &\leq \sum_{q=1}^{t'} R^*\mu \cdot (2k) \cdot (1/4)^{r'+1} \cdot (R_0/R_{q-1})^{\delta} + \sum_{q=t'+1}^{\infty} R^*\mu \cdot (2k) \cdot (1/4)^{r} \cdot (R_{q-1}/R_0)^{-\delta} \\
&\leq R^*\mu \cdot (2k) \cdot (1/4)^{r'} \sum_{q=1}^{\infty} (R_{q-1}/R_0)^{-\delta} \\
&= R^*\mu \cdot (2k) \cdot (1/4)^{r'} \sum_{q=1}^{\infty} 2^{-(q-1)\delta} \qquad \text{(since } R_t = C_1 2^t \text{ by } \mathbf{p1}\text{)} \\
&\leq \frac{1}{2^{\delta}-1} \cdot R^*\mu \cdot (2k) \cdot (1/4)^{r'} \\
&\leq \frac{1}{e^{\delta \ln 2}-1} \cdot R^*\mu (1/4)^{r'} \cdot (2k) \\
&\leq (2/\delta) \cdot R^*\mu (1/4)^{r'} \cdot (2k) \qquad \text{(since } e^x - 1 \geq x \text{ when } x \leq 1 \text{ and } \ln 2 > 1/2\text{)}
\end{aligned} \tag{40}$$

By definition of the majorizing sequence (39) the signal $y^{(r,t)}$ is obtained from $y^{(r',t')}$ by modifying the latter on S_t. We need to bound the error introduced by head and tail elements of $y^{(r',t')}$ to $y^{(r,t)}_{S_t}$ (see (39)). We now bound both terms. By Lemma 5.4 conditioned on $\mathcal{E}_{small-noise} \cap \mathcal{E}_{partition}$ (defined in the lemma) we have

$$||e^{tail, \mathcal{W}}_{S_t}(\{H_{t,s}, \mathcal{A}_{t,s}\}_{s \in [1:R_t]}, x)||_1 \leq \frac{1}{200} \mu k / R_{t-1}. \tag{41}$$

By Lemma 3.1 we have

$$||e^{head}_{S_t}(\{H_{t,s}\}_{s \in [1:R_t]}, y^{(r',t')})||_1 \leq 40 R_t^{-\delta} ||y^{(r',t')}_{S}||_1 \tag{42}$$

as long as S admits an isolating partition with respect to the hash functions $\{\{H_{t,s}\}\}$, which it does with probability at least $1-1/25$ by Lemma 3.8 (the success event is denoted by $\mathcal{E}_{partition}$). We now define the event $\mathcal{E}_{maj}$ by letting $\mathcal{E}_{maj} := \mathcal{E}_{small-noise} \cap \mathcal{E}_{partition}$. Note that $\mathbf{Pr}[\mathcal{E}_{maj}] \geq 1 - 2/25$, as required. We condition on $\mathcal{E}_{maj}$ for the rest of the proof.

We now use the bounds above to prove the result. By definition of the majorizing sequence (39) we have

$$y_i^{(r,t)} := \max\left\{20e_i^{head}(\{H_{t,s}\}_{s\in[1:R_t]}, y^{(r',t')}) + 20e_i^{tail,\mathcal{W}}(\{H_{t,s},\mathcal{A}_{t,s}\}_{s\in[1:R_t]}, x) + n^{-\Omega(c)}, \frac{1}{8}\cdot(1/4)^r R^*\mu\right\}.$$

so using the bound from (42) we get

$$\begin{aligned} ||y_{S_t}^{(r,t)}||_1 &\le \sum_{i\in S_t}\Big(20e_i^{head}(\{H_{t,s}\}_{s\in[1:R_t]}, y^{(r',t')}) + 20e_i^{tail,\mathcal{W}}(\{H_{t,s},\mathcal{A}_{t,s}\}_{s\in[1:R_t]}, x) + n^{-\Omega(c)} \\ &\quad + \frac{1}{8}\cdot(1/4)^r R^*\mu\Big) \\ &\le 400\cdot R_t^{-\delta}\cdot||y_S^{(r',t')}||_1 + \frac{1}{8}((1/4)^r R^*\mu)\cdot|S_t| + 20||e_{S_t}^{tail,\mathcal{W}}(\{H_{t,s},\mathcal{A}_{t,s}\},x)||_1 + n^{-\Omega(c)}\end{aligned}$$

We now substitute (40) together with (41) into the last line above, and use the bound $|S_t| \le 2k\cdot\frac{R_0}{R_{t-1}}2^{-2^{(1-\delta)(t-1)}+1} \le 2k\frac{R_0}{R_{t-1}}$ (from the definition of an isolating partition, Definition 3.7) to obtain

$$\begin{aligned} ||y_{S_t}^{(r,t)}||_1 &\le 400\cdot R_t^{-\delta}((2/\delta)\cdot R^*\mu\cdot k(1/4)^{r'}) + \frac{1}{8}R^*\mu k(1/4)^{r'}\cdot\frac{R_0}{R_{t-1}} + \frac{1}{10}\mu k/R_{t-1} + n^{-\Omega(c)} \\ &\le \left((2000/\delta)\cdot(R_t/R_{t-1})^{-\delta}\cdot R_0^{-\delta} + \frac{1}{8} + \frac{1}{10}\right)\cdot R^*\mu\cdot k(1/4)^{r'}\cdot(R_0/R_{t-1})^{\delta} + n^{-\Omega(c)}, \qquad (43) \\ &\le \left((2000/\delta)\cdot R_0^{-\delta} + \frac{1}{8} + \frac{1}{10}\right)\cdot R^*\mu\cdot k(1/4)^{r'}\cdot(R_0/R_{t-1})^{\delta} + n^{-\Omega(c)},\end{aligned}$$

where we used the assumption that $r \le \lfloor\log_4 R^*\rfloor$, so that $R^*\mu(1/4)^r \ge \mu$.

We now note that for every $t \ge 1$, since $R_t = C_1 2^t$ by assumption of the lemma, we have

$$(2000/\delta)\cdot R_0^{-\delta} = (2000/\delta)\cdot C_1^{-\delta}.$$

Thus, as long as $C_1 \ge (2000\cdot 100\cdot\delta)^{1/\delta}$, the rhs is upper bounded by $1/100$. Substituting this into (43), we get

$$||y_{S_t}^{(r,t)}||_1 \le \left(\frac{1}{100} + \frac{1}{8} + \frac{1}{10}\right)R^*\mu\cdot k(1/4)^{r'}\cdot(R_0/R_{t-1})^{\delta} + n^{-\Omega(c)} \le R^*\mu k(1/4)^{r'+1}\cdot(R_0/R_{t-1})^{\delta} + n^{-\Omega(c)}.$$

This completes the proof of the inductive step. □

5.4 Proof of Theorem 5.1

We now prove the following lemma, which captures the correctness part of Theorem 5.1. We then put it together with runtime and sample complexity estimates to obtain a proof of Theorem 5.1.

Lemma 5.6. *For every $\delta \in (0, 1/2)$, every even $F \ge 6$, every $x \in \mathbb{C}^n$ if the parameter μ satisfies $\mu \ge ||x_{[n]\setminus[k]}||_2/\sqrt{k}$, $R^* = ||x||_\infty/\mu, R^* = n^{O(1)}$, the following conditions hold for the SNR reduction loop in Algorithm 3. If the hashings $\{H_{t,s}\}$ and locations $\{\mathcal{A}_{t,s}\}$ are chosen as in Algorithm 3, and parameters satisfy*

q1 $R_t = C_1 2^t$ *for all* $t \ge 0$, C_1 *larger than a function of* δ*;*

q2 $B_t = C_2(2k)/R_t^2$, *where* C_2 *larger than a function of* C_1 *and* δ,

then the following conditions hold. If $S := \{i \in [n] : |x_i| > \mu\}$, *then the output* $\tilde{\chi}$ *of the* ℓ_1*-SNR reduction loop in Algorithm 3 satisfies*

$$||(x-\chi)_S||_1 = O_\delta(||x_{[n]\setminus[k]}||_2\cdot\sqrt{k}),$$

and all intermediate $\chi^{r',t'}$ *satisfy*

$$\operatorname{supp}\chi^{r',t'} \subseteq S$$

with probability at least $1 - 3/25$ *over the randomness used in the measurements.*

Proof. We start with an outline of the proof. Throughout the proof we rely on the quantities e^{head} and e^{tail} defined with respect to the set $S = \{i \in [n] : |x_i| > \mu\}$ defined in the lemma. The proof is by induction on the number of iterations of the SNR reduction loop. We will show that with high probability over the initial measurements the residual signals $x - \chi^{(r,t)}$ are majorized on S by the sequence $y^{(r,t)}$ defined in (39). This lets us argue that **(1)** with high probability over the measurements used for ESTIMATEVALUES estimation error *on the signals* $y^{(r,t)}$ is small, and then **(2)** conclude that since $x - \chi^{(r,t)}$ are majorized by $y^{(r,t)}$ on S, ESTIMATEVALUES gives precise estimates for all such residuals. This lets us argue that updates of the residual are always confined to the set S, and the residual is still majorized appropriately at the next iteration, giving the inductive proof. In what follows we condition on the event $\mathcal{E}_{maj}$ defined in Lemma 5.5, which occurs with probability at least $1 - 2/25$.

Precision bounds for ESTIMATEVALUES. We first prove bounds on the precision of the estimates provided by calls to ESTIMATEVALUES in the SNR reduction loop of Algorithm 3 (line 32). We have by Lemma 2.11, **(1a)** applied to the signals $y^{(r,t)} + x_{[n]\setminus S}$ and the set S, that with probability $1 - n^{-2}$ over the choice of measurements $\mathcal{M}^{est}$ (lines 17-21) of Algorithm 3 **for any** $\chi^{(r,t)} \in \mathbb{C}^n$ **such that** $\operatorname{supp} \chi^{(r,t)} \subseteq S$ **and** $x - \chi^{(r,t)}$ **is majorized by** $y^{(r,t)}$ **on** S (as per Definition 2.12), one has that the estimates w_i computed in the call ESTIMATEVALUES$(\chi^{(r,t)}, L, \mathcal{M}^{est})$ in line 32 satisfy

$$|w_i - (x - \chi^{(r,t)})_i| \leq 2\text{quant}_r^{1/5} e^{head}(\{H_r\}, x - \chi^{(r,t)}) + 2\text{quant}_r^{1/5} e^{tail}(\{H_r, a_r\}, x)$$

So in particular by Lemma 2.13 if $x - \chi^{(r,t)} \prec_S y^{(r,t)}$ and $\operatorname{supp} \chi^{(r,t)} \subseteq S$, we get

$$\begin{aligned}|w_i - (x - \chi^{(r,t)})_i| &\leq 2\text{quant}_r^{1/5} e^{head}(\{H_r\}, x - \chi^{(r,t)}) + 2\text{quant}_r^{1/5} e^{tail}(\{H_r, a_r\}, x)\\ &\leq 2\text{quant}_r^{1/5} e^{head}(\{H_r\}, y^{(r,t)}) + 2\text{quant}_r^{1/5} e^{tail}(\{H_r, a_r\}, x).\end{aligned}$$

By Lemma B.5, **(3)** and **(4)** one has $\text{quant}_r^{1/5} e^{head}(H_r, y) = O(||y_S||_1/B)$ and $\text{quant}_r^{1/5} e^{tail}(H_r, a_r, x) = O(||x_{[n]\setminus S}||_2/\sqrt{B})$ with probability $1 - 2^{-\Omega(C \log n)} \geq 1 - n^{-C/2}$ as long as the constant C is sufficiently large.[5]

Since $B = k/\alpha^2$ by our setting of parameters (line 17 of Algorithm 3), we have

$$O(||y_S||_1/B) \leq O(\alpha) \cdot \frac{1}{5}\alpha||y_S||_1/k \leq \frac{1}{5}\alpha||y_S||_1/k$$

and

$$O(||x_{[n]\setminus S}||_2/\sqrt{B}) \leq O(\sqrt{\alpha}) \cdot \frac{1}{2}\sqrt{\alpha}||x_{[n]\setminus S}||_2/\sqrt{k} \leq \frac{1}{2}\sqrt{\alpha}||x_{[n]\setminus S}||_2/\sqrt{k}$$

as long as α is smaller than a constant, and in particular smaller than δ^2 (we will need to set α smaller than δ^2 below to offset the $2/\delta$ factor in the upper bound on the ℓ_1 norm of $y^{(r,t)}$ in Lemma 5.5, **(C)**). We thus get that the estimates computed in ESTIMATEVALUES$(\chi, L, \mathcal{M}^{est})$ in line 32 satisfy

$$|w_i - (x - \chi^{(r,t)})_i| \leq \frac{1}{5}\alpha||y_S^{(r,t)}||_1/k + \frac{1}{2}\sqrt{\alpha}||x_{[n]\setminus S}||_2/\sqrt{k}. \tag{44}$$

We have by Lemma 5.5, **(C)**, that $||y_S^{(r,t)}||_1 \leq (2/\delta)R^*\mu \cdot (2k)(1/4)^r$, and we have by definition of S that $||x_{[n]\setminus S}||_2^2 \leq k \cdot ||x_{[n]\setminus S}||_\infty^2 + ||x_{[n]\setminus[k]}||_2^2 \leq k\mu^2 + ||x_{[n]\setminus[k]}||_2^2 \leq 2\mu^2 k$. Substituting these bounds into (44), we get by a union bound over all $i \in [n]$ and all sequences $y^{(r,t)}$ that if $x - \chi^{(r,t)} \prec_S y^{(r,t)}$, then for all $i \in [n]$ and all (r, t) one has

$$\begin{aligned}|w_i - (x - \chi^{(r,t)})_i| &\leq \frac{1}{5}\alpha||y_S^{(r,t)}||_1/k + \frac{1}{2}||x_{[n]\setminus S}||_2/\sqrt{k}\\ &\leq \frac{1}{5}\alpha(4/\delta)R^*\mu \cdot (1/4)^r + \frac{1}{2}\sqrt{\alpha}\sqrt{2}\mu\\ &\leq \sqrt{\alpha}\left(R^*\mu(1/4)^r + \mu\right),\end{aligned} \tag{45}$$

[5] Note that this is the place where we crucially use the notion of majorizing sequences: even though the actual residual signals that arise throughout the update process depend on the measurements $\mathcal{M}^{est}$, it suffices to invoke Lemma B.5 on the majorizing sequence y, which is fixed and independent of $\mathcal{M}^{est}$.

where we used the assumption that $\alpha \leq \delta^2$ to obtain the last inequality.

Equipped with the bounds on estimation quality in (45), we now give the proof of the theorem. The proof is by induction on (r, t). We prove that for every $(r, t) \in [1 : T] \times [0 : \lfloor \log_4 R^* \rfloor] \cup \{(0, 0)\}$

(A) $y^{(r,t)}$ majorizes $x - \chi^{(r,t)}$ on S;

(B) $\chi^{(r,t)}_{[n]\setminus S} \equiv 0$.

The **base** is provided by $(r, t) = (0, 0)$, where $y_i^{(0,0)} = R^*\mu$ for $i \in S$ and $y_i^{(0,0)} = 0$ otherwise. Since $||x||_\infty \leq R^*\mu$ by assumption of the lemma and $\chi^{(0,0)} = 0$ in Algorithm 3, the base of the induction holds. We now prove the **inductive step**. Let $t' = t - 1$, $r = r'$ if $t > 1$ and $t' = T, r' = r - 1$ otherwise. If $t = 1, r = 0$, then let $t' = 0, r' = 0$. Note that (r', t') is the element preceding $y^{(r,t)}$ in the majorizing sequence.

Since $x - \chi^{(r',t')} \prec_S y^{(r',t')}$ and $\operatorname{supp} \chi^{(r',t')} \subseteq S$ by the inductive hypothesis, we have by (45),

$$|w_i - (x - \chi^{(r',t')})_i| \leq \sqrt{\alpha} \left(R^*(1/4)^{r'}\mu + \mu \right), \tag{46}$$

where α is smaller than an absolute constant (see line 17 of Algorithm 3).

We first prove part **(B)** of the inductive step. Since only elements with $|w_i| \geq (1/16)R^*\mu(1/4)^r$ are updated (by the pruning step in line 33 of Algorithm 3), for all such i we have by triangle inequality using (46) that

$$|(x - \chi^{(r',t')})_i| \geq (1/16)R^*\mu(1/4)^r - \sqrt{\alpha}\,(R^*(1/4)^r\mu + \mu) \geq (1/32)R^*\mu(1/4)^r, \tag{47}$$

where we used the assumption that α is smaller than a sufficiently small absolute constant. Since the upper bound for r in the SNR reduction loop is $\lfloor \log_4 R^* \rfloor - 3$ in the SNR reduction loop, we have $(1/32)R^*\mu(1/4)^r \geq 2\mu > \mu$ for all such r. Since $\operatorname{supp} \chi^{(r',t')} \subseteq S$ by the inductive hypothesis, this means that the output χ' of the call to ESTIMATEVALUES is such that any $i \in [n]$ with $\chi'_i \neq 0$ belongs to S. We have shown that $\operatorname{supp} \chi^{(r,t)} \subseteq \operatorname{supp} \chi^{(r',t')} \cup \operatorname{supp} \chi' \subseteq S$, proving part **(B)** of the inductive step.

We now prove part **(A)** of the inductive step, i.e. prove that $x - \chi^{(r,t)} = x - \chi^{(r',t')} - \chi'$ is majorized by $y^{(r,t)}$ (defined by (39)).

Bounding elements reported in L. We first consider $i \in L \cap \operatorname{supp} \chi'$, i.e. elements that were reported in L and estimated as being above the threshold. For such $i \in L \cap \operatorname{supp} \chi'$ we have by (46)

$$|(x - \chi^{(r,t)})_i| = |(x - \chi^{(r',t')} - \chi')_i| \leq \sqrt{\alpha}\,(R^*(1/4)^r\mu + \mu) < (1/32)R^*\mu(1/4)^r.$$

At the same time for such elements ($i \in L \cap \operatorname{supp} \chi'$) we have by (47) $|(x - \chi^{(r',t')})_i| \geq (1/32)R^*\mu(1/4)^r$. This means that for all $i \in L \cap \operatorname{supp} \chi'$ one has

$$|(x - \chi^{(r,t)})_i| \leq |(x - \chi^{(r',t')})_i|, \tag{48}$$

as well as

$$|(x - \chi^{(r',t')})_i| \leq (1/32)R^*\mu(1/4)^r. \tag{49}$$

At the same time for $i \in [n]$ such that $\chi'_i = 0$ we have

$$|(x - \chi^{(r',t')})_i| = |(x - \chi^{(r,t)})_i| \leq (1/16)R^*\mu(1/4)^r + \sqrt{\alpha}(R^*(1/4)^r\mu + \mu) \leq (1/8)R^*\mu(1/4)^r \tag{50}$$

as long as α is smaller than an absolute constant.

Bounding elements not reported in L. Let $x' := x - \chi^{(r',t')}$ to simplify notation. If an element $i \in S_t$ is not reported in any of the calls to LOCATESIGNAL (i.e. does not belong to L), then by Corollary 5.3 it satisfies

$$
\begin{aligned}
|x'_i| &\le 20 e_i^{head}(\{H_{t,s}\}_{s\in[1:R_t]}, x') + 20 e_i^{tail,\mathcal{W}}(\{H_{t,s}, \mathcal{A}_{t,s}\}_{s\in[1:R_t]}, x) + n^{-\Omega(c)} \\
&\le 20 e_i^{head}(\{H_{t,s}\}_{s\in[1:R_t]}, y^{(r,t)}) + 20 e_i^{tail,\mathcal{W}}(\{H_{t,s}, \mathcal{A}_{t,s}\}_{s\in[1:R_t]}, x) + n^{-\Omega(c)},
\end{aligned}
\tag{51}
$$

where we used Lemma 2.13 to upper bound error induced by head elements of x' by error induced by head elements of $y^{(r,t)}$, which majorizes x' on S by the inductive hypothesis.

Putting it together. Recall that by (39) the signal $y^{(r,t)}$ is defined by letting for each $i \in S_t$

$$
y_i^{(r,t)} := \max\left\{20 e_i^{head}(\{H_{t,s}\}_{s\in[1:R_t]}, y^{(r',t')}) + 20 e_i^{tail}(\{H_{t,s}, \mathcal{A}_{t,s}\}_{s\in[1:R_t]}, x) + n^{-\Omega(c)}, \frac{1}{8}\cdot(1/4)^r R^*\mu\right\}
\tag{52}
$$

and letting $y_i^{(r,t)} := y_i^{(r',t')}$ otherwise.

We now have that any element $i \in S_t$ that is not reported in any of the calls to LOCATESIGNAL x'_i satisfies (51), which is upper bounded by the first argument in the maximum above. By (50) together with (49) we have $|(x' - \chi')_i| \le (1/8)R^*\mu(1/4)^r$ for all $i \in S_t$, which is upper bounded by the second term in the maximum above. Finally, for any $i \in [n]$ (not necessarily in S_t) by (48) we have $|(x' - \chi')_i| \le |x'_i|$, so $|(x' - \chi')_i| \le |x'_i| \le y_i^{(r',t')} = y_i^{(r,t)}$ for such i as well (note that we are also using the fact that $i \in S$ necessarily for all i with $\chi'_i \ne 0$, by part **(B)** of the inductive step, which we proved already). This completes the proof of part **(A)** of the inductive step, and the proof of the lemma.

Note that we conditioned on the event $\mathcal{E}_{maj}$ defined in Lemma 5.5, which occurs with probability at least $1 - 2/25$, as well as a high probability $(1 - 1/\text{poly}(n))$ success event for ESTIMATEVALUES. Thus, success probability is at least $1 - 3/25$ by a union bound, as required. □

We can now give a proof of Theorem 5.1, the main technical result of this section. We restate the theorem here for convenience of the reader:

Theorem 5.1 *(Restated) For any $\delta \in (0, 1/2)$, for any $x \in \mathbb{C}^n$, any integer $k \ge 1$, if $\mu^2 = \text{Err}_k^2(x)/k$ and $R^* \ge ||x||_\infty/\mu$, $R^* = n^{O(1)}$, the following conditions hold for the set $S := \{i \in [n] : |x_i| > \mu\} \subseteq [n]$.*

Then the SNR reduction loop of Algorithm 3 (lines 25-36) returns $\tilde{\chi}$ such that

$$
\begin{aligned}
&||(x - \tilde{\chi})_S||_1 = O_\delta(\mu k) \\
&\text{supp}\, \tilde{\chi} \subseteq S
\end{aligned}
$$

with probability at least $1 - 3/25$ over the internal randomness used by Algorithm 3. The sample complexity is $O_\delta(k \log n)$. The runtime is bounded by $O_\delta(k \log^{4+2\delta} n)$.

Proof. Correctness follows by Lemma 5.6 and setting of parameters in Algorithm 3. It remains to bound the sample and runtime complexity. For each $t = 1, \ldots, T$ we take B_t measurements using a filter of sharpness $F = O(1)$, so the total sample complexity is

$$
\begin{aligned}
\sum_{t=1}^{T} O(B_t)|\mathcal{W}| \cdot |\mathcal{A}_{t,s}| \cdot R_t &= \sum_{t=1}^{T} O(B_t)(\log n/\log\log n)(\log\log n) \cdot R_t \\
&\le O(C_2)((2k\log n)/R_t^2) \cdot R_t \\
&= O(C_2 k)\log n \cdot \sum_{t=1}^{T} R_t = O(k\log n),
\end{aligned}
$$

where we used the fact that C_1 is an absolute constant and C_2 as prescribed by Lemma 3.8, as well as the setting $|\mathcal{A}_{t,s}| = O(\log\log n)$ and $\mathcal{W} = O(\log_\Delta n) = O(\log n/\log\log n))$ Algorithm 3.

Runtime. We start by bounding the runtime for the SNR reduction loop

- Each call to LOCATESIGNAL costs $O(FB_t \log^2 n + ||\chi||_0 \log^2 n)$ by Lemma C.3.

  The total cost for calls to LOCATESIGNAL in a single iteration (i.e. one value of r) is hence bounded by

$$
\begin{aligned}
&O\left(\sum_{t=1}^{T}\sum_{s=1}^{R_t} B_t \log^2 n + (\max_{r',t'} ||\chi^{r',t'}||_0)\log^2 n\right)\\
&= O\left(\sum_{t=1}^{T} R_t B_t\right)\log^2 n + O(\sum_{t=1}^{T} R_t k \log^2 n) \quad \text{(since supp } \chi^{r',t'} \subseteq S \text{ for all } r', t' \text{ by Lemma 5.6)}\\
&= \sum_{t=1}^{T} O(k/R_t)\log^2 n + O(R_T)||\chi||_0 \log^2 n \quad \text{(since } R_t \text{ increase geometrically by setting of parameters in line 7)}\\
&= O(k\log^2 n) + ||\chi||_0 \log^{3+2\delta} n\\
&= k\log^{3+2\delta} n,
\end{aligned}
$$

  where we used the fact that

$$R_T = C_1 2^T = C_1 2^{\frac{1}{1-\delta}\log_2\log k + O(1)} = O(\log_2^{1/(1-\delta)} k) = O(\log_2^{1+2\delta} k)$$

  by setting of parameters in Algorithm 3 and the fact that $1/(1-\delta) \le 1 + 2\delta$ for $\delta \in (0, 1/2)$. Finally, accounting for $O(\log n)$ iterations of the SNR reduction loop over r, we obtain a bound of $k\log^{4+2\delta} n$, as claimed.

- Each call to ESTIMATEVALUES costs $O(FB \cdot \log n \cdot C\log n + (\max_{r',t'} ||\chi^{r',t'}||_0) \cdot \log n \cdot C\log n)$ by Lemma 2.11. The total runtime over $O(\log n)$ iterations of the SNR reduction loop is hence $O(k\log^3 n)$.

Summing the contributions, we get runtime $k\log^{4+2\delta} n$, as required. Success probability follows from the success probability of Lemma 5.6.

□

A Proof of Lemma 3.8

We restate the lemma for convenience of the reader:

Lemma 3.8 (Restated) *For every integer $k \ge 1$, every $S \subseteq [n], |S| \le k$, every $\delta \in (0, 1/2)$, if the parameters B_t, R_t are selected to satisfy* **(p1)** *$R_t = C_1 \cdot 2^t$ and* **(p2)** *$B_t \ge C_2 \cdot k/R_t^2$ for every $t \in [0:T]$, where C_1 is a sufficiently large constant and C_2 is sufficiently large as a function of C_1 and δ, then the following conditions hold.*

With probability at least $1 - 1/25$ over the choice of hashings $\{\{H_{t,s}\}_{s\in[1:R_t]}\}_{t=1}^{T}$ Algorithm 1 terminates in $T = \frac{1}{1-\delta}\log_2\log(k+1) + O(1)$ steps. When the algorithm terminates, the output partition $\{S_j\}_{j=1}^{T}$ is isolating as per Definition 3.8.

We will use

Theorem A.1 (Chernoff bound). *Let $X_1, \ldots, X_n$ be independent Bernoulli random variables, let $\mu := \mathbf{E}[\sum_{i=1}^n X_i]$. Then for any $\eta > 1$ one has $\mathbf{Pr}[\sum_{i=1}^n X_i > (1+\eta)\mu] \le e^{-\mu\eta/3}$.*

The following basic technical claim is crucial to our analysis (the short proof is given in Appendix F):

Claim A.2. *For every $C_1, C_2 > 0, \delta \in (0,1)$ there exists C_3 such that for every $C_4 \geq C_3$ one has $\frac{1}{C_4}2^{C_1 t} \cdot 2^{-C_2 2^{(1-\delta)t}+1} \leq 1$ all $t \geq 0$.*

Equipped with the technical claim above, we can now argue that Algorithm 1 constructs an isolating partition of any set $S \subseteq [n]$ that satisfies $|S| \leq k$ with at least high constant probability and prove Lemma 3.8.

Proof of Lemma 3.8: The proof proceeds in four steps. In **Step (1)** we state a set of inductive claims that we will prove, then in **Step (2)** argue that the inductive claims imply that Algorithm 1 terminates in $T = \frac{1}{1-\delta}\log_2 \log(k+1) + O(1)$ iterations, then in **Step (3)** argue that the inductive claims imply that output partition is isolating and finally in **Step (4)** prove the inductive claims (this step corresponds to the bulk of the proof).

Step (1) Our argument proceeds inductively for $t = 1, 2, \ldots$, and we think of sampling the hashings $\{H_{t,s}\}_{s=1}^{R_t}$ independently at each step t.

For each $t \geq 1$ let $k_t := |S_t^t|$. We show by induction on $t \geq 1$ that there exists a sequence of nested events $\mathcal{E}_1 \supseteq \mathcal{E}_2 \supseteq \ldots$ such that for all $t \geq 1$

(1) event $\mathcal{E}_t$ depends only on the randomness up to time t;

(2) $\mathbf{Pr}[\mathcal{E}_t] \geq 1 - \frac{3}{100}\sum_{t'=1}^{t-1}\frac{1}{R_{t'}}$;

(3) conditioned on $\mathcal{E}_t$ one has $k_t \leq k \cdot \frac{R_0}{R_{t-1}}2^{-2^{(1-\delta)(t-1)}+1}$.

Step (2) We now show that **(3)** implies that the algorithm terminates in $T = \frac{1}{1-\delta}\log_2\log(k+1) + O(1)$ steps. Indeed, by **(3)** we have

$$|S_T^T| \leq k \cdot \frac{R_0}{R_{T-1}}2^{-2^{(1-\delta)(T-1)}+1}.$$

Substituting $T = \frac{1}{1-\delta}(\log_2\log_2(k+1) + C)$, we get

$$|S_T^T| \leq k \cdot 2^{-2^{(1-\delta)(T-1)}+1} \leq k \cdot 2^{-\frac{1}{2}2^{(1-\delta)T}+1} \leq k \cdot 2^{-\frac{1}{2}2^{C}\cdot\log_2(k+1)+1} \leq 4(k+1)^{1-2^{C-1}} < 1$$

as long as $C \geq 3$.

Also, **(2)** implies that the algorithm terminates with probability at least

$$\begin{aligned}\mathbf{Pr}[\mathcal{E}_T] &\geq 1 - \frac{3}{100}\sum_{t=1}^{T-1}\frac{1}{R_t}\\ &\geq 1 - \frac{3}{100}\sum_{t=1}^{\infty}\frac{1}{C_1 2^t}\\ &\geq 1 - \frac{1}{25}\end{aligned}$$

where we used the assumption that $\{S_j\}_{j=1}^T$ satisfies property **(1)** of an isolating partition (see Definition 3.8) as well as $R_t = C_1 \cdot 2^t$ and C_1 is larger than an absolute constant.

Step (3) We now that given the inductive claims from **Step (1)**, the returned partition $S_1^T \cup \ldots \cup S_T^T$ satisfies the definition of a δ-isolating partition (Definition 3.8). We need to prove that

1. $|S_t^T| \leq k \cdot \frac{R_0}{R_{t-1}}2^{-2^{(1-\delta)\cdot(t-1)}+1}$;

2. no element of S_t^T is R_t^{-3}-crowded by S_t^T under any of $\{H_{t,s}\}_{s=1}^{R_t}$;

3. no element of S_t^T R_t-collides with a δ-bad element for S_t^T under any of $\{H_{t,s}\}_{s=1}^{R_t}$.

To prove the first property, we note that the sizes of sets $S_t^{t'}$ are non-increasing in $t' \geq t$ for every t, as $S_t^{t'} \supseteq S_t^t$ (by line 7 of Algorithm 1). Conditional on $\mathcal{E}_T$ we thus have

$$|S_t^T| \leq |S_t^t| \leq k \cdot \frac{R_0}{R_{t-1}} 2^{-2^{(1-\delta)(t-1)}+1}$$

for all $t \geq 1$, as required.

For the second property, note that no element of S_t^{t+1} is R_t^{-3}-crowded by S_t^t under any $\{H_{t,s}\}_{s=1}^{R_t}$ by construction of S_t^{t+1} (line 7 of Algorithm 1). Since $S_t^{t+1} \subseteq S_t^t$, this means that no element of S_t^{t+1} is R_t^{-3}-crowded by S_t^{t+1} under any $\{H_{t,s}\}_{s=1}^{R_t}$, and since $S_t^T \subseteq S_t^{t+1}$ this also means that no element of S_t^T is R_t^{-3}-crowded by S_t^T under any $\{H_{t,s}\}_{s=1}^{R_t}$, so property 2 is satisfied.

For the third property, note that no element of S_t^{t+1} R_t-collides with a δ-bad element for S_t^t under any $\{H_{t,s}\}_{s=1}^{R_t}$ by construction of S_t^{t+1} (line 7 of Algorithm 1). Since $S_t^T \subseteq S_t^{t+1}$, this means that no element of S_t^T R_t-collides with a δ-bad element for S_t^t under any $\{H_{t,s}\}_{s=1}^{R_t}$. Finally, note that since $S_t^T \subseteq S_t^t$, any element that is δ-bad for S_t^T is also δ-bad for S_t^t by Definition 3.4. This shows that no element of S_t^T R_t-collides with a δ-bad element for S_t^T under any $\{H_{t,s}\}_{s=1}^{R_t}$ and establishes property 3 above. This completes the proof that the constructed partition $\{S_t^T\}$ is isolating.

Step (4) In what follows we construct the events $\mathcal{E}_t, t = 1, \ldots, T$ and prove properties **(1)-(3)** above by induction on $t = 1, \ldots, T$. The proof is by induction on t.

Base:$t = 1$ We let $S_1^1 := S$, so that the base is trivial (we let $\mathcal{E}_1$ be the trivial event that occurs with probability 1).

Inductive step: $t \to t+1$ Suppose that $|S_t^t| = k_t \leq k \cdot \frac{R_0}{R_{t-1}} 2^{-2^{(1-\delta)(t-1)}+1}$. We first bound the expected size of U_t and V_t conditional on $\mathcal{E}_t$, and then put these bounds together to obtain a proof of the inductive step.

Bounding the number of crowded elements in S_t^t (size of V_t) For each element $i \in [n]$ and every scale $q \geq 0$ we have, letting $H := H_{t,s}, \pi := \pi_{t,s}$ and $h := h_{t,s}$ to simplify notation (recall that $h(i) = \text{round}((B/n)\pi(i))$; see section 2.1),

$$\mathbf{E}_H \left[\left| \pi(S_t^t \setminus \{i\}) \cap \mathbb{B}(\pi(i), \frac{n}{B_t} 2^q) \right| \right] \leq 4 \cdot 2^q |S_t^t| / B_t \leq 4 \cdot 2^q k_t / B_t, \tag{53}$$

where we used the fact that $|S_t^t| \leq k_t$ by the inductive hypothesis, as well as Lemma 2.5. Thus by Markov's inequality for every $\lambda > 0$

$$\mathbf{Pr}_H \left[\left| \pi(S_t^t \setminus \{i\}) \cap \mathbb{B}\left(\pi(i), \frac{n}{B_t} 2^q\right) \right| > \lambda \cdot 2^{2q} \right] \leq 4\lambda^{-1} 2^{-q} k_t / B_t.$$

By a union bound over all scales $q \geq 0$ (i.e. summing the rhs of the bound above over all scales $q \geq 0$) we conclude that

$$\mathbf{Pr}_H[i \text{ is } \lambda\text{-crowded under hashing } H] \leq \sum_{q \geq 0} 4\lambda^{-1} 2^{-q} k_t / B_t = 8\lambda^{-1} k_t / B_t. \tag{54}$$

We thus have for every $i \in [n]$ and a random hashing hashing $H = (\pi, B_t, G)$

$$\begin{aligned} \mathbf{Pr}_H[i \text{ is } R_t^{-3}\text{-crowded}] &\leq 8(R_t^3) \cdot k_t / B_t \qquad \text{(by (54) with } \lambda = R_t^{-3}) \\ &\leq \frac{8}{C_2}(R_t^5) \cdot \frac{R_0}{R_{t-1}} 2^{-2^{(1-\delta)(t-1)}+1}, \end{aligned} \tag{55}$$

where we used the bound $k_t \leq k \cdot \frac{R_0}{R_{t-1}} 2^{-2^{(1-\delta)(t-1)}+1}$ provided by the inductive hypothesis and the assumption that $B_t \geq C_2 \cdot k/R_t^2$ by assumption **p2** of the lemma to go from the first line to the second.

We thus have by a union bound over R_t hashings $\{H_{t,s}\}_{s\in[1:R_t]}$, for every $i \in S_t^t$

$$\begin{aligned}
\mathbf{Pr}_{\{H_{t,s}\}_{s\in[1:R_t]}}[i \in V_t] &= \mathbf{Pr}_{\{H_{t,s}\}_{s\in[1:R_t]}}[i \text{ is } R_t^{-3}\text{-crowded under at least one } H_{t,s}] \\
&\leq \frac{8}{C_2}(R_t^6) \cdot \frac{R_0}{R_{t-1}} 2^{-2^{(1-\delta)(t-1)}+1} \qquad \text{(by a union bound applied to (55))} \\
&\leq \frac{16C_1}{C_2}(R_t^5) \cdot 2^{-2^{(1-\delta)(t-1)}+1} \qquad \text{(since } R_t = C_1 \cdot 2^t \text{ by } \mathbf{p1}\text{)}
\end{aligned} \tag{56}$$

Using the upper bound on the size of S_t^t given by the inductive hypothesis again, we obtain

$$\begin{aligned}
\mathbf{E}_{\{H_{t,s}\}_{s\in[1:R_t]}}[|V_t|] &\leq \sum_{i\in S_t^t} \mathbf{Pr}_{\{H_{t,s}\}_{s\in[1:R_t]}}[i \in V_t] \\
&\leq |S_t^t| \cdot \mathbf{Pr}_{\{H_{t,s}\}_{s\in[1:R_t]}}[i \in V_t] \qquad \text{(for any } i \in S_t^t\text{)} \\
&\leq \left(k \cdot \frac{R_0}{R_{t-1}} 2^{-2^{(1-\delta)(t-1)}+1}\right) \cdot \mathbf{Pr}_{\{H_{t,s}\}_{s\in[1:R_t]}}[i \in V_t] \qquad \text{(by the inductive hypothesis)} \\
&\leq \left(k \cdot \frac{R_0}{R_{t-1}} 2^{-2^{(1-\delta)(t-1)}+1}\right) \cdot \frac{16C_1}{C_2} \cdot R_t^5 2^{-2^{(1-\delta)(t-1)}+1} \qquad \text{(by (56))} \\
&\leq k \cdot \frac{32C_1R_0R_t^5}{C_2 \cdot R_{t-1}} \cdot 2^{-2^{(1-\delta)(t-1)+1}+1} \\
&\leq k \cdot \frac{64C_1R_0R_t^4}{C_2} \cdot 2^{-2^{(1-\delta)t+\delta}+1} \\
&\leq k \cdot \frac{64C_1R_0R_t^4}{C_2} \cdot 2^{-2^{(1-\delta)t}+1} \cdot 2^{-(2^\delta-1)2^{(1-\delta)t}} \\
&\leq \left(k\frac{1}{100R_t^2} 2^{-2^{(1-\delta)t}+1}\right) \cdot \xi_t
\end{aligned} \tag{57}$$

where

$$\xi_t = \frac{6400C_1R_0R_t^6}{C_2} \cdot 2^{-(2^\delta-1)2^{(1-\delta)t}} \leq \frac{6400C_1^8}{C_2} 2^{6t} \cdot 2^{-\delta 2^{(1-\delta)t}},$$

where we used the assumption that $R_t = C_1 2^t$ for a constant $C_1 > 0$, and the bound $e^x - 1 \geq x$ for all $x \geq 0$.

It remains to note that for every $\delta > 0$, if C_2 is sufficiently large (depending on C_1 and δ), we get that $\xi_t < 1$ for all $t \geq 1$. Formally this follows by Claim A.2.

Bounding the number of bad elements (Bad$_t$). Recall (Definition 3.4) that an element a of S is *bad for* S_t with respect to a partition $S = S_1 \cup S_2 \cup \ldots \cup S_T$ and hashings $\{\{H_{t,s}\}_{s=1}^{R_t}\}_{t=1}^T$ if a participates in an R_t-collision with at least one element of S_t under more than a $R_t^{-\delta}$ fraction of hashings $H_{t,1}, \ldots, H_{t,R_t}$. We now upper bound the probability that a given i is bad.

For any $i \in [n]$ the probability that i R_t-collides with a given element $j \in S_t^t$ under a random hashing H is upper bounded as follows. First recall that that $\pi(i) = \sigma(i - q)$ for all $i \in [n]$, so

$$\begin{aligned}
\mathbf{Pr}_\pi[i \text{ and } j \text{ participate in an } R_t\text{-collision}] &= \mathbf{Pr}_\pi[|\pi(i) - \pi(j)|_\circ \leq (n/B_t)R_t] \\
&= \mathbf{Pr}_\sigma[|\sigma(i-j)|_\circ \leq (n/B_t)R_t] \leq 4R_t/B_t,
\end{aligned}$$

where we used Lemma 2.5 to obtain the last bound.

A union bound over all $j \in S_t^t$ then gives that for every $i \in S$

$$\mathbf{Pr}_H[i\ R_t\text{-collides with at least one element of } S_t^t \text{ under } H] \le 4R_t(k_t/B_t). \tag{58}$$

For each $s = 1, \ldots, R_t$ let $X_s = 1$ if i R_t-collides with an element of S_t^t under hashing $H_{t,s}$ and $X_s = 0$ otherwise. We first bound $\mathbf{E}[X_s]$, and then apply Chernoff bounds to $X := \sum_{s=1}^{R_t} X_s$ to bound the number of bad elements in S with respect to S_t^t at step t. In order to bound the expected number of bad elements, it would be sufficient to bound $\mathbf{Pr}[X > R_t^{1-\delta}]$. Instead, we will upper bound a slightly larger quantity that will be useful for upper bounding the expected number of elements that collide with a bad element (which is what we need to bound ultimately). Specifically, for any $s^* \in [1 : R_t]$ we let $X_{-s^*} := \sum_{s=1, s\neq s^*}^{R_t} X_s$. Note that $\mathbf{Pr}[X > R_t^{1-\delta}] \le \mathbf{Pr}[X_{-s^*} \ge R_t^{1-\delta}]$ for any s^*, and it is the latter quantity that we bound now. We now have for every $s^* \in [1 : R_t]$

$$\begin{aligned}
\mathbf{E}_{H_{t,s}}[X_{-s^*}] &\le \sum_{s=1, s\neq s^*}^{R_t} \mathbf{E}_{H_{t,s}}[X_s] \\
&\le \sum_{s=1}^{R_t} 4R_t \cdot (k_t/B_t) \quad \text{(by (58) and definition of } X_s) \\
&\le 4R_t^2 \cdot (k_t/B_t)
\end{aligned}$$

By the inductive hypothesis we have $k_t \le k \cdot \frac{R_0}{R_{t-1}} 2^{-2^{(1-\delta)(t-1)}+1}$ and $B_t \ge C_2 \cdot k/R_t^2$ by assumption **p2** of the lemma. Substituting these bounds on the last line of the equation above, we obtain

$$\begin{aligned}
&4R_t^2 \cdot \left[k \cdot \frac{R_0}{R_{t-1}} 2^{-2^{(1-\delta)(t-1)}+1}\right] \cdot \left[C_2 \cdot k/R_t^2\right]^{-1} \\
&= \frac{1}{10R_t}\left[\frac{40}{C_2} R_t^5 \cdot \frac{R_0}{R_{t-1}} \cdot 2^{-2^{(1-\delta)(t-1)}+1}\right] \\
&\le \frac{1}{10R_t}\left[\frac{40}{C_2} R_t^5 \cdot 2^{-2^{(1-\delta)(t-1)}+1}\right] \quad \text{(since } R_0/R_{t-1} = 2^{-(t-1)} \le 1 \text{ for all } t \ge 1) \\
&\le \frac{1}{10R_t} \quad \text{(by Claim A.2, as long as } C_2 \text{ is larger than a constant that may depend on } C_1 \text{ and } \delta).
\end{aligned}$$

Let $\mu_{-s^*} := \mathbf{E}\left[\sum_{s=1, s\neq s^*}^{R_t} X_s\right]$, and note that by the bound on $\mathbf{E}[X_s]$ above we have $\mu_{-s^*} \le 1/10$ (we omit the subscript in μ_{-s^*} in what follows). Since the permutations $H_{t,s}$ were chosen independently, we have by Chernoff bounds (Theorem A.1) for any $\eta > 1$ $\mathbf{Pr}\left[X_{-s^*} \ge R_t^{1-\delta}\right] = \mathbf{Pr}\left[X_{-s^*} \ge (1+\eta)\mu\right]$ with $\eta = R_t^{1-\delta}/\mu - 1$. Since $R_t^{1-\delta} > 1$ by assumption of the lemma and $\mu \le 1/10$, we have $R_t^{1-\delta}/\mu - 1 \ge R_t^{1-\delta}/(2\mu)$. We thus have

$$\mathbf{Pr}\left[X_{-s^*} \ge R_t^{1-\delta}\right] \le e^{-R_t^{1-\delta}/6}, \tag{59}$$

and by linearity of expectation

$$\mathbf{E}_{H_{t,1},\ldots,H_{t,R_t}}[|\mathrm{Bad}_t|] \le k \cdot e^{-R_t^{1-\delta}/6}. \tag{60}$$

Bounding the number of elements $i \in S_t^t$ that R_t-collide with Bad$_t$. Consider $i \in S_t^t$. For fixed $s \in [1 : R_t]$ let $Q_s(i) \subseteq S$ denote the set of elements that i R_t-collides with under hashing $H_{t,s}$. We have by (53)

$$\mathbf{E}_{H_{t,s}}[|Q_s(i)|] \le \mathbf{E}_{H_{t,s}}\left[\left|\pi(S \setminus \{i\}) \cap \mathbb{B}(\pi(i), \frac{n}{B_t} \cdot R_t)\right|\right] \le 4 \cdot R_t(k/B_t) \le (4R_t^3/C_2)$$

using assumption **p2** of the lemma.

For every $j \in Q_s(i)$ one has that j is bad only if j collides with S_t^t under $H_{t,s'}$ for at least $R_t^{1-\delta}$ values of $s' \in [1:R_t] \setminus \{s\}$. This probability is bounded by $\mathbf{Pr}[X_{-s} \geq R_t^{1-\delta}]$, where X_{-s} are as defined above. We thus have using (59) that $\mathbf{Pr}[j \text{ is bad}|i \text{ collides with } j \text{ under } H_{t,s}] \leq e^{-R_t^{1-\delta}/6}$. Note that this is where we crucially use the fact that X_{-s} does not depend on hashing $H_{t,s}$. By a union bound over all $j \in Q_s(i)$ and all $s \in [1:R_t]$ that the probability that i collides with a bad element is at most

$$\begin{aligned}\sum_{s=1}^{R_t} \mathbf{E}_{H_{t,s}}[|Q_s(i)|] \cdot e^{-R_t^{1-\delta}/6} &\leq \frac{1}{C_2} R_t \cdot (4 \cdot R_t^3) \cdot e^{-R_t^{1-\delta}/6} \\ &\leq \frac{1}{C_2} 4R_t^4 \cdot e^{-R_t^{1-\delta}/6} \\ &= e^{-R_t^{1-\delta}/12},\end{aligned} \tag{61}$$

where we used the fact that the last inequality holds for all $t \geq 1$ simultaneously as long as C_2 is larger than a constant that may depend on C_1 and δ. Summing over all elements in S_t^t, we get

$$\mathbf{E}_{H_{t,1},\ldots,H_{t,R_t}}[|U_t|] \leq k_t e^{-R_t^{1-\delta}/12} \leq k e^{-R_t^{1-\delta}/12}. \tag{62}$$

Putting it together. Gathering bounds from (57), (60) and (62), we get, using the fact that $S_{t+1}^{t+1} = \text{Bad}_t \cup U_t \cup V_t$ by Algorithm 1 (line 7),

$$\begin{aligned}\mathbf{E}_{H_{t,1},\ldots,H_{t,R_t}}\left[|S_{t+1}^{t+1}|\right] &= k \cdot \left(\frac{1}{100}\frac{1}{R_t^2} 2^{-2^{(1-\delta)t}+1} + e^{-R_t^{1-\delta}/6} + e^{-R_t^{1-\delta}/12}\right) \\ &= k \cdot \left(\frac{1}{100}\frac{1}{R_t^2} 2^{-2^{(1-\delta)t}+1} + e^{-C_1^{1-\delta}2^{(1-\delta)t}/6} + e^{-C_1^{1-\delta}2^{(1-\delta)t}/12}\right).\end{aligned} \tag{63}$$

We now show that for every $\delta \in (0, 1/2)$ if C_1 is larger than an absolute constant, the first term in parentheses on the last line above is at least as large as the other two for all $t \geq 1$. We first note that $e^{-C_1^{1-\delta}2^{(1-\delta)t}/6} \leq e^{-C_1^{1-\delta}2^{(1-\delta)t}/12}$ for all $C_1 > 0, \delta \in (0, 1/2), t \geq 1$, so it suffices to prove that

$$\frac{1}{100}\frac{1}{R_t^2} 2^{-2^{(1-\delta)t}+1} \geq e^{-C_1^{1-\delta}2^{(1-\delta)t}/12} \tag{64}$$

for all $t \geq 1$ if C_1 is larger than an absolute constant. Intuitively, this is true because the rhs decays exponentially in C_1 for all $t \geq 0$, while the lhs decays only polynomially in C_1. More formally, taking the ratio of the two quantities, we get, assuming that $C_1 \geq 12^2$,

$$\begin{aligned}e^{-C_1^{1-\delta}2^{(1-\delta)t}/12} \cdot \left(\frac{1}{100}\frac{1}{R_t^2} 2^{-2^{(1-\delta)t}+1}\right)^{-1} &\leq e^{-\sqrt{C_1}2^{(1-\delta)t}/12 + (\ln 2)2^{(1-\delta)t}} \cdot 100 \cdot R_t^2 \\ &\leq e^{-(\sqrt{C_1}/12 - \ln 2)\cdot 2^{(1-\delta)t}} \cdot 100 \cdot C_1^2 4^t \\ &\leq \exp\left(-(\sqrt{C_1}/12 - \ln 2) \cdot 2^{(1-\delta)t} + (\ln 4)t + 2\ln(100C_1)\right)\end{aligned}$$

We now show that the exponent above is non-positive for all $t \geq 1$ as long as C_1 is larger than a constant. Indeed, taking the derivative of the exponent with respect to t, we get

$$\left(-(\sqrt{C_1}/12 - \ln 2)2^{(1-\delta)t} + (\ln 2)t + \ln(100C_1)\right)' = -(\sqrt{C_1}/12 - \ln 2)(1-\delta)\ln 2 \cdot 2^{(1-\delta)t} + \ln 2,$$

which is nonpositive for all $t \geq 1$ as long as C_1 is larger than an absolute constant. This means that

$$\max_{t \geq 1}\left[-(\sqrt{C_1}/12 - \ln 2)2^{(1-\delta)t} + (\ln 2)t + \ln(100C_1)\right] \leq -(\sqrt{C_1}/12 - \ln 2)2^{1-\delta} + \ln 2 + \ln(100C_1) \leq 0$$

as long as C_1 is larger than an absolute constant (since $\sqrt{C_1}$ asympotitcally dominates $\ln C_1$). This establishes (64).

Substituting this upper bound into (63), we get

$$\mathbf{E}_{H_{t,1},\ldots,H_{t,R_t}}\left[|S_{t+1}^{t+1}|\right] \leq k \cdot \frac{3}{100}\frac{1}{R_t^2}2^{-2^{(1-\delta)t}+1}$$

By Markov's inequality applied to the last expression above we have

$$\mathbf{Pr}_{H_{t,1},\ldots,H_{t,R_t}}\left[|S_{t+1}^{t+1}| > k \cdot \frac{1}{R_t}2^{-2^{(1-\delta)t}+1}\right] \leq \frac{3}{100}\frac{1}{R_t}. \tag{65}$$

Let $\mathcal{E}_t$ denote the intersection of $\mathcal{E}_{t-1}$ with the failure event in (65), we get, conditioned on $\mathcal{E}_t$

$$|S_{t+1}^{t+1}| \leq k \cdot \frac{1}{R_t}2^{-2^{(1-\delta)t}+1}.$$

This completes the inductive step and the proof of the lemma. □

Proof of Lemma 3.9: First note that there exists a (simple) efficient data structure for answering queries of the form 'how many elements of a set T hash within circular distance Δ of a point x under hash function π?'. Indeed, it suffices to cut the circle into two halves and for each of the halves construct a binary search tree on T, with each node annotated with the number of nodes in its subtree. Then each query about neighbors in the circular distance can be answered by answering at most two queries for the two data structure on the half-circles, for a total time of $O(\log|T|)$. We now show how to use this data structure to implement Algorithm 1.

For each t at the beginning of the t-th iteration of Algorithm 1 for each $s = 1, \ldots, R_t$ one constructs two binary search trees on the permuted elements $\pi_{t,s}(S_t^t)$ as described above. This takes time $O(R_t|S_t^t|\log|S_t^t|)$. Equipped with this data structure, we can construct the set Bad_t and the set V_t in time $O(R_t \cdot S\log|S_t^t|)$. Then construct a similar pair of augmented binary search trees for the set Bad_t in time $O(|S|\log|S|)$. Using this data structure the set U_t can be constructed in time $O(R_t|S|\log|S|)$. Summing over t gives the final result. □

B Properties of ESTIMATEVALUES

In this section we describe the procedure ESTIMATEVALUES from [Kap16] (see Algorithm 4), which, given access to a signal x in frequency domain (i.e. given $\widehat{x}$), a partially recovered signal χ and a target list of locations $L \subseteq [n]$, estimates values of the elements in L, and outputs the elements that are above a threshold ν in absolute value. We need a slight strengthening of Lemma 9.1 from [Kap16], which we state here.

We will use

Definition B.1. *For any $x \in \mathbb{C}^{[n]}$ and any hashing $H = (\pi, B, F)$ define the vector $\mu^2_{H,\cdot}(x) \in \mathbb{R}^{[n]}$ by letting for every $i \in [n]$ $\mu^2_{H,i}(x) := |G^{-2}_{o_i(i)}|\sum_{j\in[n]\setminus\{i\}}|x_j|^2|G_{o_i(j)}|^2$.*

The following properties of HASHTOBINS will be using in the analysis of ESTIMATEVALUES:

Lemma B.2 (Lemma 2.9 of [Kap16])**.** *There exists a constant $C > 0$ such that for any integer $B \geq 1$, any $x, \chi \in \mathbb{C}^{[n]}, x' := x - \chi$, if $\sigma, a \in [n]$, σ odd, are selected uniformly at random, the following conditions hold for every $q \in [n]$.*

Let $\pi = (\sigma, q)$, $H = (\pi, B, F)$, where G is the filter with B buckets and sharpness F as per Definition 2.3, and let $u =$ HASHTOBINS$(\hat{x}, \chi, (H, a))$. Then if $F \geq 2$, for any $i \in [n]$

(1) *For any H one has $\max_{a\in[n]}|G^{-1}_{o_i(i)}\omega^{-a\sigma i}u_{h(i)} - x'_i| \leq G^{-1}_{o_i(i)} \cdot \sum_{j\in S\setminus\{i\}} G_{o_i(j)}|x'_j|$. Furthermore, $\mathbf{E}_H[G^{-1}_{o_i(i)} \cdot \sum_{j\in S\setminus\{i\}} G_{o_i(j)}|x'_j|] \leq C||x'||_1/B + n^{-\Omega(c)}$;*

Algorithm 4 ESTIMATEVALUES$(\chi, L, \{(H_r, a_r, m(x, H_r, a_r)))\}_{r=1}^{r_{max}})$

```
procedure ESTIMATEVALUES$(\chi, L, \{(H_r, a_r, m(x, H_r, a_r)))\}_{r=1}^{r_{max}})$    ▷ $H_r$ are $(\pi_r, B, F)$-hashings
    for $r = 1$ to $r_{max}$ do    ▷ $\pi_r = (\sigma_r, q_r)$ for $r = 1, \ldots, r_{max}$
        Compute $m_j(x - \chi, H_r, a_r)$ for $j \in [B]$    ▷ Computation is done with polynomial precision,
                                                              ▷ using HASHTOBINS as per Lemma 2.8
    end for
    for $i \in L$ do
        for $r = 1$ to $r_{max}$ do    ▷ Note that $o_i(i)$ implicitly depends on $H_r$
            $w_i^r \leftarrow G_{o_i(i)}^{-1} m_{h_r(i)}(x - \chi, H_r, a_r)\omega^{-a_r \sigma_r i}$    ▷ Estimate $(x - \chi)_i$ from each measurement
        end for
        $w_i \leftarrow$ median$\{w_i^r\}_{r=1}^{r_{max}}$    ▷ Median is taken coordinatewise
    end for
    return $w_L$
end procedure
```

(2) $\mathbf{E}_H[\mu_{H,i}^2(x')] \leq C\|x'\|_2^2/B$,

Furthermore,

(3) *for any hashing H, if a is chosen uniformly at random from $[n]$, one has*

$$\mathbf{E}_a[|G_{o_i(i)}^{-1}\omega^{-a\sigma i}u_{h(i)} - x_i'|^2] \leq \mu_{H,i}^2(x') + n^{-\Omega(c)}.$$

Here $c > 0$ is an absolute constant that can be chosen arbitrarily large at the expense of increasing runtime by a factor of c.

We note that Lemma B.2 was proved in [Kap16] for a slightly different choice of filter G and for a uniformly random q, but the proof carries over directly to our setting.

Lemma 2.11 (Restated; bounds on estimation quality for Algorithm 4) *For every $x, \chi \in \mathbb{C}^n$, every $L \subseteq [n]$, every set $S \subseteq [n]$ the following conditions hold for functions e^{head} and e^{tail} are defined with respect to S (see* (7) *and* (9)*). If r_{max} is larger than an absolute constant, then for every sequence $H_r, r = 1, \ldots, r_{max}$ of (π_r, B, F) hashings the output w of*

$$\text{ESTIMATEVALUES}(\chi, L, \{(H_r, a_r, m(x, H_r, a_r)))\}_{r=1}^{r_{max}})$$

satisfies, for each $i \in L$

$$|w_i - (x - \chi)_i| \leq 2 \cdot \textit{quant}_r^{1/5} e_i^{head}(H_r, x - \chi) + 2 \cdot \textit{quant}_r^{1/5} e_i^{tail}(H_r, a_r, x - \chi) + n^{-\Omega(c)}.$$

The sample complexity is bounded by $O(FBr_{max})$. The runtime is bounded by $O((F \cdot B \cdot \log n + ||\chi||_0 \log n + |L|) \cdot r_{max})$.

Proof. We have by definition of the measurements m_j (see Definition 2.7) for every hashing H and $a \in [n]$

$$m_{h(i)} = \sum_{j \in [n]} G_{o_i(j)}(x - \chi)_j \omega^{a\sigma j},$$

so

$$G_{o_i(i)}^{-1} m_{h(i)} \omega^{-a\sigma i} = (x - \chi)_i + G_{o_i(i)}^{-1} \sum_{j \in [n] \setminus \{i\}} G_{o_i(j)}(x - \chi)_j \omega^{a\sigma(j-i)}.$$

We thus have by triangle inequality, splitting the rhs into the contribution of the 'head' elements (i.e., elements in S) and 'tail' elements (i.e. elements in $[n] \setminus S$), that

$$\begin{aligned}
|G_{o_i(i)}^{-1} m_{h(i)} \omega^{-a\sigma i} - (x-\chi)_i| &\le \left| G_{o_i(i)}^{-1} \sum_{j \in [n] \setminus \{i\}} G_{o_i(j)} (x-\chi)_j \omega^{a\sigma(j-i)} \right| \\
&\le \left| G_{o_i(i)}^{-1} \sum_{j \in S \setminus \{i\}} G_{o_i(j)} (x-\chi)_j \omega^{a\sigma(j-i)} \right| \\
&+ \left| G_{o_i(i)}^{-1} \sum_{j \in [n] \setminus (S \cup \{i\})} G_{o_i(j)} (x-\chi)_j \omega^{a\sigma(j-i)} \right| \\
&\le G_{o_i(i)}^{-1} \sum_{j \in S \setminus \{i\}} G_{o_i(j)} |(x-\chi)_j| + G_{o_i(i)}^{-1} \left| \sum_{j \in [n] \setminus (S \cup \{i\})} G_{o_i(j)} (x-\chi)_j \omega^{a\sigma(j-i)} \right| \\
&+ |\Delta_{(n/B) \cdot h(i)}| \\
&= e_i^{head}(x-\chi, H) + e_i^{tail}(x-\chi, H, a) + |\Delta_{(n/B) \cdot h(i)}|
\end{aligned}$$

We now use the bound above to obtain the conclusion of the lemma. Recall that for each $i \in L$ the final estimate w_i is computed as a median of w_i^r's along real and imaginary axes in line 9 of Algorithm 4. Let $r' \in [1 : r_{max}]$ and $r'' \in [1 : r_{max}]$ be such that

$$w_i = \text{Re}(w_i^{r'}) + \text{Im}(w_i^{r''}) \cdot \mathbf{i}.$$

We have

$$|\text{Re}(w_i^{r'} - (x-\chi)_i)| \le |w_i^{r'} - (x-\chi)_i| \le e_i^{head}(H_{r'}, x-\chi) + e_i^{tail}(H_{r'}, a_{r'}, x-\chi), \tag{66}$$

and since r' is the result of taking the median of the list $\{\text{Re}(w_i^{r'})\}$, we have

$$|\text{Re}(w_i^{r'} - (x-\chi)_i)| \le \text{quant}_r^{1/5} e_i^{head}(H_r, x-\chi) + \text{quant}_r^{1/5} e_i^{tail}(H_r, a_r, x-\chi).$$

Indeed, at most a $2/5$ fraction of the error terms on the rhs of (66), namely $e_i^{head}(H_{r'}, x-\chi)+e_i^{tail}(H_{r'}, a_{r'}, x-\chi)$, are larger than $\text{quant}_r^{1/5} e_i^{head}(H_r, x-\chi) + \text{quant}_r^{1/5} e_i^{tail}(H_r, a_r, x-\chi)$. These error terms correspond to either the bottom or the top of the list $\{\text{Re}(w_i^{r'})\}$, and since $2/5 < 1/2$, the median estimate satisfies the upper bound above.

A similar argument for the imaginary part shows that

$$|\text{Im}(w_i^{r'} - (x-\chi)_i)| \le \text{quant}_r^{1/5} e_i^{head}(H_r, x-\chi) + \text{quant}^{1/5} e_i^{tail}(H_r, a_r, x-\chi).$$

Putting the two estimates together and using the bound $|a + b \cdot \mathbf{i}| \le |a| + |b|$, we get for each $i \in L$

$$|w_i - (x-\chi)_i| \le 2 \cdot \text{quant}_r^{1/5} e_i^{head}(H_r, x-\chi) + 2 \cdot \text{quant}_r^{1/5} e_i^{tail}(H_r, a_r, x-\chi) + n^{-\Omega(c)}$$

as required.

The sample complexity follows by Lemma 2.8.

The runtime analysis is as follows:

- Computing $m_j(x-\chi, H_r, a_r)$ for $j \in [B]$ and $r = 1, \ldots, r_{max}$ takes $O((FB \log B + ||\chi||_0 \log n) r_{max})$ time Lemma 2.8.

- Computing estimates for each $i \in L$. This takes time $|L| \cdot r_{max}$ since median can be found in linear time.

□

Theorem B.3 (Chernoff bound)**.** *Let $X_1, \ldots, X_n$ be independent $0/1$ Bernoulli random variables with $\sum_{i=1}^n \mathbf{E}[X_i] = \mu$. Then for any $\delta > 0$ one has $\mathbf{Pr}[\sum_{i=1}^n X_i > (1+\delta)\mu] < e^{(\delta-(1+\delta)\ln(1+\delta))\mu}$.*

We will use

Lemma B.4. *Let $X_1, \ldots, X_n \geq 0$ be independent random variables with $\mathbf{E}[X_i] \leq \mu$ for each $i = 1, \ldots, n$. Then for any $\gamma \in (0,1)$ if $Y \leq quant^{\gamma}(X_1, \ldots, X_n)$, then*

(1) $\mathbf{E}[|Y - 4\mu/\gamma|_+] \leq (\mu/\gamma) \cdot 2^{-\Omega(\gamma n)}$;

(2) $\mathbf{E}[|Y - 4\mu/\gamma|_+^2] \leq (\mu/\gamma)^2 \cdot 2^{-\Omega(\gamma n)}$;

(3) $\mathbf{Pr}[Y \geq 4\mu/\gamma] \leq 2^{-\Omega(\gamma n)}$;

(4) *For every $t \geq 1$ one has*
$$\mathbf{Pr}[Y \geq t\mu/\gamma] \leq (0.99t/e)^{-0.99\gamma n}.$$

Proof. For any $t \geq 1$ by Markov's inequality $\mathbf{Pr}[X_i > t\mu/\gamma] \leq \gamma/t$. Define indicator random variables Z_i by letting $Z_i = 1$ if $X_i > t\mu/\gamma$ and $Z_i = 0$ otherwise. Note that

$$\mathbf{E}[Z_i] \leq \gamma/t$$

for each i. Then since Y is bounded above by the γn-th largest of $\{X_i\}_{i=1}^n$, we have $\mathbf{Pr}[Y > t\mu/\gamma] \leq \mathbf{Pr}[\sum_{i=1}^n Z_i \geq \gamma n]$. Let $\nu := \sum_{i=1}^n \mathbf{E}[Z_i]$. We now apply the Chernoff bound (Theorem B.3) with $\delta = \gamma' n/\nu - 1$, $\gamma' = 0.99\gamma$, to the sequence $Z_i, i = 1, \ldots, n$. Note that by our setting of δ we have $(1+\delta)\nu = \gamma' n$, so

$$\begin{aligned}
\mathbf{Pr}\left[\sum_{i=1}^n Z_i > \gamma' n\right] &\leq \exp\big((\gamma' n/\nu - 1 - (\gamma' n/\nu)\ln(\gamma' n/\nu))\nu\big) \\
&= \exp\big(\gamma' n - \nu - \gamma' n \ln(\gamma' n/\nu)\big) \\
&\leq \exp\big(\gamma' n(1 - \ln(\gamma' n/\nu))\big) \\
&\leq \exp\big(\gamma' n(1 - \ln((\gamma'/\gamma)t))\big) \qquad \text{(since } \nu \leq n\gamma/t\text{)} \\
&= e^{\gamma' n}((\gamma'/\gamma)t)^{-\gamma' n}.
\end{aligned}$$

We thus get
$$\mathbf{Pr}\left[\sum_{i=1}^n Z_i \geq \gamma n\right] \leq \mathbf{Pr}\left[\sum_{i=1}^n Z_i > \gamma' n\right] \leq (0.99t/e)^{-0.99\gamma n}. \tag{67}$$
This proves **(4)**. Letting $t = 4$ in the bound above proves **(3)**.

For **(1)** we have, as long as n is sufficiently large (depending on γ),

$$\begin{aligned}
\mathbf{E}[Y \cdot \mathbf{1}_{Y \geq 4\cdot\mu/\gamma}] &\leq \int_4^\infty t\mu \cdot \mathbf{Pr}[Y \geq t \cdot \mu/\gamma] dt \\
&\leq \int_4^\infty t\mu (0.99t/e)^{-0.99\gamma n} dt \qquad \text{(by (67))} \\
&\leq e^{-\gamma n/4} \int_4^\infty t\mu (0.99t/e)^{-\gamma n/4} dt \\
&= O(\mu \cdot e^{-\gamma n/4}).
\end{aligned}$$

For **(2)** we have, as long as n is sufficiently large (depending on γ),

$$\begin{aligned} \mathbf{E}[Y \cdot \mathbf{1}_{Y \geq 4 \cdot \mu/\gamma}] &\leq \int_4^\infty t^2 \mu^2 \cdot \mathbf{Pr}[Y \geq t \cdot \mu/\gamma] dt \\ &\leq \int_4^\infty t^2 \mu^2 (0.99t/e)^{-0.99\gamma n} dt \qquad \text{(by (67))} \\ &\leq e^{-\gamma n/4} \int_4^\infty t^2 \mu (0.99t/e)^{-\gamma n/4} dt \\ &= O(\mu^2 \cdot e^{-\gamma n/4}). \end{aligned}$$

as required.

□

We also have

Lemma B.5. *For every $x \in \mathbb{C}^n$, every $S \subseteq [n]$, every $i \in [n]$, every integer r_{max} larger than an absolute constant, integers B, F with B a power of two and $F \geq 2$, the following conditions are satisfied for a sequence of random hashings $H_r = (\pi_r, B, F)$, and random evaluation points a_r, $r = 1, 2, \ldots, r_{max}$.*

If $Z^{head} := e_i^{head}(\{H_r\}, x) = quant_r^{1/5} e_i^{head}(H_r, x)$ (as per (8)) and $Z^{tail} := e_i^{tail}(\{H_r, a_r\}, x) = quant_r^{1/5} e_i^{tail}(H_r, a_r, x)$ (as per (10)), where e^{head} and e^{tail} are defined with respect to the set S, one has

(1) $\mathbf{E}_{\{H_r\}}\left[(Z^{head})^2\right] = O\left(\left(\frac{1}{B}||x_S||_1\right)^2\right)$*;*

(2) $\mathbf{E}_{\{H_r, a_r\}}\left[(Z^{tail})^2\right] = O(||x_{[n]\setminus S}||_2^2/B)$*;*

(3) $\mathbf{Pr}_{\{H_r\}}\left[Z^{head} > O\left(\frac{1}{B}||x_S||_1\right)\right] = 2^{-\Omega(r_{max})}$*;*

(4) $\mathbf{Pr}_{\{H_r\}}\left[Z^{tail} > O(||x_{[n]\setminus S}||_2/\sqrt{B})\right] = 2^{-\Omega(r_{max})}$.

Proof. We have $Z^{head} \leq |Z^{head} - 40\mathbf{E}[Z^{head}]|_+ + 40\mathbf{E}[Z^{head}]$, so, since $(a+b)^2 \leq 2a^2 + 2b^2$ for all $a, b \in \mathbb{R}$,

$$\mathbf{E}\left[(Z^{head})^2\right] \leq 2\mathbf{E}\left[(|Z^{head} - 40\mathbf{E}[Z^{head}]|_+)^2\right] + 2(40\mathbf{E}[Z^{head}])^2.$$

By Lemma B.4, **(2)** we have $\mathbf{E}\left[(|Z^{head} - 40\mathbf{E}[Z^{head}]|_+)^2\right] = O((\mathbf{E}[Z^{head}])^2)$ as long as r_{max} is larger than an absolute constant, as assumed by the lemma. An application of Lemma B.2, **(1)** now gives the first bound. The proof of the second claim is analogous using Lemma B.2, **(2)**.

Claims **(3)** and **(4)** follow similarly using Lemma B.4. □

B.1 Properties of HASHTOBINS

Algorithm 5 Hashing using Fourier samples (analyzed in Lemma 2.8)

```
procedure HashToBins($\widehat{x}, \chi, (H, a)$)                ▷ Hashing $H = (\pi, B, F)$, $a \in [n]$
    Compute $y' = \widehat{G} \cdot P_{\sigma,a,q}(\hat{x} - \hat{\chi}')$, for some $\chi'$ with $\|\widehat{\chi} - \widehat{\chi}'\|_\infty < \|\chi\|_2 \cdot n^{-c}$   ▷ Using Lemma E.1 with $\delta = n^{-2c}$, $c \geq 2$
    Compute $u_j = \sqrt{n}\mathcal{F}^{-1}(y')_{(n/B) \cdot j}$ for $j \in [B]$
    return $u$
end procedure
```

The main lemma about the performance of HASHTOBINS is

Lemma 2.8 (Restated) HASHTOBINS*($\widehat{x}, \chi, (H, a)$), where $H = (\pi, B, F)$, computes $u \in \mathbb{C}^B$ such that for any $i \in [n]$, $u_{h(i)} = \Delta_{h(i)} + \sum_j G_{o_i(j)}(x-\chi)_j \omega^{a\sigma j}$, where G is the filter defined in section 2, and for all $i \in [n]$ we have that $\Delta_{h(i)}^2 \le \|\chi\|_2^2 \cdot n^{-c}$ is a negligible error term (and $c > 0$ is an absolute constant that governs the precision that semi-equispaced FFT, i.e. Lemma E.1, is invoked with). It takes $O(BF)$ samples, and $O(F \cdot B \log B + \|\chi\|_0 \log n)$ time.*

Proof. The **first step** (line 2) in HASHTOBINS is to compute

$$y' = \widehat{G} \cdot P_{\sigma,a,q}\widehat{x-\chi'} = \widehat{G} \cdot P_{\sigma,a,q}\widehat{x-\chi} + \widehat{G} \cdot P_{\sigma,a,q}\widehat{\chi-\chi'},$$

for an approximation $\widehat{\chi}'$ to $\widehat{\chi}$ obtained using Lemma E.1, **(b)**. We now verify the runtime and precision guarantees. Recall that $\operatorname{supp}\widehat{G} \subseteq [-O(FB), O(FB)]$ by Lemma 2.4. This means that it is sufficient to compute $\widehat{\chi}_i$ on the set $S \subseteq [n]$ defined as $S = \{i \in [n] : \sigma(i-a) \in [-O(FB), O(FB)]\}$. By Lemma E.1, **(b)**, an approximation $\widehat{\chi}'$ to $\widehat{\chi}$ can be computed in $O(F \cdot B \log n)$ time such that

$$|\widehat{\chi}_i - \widehat{\chi}'_i| < \|\chi\|_2 \cdot n^{-2c}$$

for all such i. Since $\|\widehat{G}\|_1 \le \sqrt{n}\|\widehat{G}\|_2 = \sqrt{n}\|G\|_2 \le n\|G\|_\infty \le n$ and $\widehat{G}$ is 0 outside S, this implies that

$$\|\widehat{G} \cdot P_{\sigma,a,q}(\widehat{\chi-\chi'})\|_2 \le \|\widehat{G}\|_1 \max_{j \in S} |\widehat{\chi-\chi'}_i| \le \|\chi\|_2 \cdot n^{1-2c}. \tag{68}$$

Define Δ by $\widehat{\Delta} = \sqrt{n}\widehat{G} \cdot P_{\sigma,a,q}(\widehat{\chi-\chi'})$.

The **second step** (line 3) in HASHTOBINS is to compute $u \in \mathbb{C}^B$ such that for all i,

$$u_{h(i)} = \sqrt{n}\mathcal{F}^{-1}(y')_{(n/B)\cdot h(i)} = \sqrt{n}\mathcal{F}^{-1}(y)_{(n/B)\cdot h(i)} + \Delta_{(n/B)\cdot h(i)},$$

for $y = \widehat{G} \cdot P_{\sigma,a,q}\widehat{x-\chi}$. This computation takes $O(\|y'\|_0 + B \log B) = O(FB \log n)$ time (alias y' to length B and compute a length B FFT). We have by the convolution theorem (see (81)) that

$$\begin{aligned}
u_{h(i)} &= \sqrt{n}\mathcal{F}^{-1}(\widehat{G} \cdot P_{\sigma,a,q}(\widehat{x-\chi}))_{(n/B)\cdot h(i)} + \Delta_{(n/B)\cdot h(i)} \\
&= (G * \mathcal{F}^{-1}(P_{\sigma,a,q}(\widehat{x-\chi})))_{(n/B)\cdot h(i)} + \Delta_{(n/B)\cdot h(i)} \\
&= \sum_{\pi(j) \in [n]} G_{(n/B)\cdot h(i) - \pi(j)} \mathcal{F}^{-1}(P_{\sigma,a,q}(\widehat{x-\chi}))_{\pi(j)} + \Delta_{(n/B)\cdot h(i)} \\
&= \sum_{j \in [n]} G_{o_i(j)}(x-\chi)_j \omega^{a\sigma j} + \Delta_{(n/B)\cdot h(i)}
\end{aligned}$$

where the last step is the definition of $o_i(j)$ and Lemma 2.2.

Finally, we note that

$$|\Delta_{(n/B)\cdot h(i)}| \le \|\Delta\|_2 = \|\widehat{\Delta}\|_2 = \sqrt{n}\|\widehat{G} \cdot P_{\sigma,a,q}(\widehat{\chi-\chi'})\|_2 \le \|\chi\|_2 n^{3/2-2c} \le \|\chi\|_2 n^{-c},$$

where we used (68) and the assumption that $c \ge 2$ in the last step. This completes the proof. □

The following lemma is analogous to Lemma 9.4 of [IKP14], but does not make the assumption that the number of repetitions involved in the quantile operation is a constant. The proof is essentially the same, but is given below for completeness.

Lemma B.6. *For every $\gamma \in (0,1)$, integers $m, n \geq 1$ such that $n > 4/\gamma$, every sequence $X^1, \ldots, X^n \in \mathbb{R}_+^m$ of random variables with non-negative entries such that $X^j \in \mathbb{R}_+^n$ are independent, $\mathbf{E}[X_i^j] \leq \nu$ for every $i = 1, \ldots, m, j = 1, \ldots, n$ and $\nu > 0$, the following conditions hold. If for every $i = 1, \ldots, m$*

$$Y_i = \textit{quant}^\gamma(X^1, \ldots, X^n),$$

then for every U between 1 and m

$$\mathbf{E}\left[\max_{Q \subseteq [m], |Q| \leq U} \sum_{i \in Q} Y_i\right] \leq U \cdot (4e\nu/\gamma) \cdot (m/U)^{2/(\gamma n)}$$

Note that the lemma assumes that $X^j \in \mathbb{R}_+^n$ are independent, but allows for the coordinates of each X^j to be arbitrarily correlated.

Proof. First fix $i \in \{1, 2, \ldots, m\}$. By Lemma B.4, **(4)** we have for every $t \geq 1$

$$\mathbf{Pr}[Y \geq t\nu/\gamma] \leq (0.99t/e)^{-0.99\gamma n} \leq (t/(2e))^{-\gamma n/2}.$$

We thus have for $t \geq 1$

$$\mathbf{E}[|\{i : Y_i \geq t\nu/\gamma\}|] \leq m \cdot (t/(2e))^{-\gamma n/2}, \tag{69}$$

and hence for every threshold $\theta > 0$ one has

$$\begin{aligned}
\mathbf{E}\left[\max_{Q \subseteq [m], |Q| \leq U} \sum_{i \in Q} Y_i\right] &\leq \mathbf{E}\left[\max_{Q \subseteq [m], |Q| = U} \sum_{i \in Q} Y_i\right] && \text{(since } Y_i \geq 0 \text{ for all } i\text{)} \\
&= \mathbf{E}\left[\int_0^\infty \min(U, |\{i : Y_i > \eta\}|) d\eta\right] \\
&\leq \int_0^\infty \min(U, \mathbf{E}\left[|\{i : Y_i > \eta\}|\right]) d\eta && \text{(by convexity of } \min(U, x) \text{ as a function of } x\text{)} \\
&\leq U \cdot \theta + \int_\theta^\infty \min(U, \mathbf{E}\left[|\{i : Y_i > \eta\}|\right]) d\eta \\
&\leq U \cdot \theta + \int_\theta^\infty \min(U, m \cdot (\eta\gamma/(2e\nu))^{-\gamma n/2}) d\eta && \text{(by (69), as long as } \theta \geq \nu/\gamma\text{)} \\
&\leq U \cdot \theta + \int_\theta^\infty m \cdot (\eta\gamma/(2e\nu))^{-\gamma n/2} d\eta \\
&\leq U \cdot \theta + m \cdot (\gamma/(2e\nu))^{-\gamma n/2} \int_\theta^\infty \eta^{-\gamma n/2} d\eta \\
&\leq U \cdot \theta + m \cdot (\gamma/(2e\nu))^{-\gamma n/2} \frac{1}{\gamma n/2 - 1} \theta^{-\gamma n/2 + 1} \\
&\leq U \cdot \theta + m \cdot (\theta \cdot \gamma/(2e\nu))^{-\gamma n/2} \cdot \theta && \text{(since } \gamma n/2 > 2 \text{ by assumption)}
\end{aligned} \tag{70}$$

We now let $\theta = (2e\nu/\gamma) \cdot (m/U)^{2/(\gamma n)} > \nu/\gamma$, so that

$$m \cdot (\theta \cdot \gamma/(2e\nu))^{-\gamma n/2} = k,$$

and substituting into (70), we get

$$\mathbf{E}\left[\max_{Q \subseteq [m], |Q| \leq U} \sum_{i \in Q} Y_i\right] \leq 2U \cdot \theta = U \cdot (4e\nu/\gamma) \cdot (m/U)^{2/(\gamma n)},$$

as required. □

C Signal location primitive and its analysis

We reuse the location primitive from [Kap16] (LOCATESIGNAL, see Algorithm 6), but present it here with simplified notation adapted to the 1d setting (thus obviating the need for the $\star$ operation). As in [Kap16], we will use

Definition C.1 (Balanced set of points)**.** *For an integer $\Delta \geq 2$ we say that a (multi)set $\mathcal{Z} \subseteq [n]$ is* Δ-balanced *if for every $r = 1, \ldots, \Delta - 1$ at least $49/100$ fraction of elements in the set $\{\omega_\Delta^{r \cdot z}\}_{z \in \mathcal{Z}}$ belong to the left halfplane $\{u \in \mathbb{C} : \mathit{Re}(u) \leq 0\}$ in the complex plane, where $\omega_\Delta = e^{2\pi i/\Delta}$ is the Δ-th root of unity.*

We will also need

Claim C.2 (Claim 2.14 of [Kap16])**.** *There exists a constant $C > 0$ such that for any Δ a power of two, $\Delta = \log^{O(1)} n$, and n a power of 2 the following holds if $\Delta < n$. If elements of a (multi)set $\mathcal{A} \subseteq [n] \times [n]$ of size $C \log \log n$ are chosen uniformly at random with replacement from $[n] \times [n]$, then with probability at least $1 - 1/\log^4 n$ one has that the set $\{\beta\}_{(\alpha,\beta) \in \mathcal{A}}$ is Δ-balanced.*

Since we only use one value of Δ in the paper (see line 4 in Algorithm 6), we will usually say that a set is simply 'balanced' to denote the Δ-balanced property for this value of Δ. Before we state the algorithm and give the analysis, we need to introduce notation for bounding the influence of tail noise on the location process. We do this in the next section.

C.1 Analysis of LOCATESIGNAL

Algorithm 6 Location primitive: given a set of measurements corresponding to a single hash function, returns a list of elements in $[n]$, one per each hash bucket

```
 1: procedure LOCATESIGNAL($\chi, H, \{m(x, H, \alpha + \mathbf{w} \cdot \beta\}_{(\alpha,\beta) \in \mathcal{A}, \mathbf{w} \in \mathcal{W}}$)    ▷ $H = (\pi, B, F)$
 2:     Let $x' := x - \chi$. Compute $\{m(x', H, \alpha + \mathbf{w} \cdot \beta)\}_{(\alpha,\beta) \in \mathcal{A}, \mathbf{w} \in \mathcal{W}}$ using HASHTOBINS, as per Lemma 2.8.
 3:     $L \leftarrow \emptyset$
 4:     $\Delta \leftarrow 2^{\lfloor \frac{1}{2} \log_2 \log_2 n \rfloor}$
 5:     $N \leftarrow \Delta^{\lceil \log_\Delta n \rceil}$    ▷ Extend $\widehat{x}$ implicitly to $\mathbb{C}^N$ periodically
 6:     for $j \in [B]$ do    ▷ Loop over all hash buckets, indexed by $j \in [B]$
 7:         $\mathbf{f} \leftarrow 0$
 8:         for $g = 1$ to $\log_\Delta N$ do
 9:             $\mathbf{w} \leftarrow N\Delta^{-g}$    ▷ Note that $\mathbf{w} \in \mathcal{W}$
10:             If there exists a unique $r \in [0 : \Delta - 1]$ such that
11:                 $\left|\omega_\Delta^{-r \cdot \beta} \cdot \omega^{-(N \cdot \Delta^{-g} \mathbf{f} \cdot \beta} \cdot \frac{m_j(x', H, \alpha + \mathbf{w} \cdot \beta)}{m_j(x', H, \alpha)} - 1\right| < 1/3$ for at least $3/5$ fraction of $(\alpha, \beta) \in \mathcal{A}$
12:             then $\mathbf{f} \leftarrow \mathbf{f} + \Delta^{g-1} \cdot r$
13:         end for
14:         $L \leftarrow L \cup \left\{\sigma^{-1} \mathbf{f} \cdot \frac{n}{N}\right\}$    ▷ Add recovered element to output list
15:     end for
16:     return $L$
17: end procedure
```

Equipped with the definitions above, we now prove the following lemma, which yields sufficient conditions for recovery of elements $i \in S$ in LOCATESIGNAL in terms of e^{head} and e^{tail}.

Lemma C.3. *Let $H = (\pi, B, F)$ be a hashing, and let $\mathcal{A} \subseteq [n] \times [n]$. Then for every $S \subseteq [n]$ and for every $x, \chi \in \mathbb{C}^{[n]}$ and $x' = x - \chi$, the following conditions hold. Let L denote the output of*

$$\textsc{LocateSignal}(\chi, H, \{m(x, H, \alpha + \mathbf{w} \cdot \beta)\}_{(\alpha,\beta)\in\mathcal{A},\mathbf{w}\in\mathcal{W}}).$$

Then for any $i \in S$ such that $|x'_i| > n^{-\Omega(c)}$, if there exists $r \in [1 : r_{max}]$ such that

1. *$e_i^{head}(H, x') < |x'_i|/20$;*
2. *$e_i^{tail}(H, \{\alpha + \mathbf{w} \cdot \beta\}, x') < |x'_i|/20$ for all $\mathbf{w} \in \mathcal{W}$;*
3. *the set $\{\beta\}_{(\alpha,\beta)\in\mathcal{A}}$ is balanced (as per Definition C.1),*

then $i \in L$. The time taken by the invocation of LocateSignal *is $O(FB \log^2 n + ||\chi||_0 \log^2 n)$.*

Proof. Let $q = \sigma i$ for convenience. We show by induction on $g = 1, \ldots, \log_\Delta N$ that after the g-th iteration of lines 9-12 of Algorithm 6 we have that $\mathbf{f}$ coincides with $\mathbf{q}$ on the bottom $g \cdot \log_2 \Delta$ bits, i.e. $\mathbf{f} - \mathbf{q} = 0 \mod \Delta^g$ (note that we trivially have $\mathbf{f} < \Delta^g$ after iteration g).

The **base** of the induction is trivial and is provided by $g = 1$. We now show the **inductive step**. Assume by the inductive hypothesis that $\mathbf{f} - \mathbf{q} = 0 \mod \Delta^{g-1}$, so that $\mathbf{q} = \mathbf{f} + \Delta^{g-1}(r_0 + \Delta r_1 + \Delta^2 r_2 + \ldots)$ for some sequence $r_0, r_1, \ldots$, $0 \le r_j < \Delta$. Thus, $(r_0, r_1, \ldots)$ is the expansion of $(\mathbf{q} - \mathbf{f})/\Delta^{g-1}$ base Δ, and r_0 is the least significant digit. We now show that r_0 is the unique value of r that satisfies the conditions of lines 10-11 of Algorithm 6.

First, we have by (6) together with (7) and (9) one has for each $(\alpha, \beta) \in \mathcal{A}$ and $\mathbf{w} \in \mathcal{W}$

$$\left| G^{-1}_{o_i(i)} m_{h(i)}(x', H, \alpha + \mathbf{w} \cdot \beta) - x'_i \omega^{(\alpha + \mathbf{w}\cdot\beta)\mathbf{q}} \right| \le e_i^{head}(H, x') + e_i^{tail}(H, \alpha + \mathbf{w} \cdot \beta, x) + n^{-\Omega(c)}.$$

Let $j := h(i)$. We will show that i is recovered from bucket j. The bounds above imply that

$$\frac{m_j(x', H, \alpha + \mathbf{w} \cdot \beta)}{m_j(x', H, \alpha)} = \frac{x'_i \omega^{(\alpha+\mathbf{w}\cdot\beta)\mathbf{q}} + E'}{x'_i \omega^{\alpha \mathbf{q}} + E''} \tag{71}$$

for some E', E'' satisfying $|E'| \le e_i^{head}(H, x') + e_i^{tail}(H, \alpha + \mathbf{w} \cdot \beta, x) + n^{-\Omega(c)}$ and $|E''| \le e_i^{head}(H, x') + e_i^{tail}(H, \alpha) + n^{-\Omega(c)}$. For all but $1/5$ fraction of $(\alpha, \beta) \in \mathcal{A}$ we have by definition of e^{tail} (see (36)) that **both**

$$e_i^{tail}(H, \alpha + \mathbf{w} \cdot \beta, x) \le e_i^{tail}(H, \{\alpha + \mathbf{w} \cdot \beta\}, x) \le |x'_i|/20 \tag{72}$$

and

$$e_i^{tail}(H, \alpha, x) \le e_i^{tail}(H, \{\alpha\}, x) \le |x'_i|/20. \tag{73}$$

In particular, we can rewrite (71) as

$$\begin{aligned} \frac{m_j(x', H, \alpha + \mathbf{w} \cdot \beta)}{m_j(x', H, \alpha)} &= \frac{x'_i \omega^{(\alpha+\mathbf{w}\cdot\beta)\mathbf{q}} + E'}{x'_i \omega^{\alpha \mathbf{q}} + E''} \\ &= \frac{\omega^{(\alpha+\mathbf{w}\cdot\beta)\mathbf{q}}}{\omega^{\alpha\mathbf{q}}} \cdot \xi \quad \text{where } \xi = \frac{1 + \omega^{-(\alpha+\mathbf{w}\cdot\beta)\mathbf{q}} E'/x'_i}{1 + \omega^{-(\alpha)\mathbf{q}} E''/x'_i} \\ &= \omega^{(\alpha+\mathbf{w}\cdot\beta)\mathbf{q} - \alpha\mathbf{q}} \cdot \xi \\ &= \omega^{(\mathbf{w}\cdot\beta)\mathbf{q}} \cdot \xi. \end{aligned} \tag{74}$$

Let $\mathcal{A}^* \subseteq \mathcal{A}$ denote the set of values of $(\alpha, \beta) \in \mathcal{A}$ that satisfy the bounds (72) and (73) above. We thus have for $(\alpha, \beta) \in \mathcal{A}^*$, combining (74) with assumptions **1-2** of the lemma, that

$$|E'|/x'_i \le (2/20) + n^{-\Omega(c)} \le 1/8 \quad \text{and} \quad |E''|/x'_i \le (2/20) + n^{-\Omega(c)} \le 1/8 \tag{75}$$

for sufficiently large n, where $O(c)$ is the word precision of our semi-equispaced Fourier transform computation. Note that we used the assumption that $|x'_i| \geq n^{-\Omega(c)}$.

Writing $(\alpha, \beta) \in [n] \times [n]$, we have by (74) that $\frac{m_j(x',H,\alpha+\mathbf{w}\cdot\beta)}{m_j(x',H,\alpha)} = \omega^{\mathbf{w}\cdot\beta\mathbf{q}} \cdot \xi$, and since $\mathbf{wq} = n\Delta^{-g}\mathbf{q}$ when $\mathbf{w} = N\Delta^{-g}$ (as in line 8 of Algorithm 6), we get

$$\frac{m_j(x', H, \alpha + \mathbf{w}\cdot\beta)}{m_j(x', H, \alpha)} = \omega^{\mathbf{w}\cdot\beta\mathbf{q}} \cdot \xi = \omega^{n\Delta^{-g}\beta\mathbf{q}} \cdot \xi = \omega^{n\Delta^{-g}\beta\mathbf{q}} + \omega^{n\Delta^{-g}\beta\mathbf{q}}(\xi - 1).$$

We analyze the first term now, and will show later that the second term is small. Since $\mathbf{q} = \mathbf{f} + \Delta^{g-1}(r_0 + \Delta r_1 + \Delta^2 r_2 + \ldots)$ by the inductive hypothesis, we have, substituting the first term above into the expression in line 10 of Algorithm 6,

$$\begin{aligned}
\omega_\Delta^{-r\cdot\beta} \cdot \omega^{-n\Delta^{-g}\mathbf{f}\cdot\beta} \cdot \omega^{n\Delta^{-g}\beta\mathbf{q}} &= \omega_\Delta^{-r\cdot\beta} \cdot \omega^{n\Delta^{-g}(\mathbf{q}-\mathbf{f})\cdot\beta} \\
&= \omega_\Delta^{-r\cdot\beta} \cdot \omega^{n\Delta^{-g}(\Delta^{g-1}(r_0+\Delta r_1+\Delta^2 r_2+\ldots))\cdot\beta} \\
&= \omega_\Delta^{-r\cdot\beta} \cdot \omega^{(n/\Delta)\cdot(r_0+\Delta r_1+\Delta^2 r_2+\ldots)\cdot\beta} \\
&= \omega_\Delta^{-r\cdot\beta} \cdot \omega_\Delta^{r_0\cdot\beta} \\
&= \omega_\Delta^{(-r+r_0)\cdot\beta}.
\end{aligned}$$

We used the fact that $\omega^{n/\Delta} = e^{2\pi i(n/\Delta)/n} = e^{2\pi i/\Delta} = \omega_\Delta$ and $(\omega_\Delta)^\Delta = 1$. Thus, we have

$$\omega_\Delta^{-r\cdot\beta}\omega^{-(n2^{-g}\mathbf{f})\cdot\beta}\frac{m_j(x', H, \alpha + \mathbf{w}\cdot\beta)}{m_j(x', H, \alpha)} = \omega_\Delta^{(-r+r_0)\cdot\beta} + \omega_\Delta^{(-r+r_0)\cdot\beta}(\xi - 1). \tag{76}$$

We now consider two cases. First suppose that $r = r_0$. Then $\omega_\Delta^{(-r+r_0)\cdot\beta} = 1$, and it remains to note that by (75) we have $|\xi - 1| \leq \frac{1+1/8}{1-1/8} - 1 \leq 2/7 < 1/3$. Thus every $(\alpha, \beta) \in \mathcal{A}^*$ passes the test in line 11 of Algorithm 6. Since $|\mathcal{A}^*| > (3/5)|\mathcal{A}|$ by the argument above, we have that r_0 passes the test in line 11. It remains to show that r_0 is the unique element in $0, \ldots, \Delta - 1$ that passes this test.

Now suppose that $r \neq r_0$. Then by the assumption that $\{\beta\}_{(\alpha,\beta)\in\mathcal{A}}$ is balanced (assumption **3** of the lemma) at least $49/100$ fraction of $\omega_\Delta^{(-r+r_0)\cdot\beta}$ have negative real part. This means that for at least $49/100$ of $(\alpha, \beta) \in \mathcal{A}$ we have using triangle inequality

$$\begin{aligned}
\left|\left[\omega_\Delta^{(-r+r_0)\cdot\beta} + \omega_\Delta^{(-r+r_0)\cdot\beta}(\xi - 1)\right] - 1\right| &\geq \left|\omega_\Delta^{(-r+r_0)\cdot\beta} - 1\right| - \left|\omega_\Delta^{(-r+r_0)\cdot\beta}(\xi - 1)\right| \\
&\geq |\mathbf{i} - 1| - 1/3 \\
&\geq \sqrt{2} - 1/3 > 1/3,
\end{aligned}$$

and hence the condition in line 11 of Algorithm 6 is not satisfied for any $r \neq r_0$. This shows that location is successful and completes the proof of correctness.

Runtime. We perform $|\mathcal{A}| \cdot |\mathcal{W}| = O(\log n)$ invocations of HASHTOBINS in line 1 of the algorithm. Each invocation costs $O(FB\log B + ||\chi||_0 \log n)$ by Lemma 2.8, for a total runtime of $O(FB\log B\log n + ||\chi||_0 \log^2 n)$ for line 1.

After this for each of B buckets the algorithm performs decoding in blocks of $\log\Delta$ bits, amounting to $O(\log_\Delta n)$ iterations. The decoding of each block requires looping over Δ possibilities, and testing each against the evaluation points in $\mathcal{A}$. Since $|\mathcal{A}| = O(\log\log n)$. Thus the total runtime is $O(\log_\Delta n \cdot |\mathcal{A}| \cdot \Delta) = O(\Delta\log n) = O(\log^2 n)$, as $\log\Delta = \Theta(\log\log n)$ and $\Delta = O(\log n)$. The total runtime is thus $O(FB\log^2 n + ||\chi||_0 \log^2 n)$, as claimed. □

We also get an immediate corollary of Lemma C.3.

Lemma 5.3 (Restated from section 5.2 *For any integer $r_{max} \geq 1$, for any sequence of r_{max} hashings $H_r = (\pi_r, B, R), r \in [1 : r_{max}]$ and evaluation points $\mathcal{A}_r \subseteq [n] \times [n]$, for every $S \subseteq [n]$ and for every $x, \chi \in \mathbb{C}^{[n]}, x' := x - \chi$, the following conditions hold. If for each $r \in [1 : r_{max}]$ $L_r \subseteq [n]$ denotes the output of* LOCATESIGNAL*($\widehat{x}, \chi, H_r, \{m(x, H_r, \alpha + \mathbf{w} \cdot \beta)\}_{(\alpha,\beta)\in\mathcal{A}_r, \mathbf{w}\in\mathcal{W}}$), $L = \bigcup_{r=1}^{r_{max}} L_r$, and the sets $\{\beta\}_{(\alpha,\beta)\in\mathcal{A}_r}$ are balanced $r \in [1 : r_{max}]$, then*

$$||x'_{S\setminus L}||_1 \leq 20||e_S^{head}(\{H_r\}, x')||_1 + 20||e_S^{tail,\mathcal{W}}(\{H_r, \mathcal{A}_r\}, x)||_1 + |S| \cdot n^{-\Omega(c)}. \tag{*}$$

Furthermore, every element $i \in S$ such that

$$|x'_i| > 20(e_i^{head}(\{H_r\}, x') + e_i^{tail,\mathcal{W}}(\{H_r, \mathcal{A}_r\}, x)) + n^{-\Omega(c)} \tag{**}$$

belongs to L.

Proof. Suppose that $i \in S$ fails to be located in any of the R calls, and $|x'_i| \geq n^{-\Omega(c)}$. By Lemma C.3 and the assumption that the sets $\{\beta\}_{(\alpha,\beta)\in\mathcal{A}_r}$ are balanced for all $r \in [1 : r_{max}]$ this means that for at least one half of values $r \in [1 : r_{max}]$ either **(A)** $e_i^{head}(H_r, x') \geq |x'_i|/20$ or **(B)** $e_i^{tail}(H_r, \{\alpha + \mathbf{w} \cdot \beta\}_{(\alpha,\beta)\in\mathcal{A}_r}, x) > |x'_i|/20$ for at least one $\mathbf{w} \in \mathcal{W}$. We consider these two cases separately.

Case (A). In this case we have $e_i^{head}(H_s, x') \geq |x'_i|/20$ for at least one half of $r \in [1 : r_{max}]$, so in particular $e_i^{head}(\{H_r\}, x') \geq \text{quant}_r^{1/5} e_i^{head}(H_r, x') \geq |x'_i|/20$.

Case (B). Suppose that $e_i^{tail}(H_r, \{\alpha + \mathbf{w} \cdot \beta\}_{(\alpha,\beta)\in\mathcal{A}_r}, x) > |x'_i|/20$ for some $\mathbf{w} = \mathbf{w}(r) \in \mathcal{W}$ for at least one half of $r \in [1 : r_{max}]$ (denote this set by $Q \subseteq [1 : r_{max}]$). We then have

$$\begin{aligned}
e_i^{tail,\mathcal{W}}(\{H_r, \mathcal{A}_r\}, x) &= \text{quant}^{1/5}_{r\in[1:r_{max}]} e_i^{tail}(H_r, \mathcal{A}_r, x) \\
&= \text{quant}^{1/5}_{r\in[1:r_{max}]} \left[40\mu_{H_r,i}(x) + \sum_{\mathbf{w}\in\mathcal{W}} \left| e_i^{tail}(H_r, \{\alpha + \mathbf{w} \cdot \beta\}_{(\alpha,\beta)\in\mathcal{A}_r}, x) - 40\mu_{H_r,i}(x) \right|_+ \right] \\
&\geq \min_{r\in Q} \left[40\mu_{H_r,i}(x) + \left| e_i^{tail}(H_r, \{\alpha + \mathbf{w}(r) \cdot \beta\}_{(\alpha,\beta)\in\mathcal{A}_r}, x) - 40\mu_{H_r,i}(x) \right|_+ \right] \\
&\geq \min_{r\in Q} e_i^{tail}(H_r, \{\alpha + \mathbf{w}(r) \cdot \beta\}_{(\alpha,\beta)\in\mathcal{A}_r}, x) \\
&\geq |x'_i|/20
\end{aligned}$$

as required. This completes the proof of **(*)** as well as **(**)**. □

D Proof of Lemma 5.4 (tail noise error bounds)

We will use

Lemma D.1 (Lemma 6.6 of [Kap16]). *For any constant $C' > 0$ there exists an absolute constant $C > 0$ such that for any $x \in \mathbb{C}^n$, any integer $k \geq 1$ and $S \subseteq [n]$ such that $||x_{[n]\setminus S}||_\infty \leq C'||x_{[n]\setminus[k]}||/\sqrt{k}$, if $B \geq 1$, then the following conditions hold for $e^{tail,\mathcal{W}}$ defined with respect to S.*

If hashings $H_r = (\pi_r, B, F), F \geq 2$ and sets $\mathcal{A}_r, |\mathcal{A}_r| \geq c_{max}$ for $r = 1, \ldots, r_{max}$ are chosen at random, then for every $i \in [n]$ one has $\mathbf{E}_{\{(H_r,\mathcal{A}_r)\}} \left[e_i^{tail,\mathcal{W}}(\{H_r, \mathcal{A}_r\}, x) \right] \leq C(40 + |\mathcal{W}|2^{-\Omega(c_{max})})||x_{[n]\setminus[k]}||_2/\sqrt{B}$.

Note that this lemma was stated in [Kap16] with slightly different notation (e^{tail} instead of $e^{tail,\mathcal{W}}$).

Proof of Lemma 5.4: First recall that by Lemma D.1 for every $t \in [1:T]$, $s \in [1:R_t]$ and $i \in S$ one has

$$\mathbf{E}_{H_{t,s},\mathcal{A}_{t,s}}\left[e_i^{tail,\mathcal{W}}(H_{t,s},\mathcal{A}_{t,s},x)\right] = \nu^2,$$

where $\nu^2 \le C'||x_{[n]\setminus S}||_2/\sqrt{B_t}$ for an absolute constant $C' > 0$ (we used the fact that $|\mathcal{W}| = O(\log N)$ and $|\mathcal{A}| = C'' \log\log N$ for a sufficiently large absolute constant C'').

To upper bound $\mathbf{E}_{\{H_{t,s},\mathcal{A}_{t,s}\}}\left[||e_{S_t}^{tail,\mathcal{W}}(\{H_{t,s},\mathcal{A}_{t,s}\},x)||_1\right]$, we note that by conditioning on $\mathcal{E}_{partition}$ we have $|S_t| \le 2k\frac{R_0}{R_{t-1}}2^{-2^{(1-\delta)(t-1)}+1}$. Letting $U := 2k\frac{R_0}{R_{t-1}}2^{-2^{(1-\delta)(t-1)}+1}$ to simplify notation, we get that

$$\mathbf{E}_{\{H_{t,s},\mathcal{A}_{t,s}\}}\left[||e_{S_t}^{tail,\mathcal{W}}(H_{t,s},\mathcal{A}_{t,s},x)||_1\right] \le \mathbf{E}\left[\max_{Q\subseteq S,|Q|\le U}||e_Q^{tail,\mathcal{W}}(H_{t,s},\mathcal{A}_{t,s},x)||_1\right] \tag{77}$$

We now recall that by (38)

$$e_i^{tail,\mathcal{W}}(\{H_{t,s},\mathcal{A}_{t,s}\},x) := \text{quant}^{1/5}_{s=1,\dots,R_t} e_i^{tail,\mathcal{W}}(H_{t,s}\mathcal{A}_{t,s},x),$$

and apply Lemma B.6 with $\gamma = 1/5$, $m = |S|$, $n = R_t$ and

$$X_i^s = e_i^{tail,\mathcal{W}}(H_{t,s}\mathcal{A}_{t,s},x) \quad \text{for } i \in S \text{ and } s = 1,\dots,R_t,$$

so that $\mathbf{E}_{H_{t,s},\mathcal{A}_{t,s}}[X_i^s] \le \nu$ for each $i \in S$, $s = 1,\dots,R_t$. Note that $Y_i := \text{quant}^{1/5}_{s=1,\dots,R_t}X_i^s = e_i^{tail}(\{H_{t,s},z_{t,s}\},x)$ is exactly the quantity that we are interested in. We thus have by Lemma B.6

$$\begin{aligned}\mathbf{E}_{\{H_{t,s},\mathcal{A}_{t,s}\}}\left[\max_{Q\subseteq S,|Q|\le U}||e_Q^{tail,\mathcal{W}}(H_{t,s},\mathcal{A}_{t,s},x)||_1\right] &= \mathbf{E}_{\{H_{t,s},\mathcal{A}_{t,s}\}}\left[\max_{Q\subseteq S,|Q|\le U}\sum_{i\in Q}Y_i\right]\\ &\le U\cdot(20e\nu)\cdot(|S|/U)^{10/R_t}\end{aligned} \tag{78}$$

Since $R_{t'} = C_1 2^{t'}$ for every t', $|S| = |S_0| \le 2k$ and $U = 2k\frac{R_0}{R_{t-1}}2^{-2^{(1-\delta)(t-1)}+1} = 2k2^{-2^{(1-\delta)(t-1)}+1-(t-1)}$, we have

$$(|S|/U)^{10/R_t} = 2^{10(2^{(1-\delta)(t-1)}-1+(t-1))/(C_1 2^t)} \le 2^{10(1+(t-1)/2^t)/C_1} \le 2^{20/C_1} \le 2$$

for all $t \ge 0$ as long as $C_1 > 20$. Substituting the above into (78), we get

$$\mathbf{E}\left[\max_{Q\subseteq S,|Q|\le U}||e_Q^{tail}(\{H_{t,s},a_{t,s}\}_{s\in[1:R_t]},x)||_1\right] \le (40e)\cdot U\cdot\nu.$$

We thus get by combining the above with (77)

$$\mathbf{E}_{H_{t,s},\mathcal{A}_{t,s}}\left[||e_{S_t}^{tail,\mathcal{W}}(H_{t,s},\mathcal{A}_{t,s},x)||_1\right] = (40eC')U||x_{[n]\setminus[k]}||_2/\sqrt{B_t}.$$

We now use assumption **q2** of the lemma to upper bound

$$\begin{aligned}(40eC')U/\sqrt{B_t} &\le (40eC')\left(2k\frac{R_0}{R_{t-1}}\cdot 2^{-2^{(1-\delta)(t-1)}+1}\right)/\sqrt{C_2 2k/R_t^2}\\ &\le \frac{\sqrt{k}}{R_{t-1}}\cdot\frac{1}{R_t}\cdot\left(\frac{(80eC')C_1}{\sqrt{2C_2}}R_t^2\cdot 2^{-2^{(1-\delta)(t-1)}+1}\right)\\ &= \frac{\sqrt{k}}{R_{t-1}}\cdot\frac{1}{R_t}\cdot\left(\frac{(80eC')C_1^2}{\sqrt{2C_2}}2^{2t}\cdot 2^{-2^{(1-\delta)(t-1)}+1}\right)\\ &\le \frac{1}{2}\frac{\sqrt{k}}{R_{t-1}}\cdot\frac{1}{R_t}\end{aligned} \tag{79}$$

as long as C_2 is sufficiently large as a function of C' and C_1 (to ensure that $\frac{(80eC')C'C_1^2}{\sqrt{2C_2}} 2^{2t} 2^{-2^{(1-\delta)(t-1)}} \le 1$ for all $t \geq 1$; see Claim A.2). Substituting this bound into the upper bound on the expectation above yields $\mathbf{E}_{\{H_{t,s},\mathcal{A}_{t,s}\}}\left[||e_{S_t}^{tail,\mathcal{W}}(\{H_{t,s},\mathcal{A}_{t,s}\},x)||_1\right] \le \frac{1}{R_t}\cdot ||x_{[n]\setminus[k]}||_2\sqrt{k}/R_{t-1}$ for every $t \geq 1$. It now follows by Markov's inequality that for every $t \geq 1$

$$\mathbf{Pr}_{\{H_{t,s}\}_{s\in[1:R_t]}}\left[||e_{S_t}^{tail,\mathcal{W}}(\{H_{t,s},\mathcal{A}_{t,s}\},x)||_1 > \frac{1}{200}||x_{[n]\setminus[k]}||_2\sqrt{k}/R_{t-1}\right] \le 200/R_t.$$

By a union bound over $t = 1, \ldots, T$ we have

$$\mathbf{Pr}_{\{\{H_{t,s}\}_{s\in[1:R_t]}\}_{t=1}^T}\left[||e_{S_t}^{tail,\mathcal{W}}(\{H_{t,s},\mathcal{A}_{t,s}\},x)||_1 \le \frac{1}{200}||x_{[n]\setminus[k]}||_2\sqrt{k}/R_{t-1} \text{ for all } t = 1,\ldots,T\right]$$
$$\ge 1 - \sum_{t=1}^{T} 200/R_t \ge 1 - \sum_{t=1}^{T} 200/(C_1 2^t) \ge 1 - O(1/C_1),$$

which gives the result as long as C_1 is larger than a constant, as required. Letting $\mathcal{E}_{small-noise}$ denote the intersection of the success events above completes the proof. □

E Semi-equispaced Fourier transform

One of the steps of our algorithm is to take the Fourier transform of our current estimate of x, so that it can be subtracted off in frequency domain and we can work with the residual. The *semi-equispaced FFT* provides an efficient method for doing this, and is based on the application of the standard inverse FFT to a filtered and downsampled signal. The following guarantee, which we rely on, was given in [IKP14, Sec. 12]:

Lemma E.1. [IKP14, Lemma 12.1, Cor. 12.2] **(a)** *Fix a power of two n and a constant $\delta > 0$. For every $x \in \mathbb{C}^n$ the procedure* SEMIEQUIFFT*(x, k, δ) returns a set of values $\{\widehat{y}_j\}_{|j|\le k/2}$ in time $O(\|x\|_0 \log(1/\delta) + k \log k)$, satisfying*

$$|\widehat{y}_j - \widehat{x}_j| \le \delta\|x\|_2$$

for every j, $|j| \le k/2$.

(b) *Given two additional parameters $\sigma, \Delta \in [n]$ with σ odd, it is possible to compute a set of values $\{\widehat{y}_j\}$ for all j equaling $\sigma j' + \Delta$ for some j' with $|j'| \le k/2$, with the same running time and approximation guarantee.*

Remark E.2. *We note that Corollary 12.2 is not stated in this form, but rather for the special case when the sparsity of the signal that we are working with is comparable with the length of the interval. The more general bounds stated above follow immediately from their proof.*

We will also use

Lemma E.3. [IKP14, Lemma 12.3] *Fix a power of two n and a constant $\delta > 0$. For every integer $k > 1$, every $S \subseteq [n]$, $|S| = k$, every $\widehat{x} \in \mathbb{C}^n$ such that* $\text{supp}(\widehat{x}) \subseteq [-k, k]$ *it is possible to compute a set of values $\{y_j\}_{j\in S}$ in time $O(k \log(n/\delta))$ satisfying*

$$|y_j - x_j| \le \delta\|x\|_2.$$

F Basic theorems

F.1 Basic identities involving the Fourier transform

Recall that we use the following normalization of the Fourier transform (as per (4)):

$$\hat{x}_f = \frac{1}{\sqrt{n}} \sum_{i\in[n]} \omega^{-if} x_i \text{ and } x_j = \frac{1}{\sqrt{n}} \sum_{f\in[n]} \omega^{jf} \hat{x}_f \tag{80}$$

We also use $\mathcal{F}$ and $\mathcal{F}^{-1}$ to denote the forward and inverse Fourier transforms respectively. Covolution is denoted by $(x * y)_i = \sum_{j\in[n]} x_{i-j} y_j$. With this normalization of the Fourier transform the convolution theorem takes form

$$\begin{aligned}(\mathcal{F}^{-1}(\widehat{x}\cdot\widehat{y}))_i &= \frac{1}{\sqrt{n}}\sum_{f\in[n]}\omega^{if}\widehat{x}_f\cdot\widehat{y}_f\\ &= \frac{1}{\sqrt{n}}\sum_{f\in[n]}\omega^{if}\frac{1}{n}\sum_{i',i''\in[n]}x_{i'}y_{i''}\omega^{-f(i'+i'')}\\ &= \frac{1}{\sqrt{n}}\sum_{i',i''\in[n]}x_{i'}y_{i''}\cdot\frac{1}{n}\sum_{f\in[n]}\omega^{f(i'+i''-i)}\\ &= \frac{1}{\sqrt{n}}\sum_{f'\in[n]}x_{f'}y_{i-i'}\end{aligned} \tag{81}$$

F.2 Proof of Claim A.2

We restate the claim here for convenience of the reader:
Claim A.2 *For every $C_1, C_2 > 0, \delta \in (0,1)$ there exists C_3 such that for every $C_4 \geq C_3$ one has $\frac{1}{C_4}2^{C_1 t}\cdot 2^{-C_2 2^{(1-\delta)t}+1} \leq 1$ all $t \geq 0$.*

Proof. One has $2^{C_1 t}\cdot 2^{-C_2 2^{(1-\delta)t+1}} = 2^{C_1 t - C_2 2^{(1-\delta)t}+1}$, so it suffices to note that since $\delta < 1$ by assumption of the claim,

$$\max_{t\geq 0}(2C_1 t - C_2^{(1-\delta)t})$$

is a constant(that may depend on C_1, C_2 and δ. Thus, the claim follows for sufficiently large C_3 (as a function of C_1, C_2, δ).

□

References

[AGS03] A. Akavia, S. Goldwasser, and S. Safra. Proving hard-core predicates using list decoding. *FOCS*, 44:146–159, 2003.

[Aka10] A. Akavia. Deterministic sparse Fourier approximation via fooling arithmetic progressions. *COLT*, pages 381–393, 2010.

[BCG$^+$12] P. Boufounos, V. Cevher, A. C. Gilbert, Y. Li, and M. J. Strauss. What's the frequency, Kenneth?: Sublinear Fourier sampling off the grid. *RANDOM/APPROX*, 2012.

[CCFC02] M. Charikar, K. Chen, and M. Farach-Colton. Finding frequent items in data streams. *ICALP*, 2002.

[Cip00] B. A. Cipra. The Best of the 20th Century: Editors Name Top 10 Algorithms. *SIAM News*, 33, 2000.

[CKPS16] Xue Chen, Daniel M. Kane, Eric Price, and Zhao Song. Fourier-sparse interpolation without a frequency gap. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 741–750, 2016.

[CKSZ17] Volkan Cevher, Michael Kapralov, Jonathan Scarlett, and Amir Zandieh. An adaptive sublinear-time block sparse Fourier transform, `https://arxiv.org/abs/1702.01286`. In *STOC*, 2017.

[CT06] E. Candes and T. Tao. Near optimal signal recovery from random projections: Universal encoding strategies. *IEEE Trans. on Info.Theory*, 2006.

[DIPW10] Khanh Do Ba, Piotr Indyk, Eric Price, and David P. Woodruff. Lower Bounds for Sparse Recovery. *SODA*, 2010.

[GGI$^+$02] A. Gilbert, S. Guha, P. Indyk, M. Muthukrishnan, and M. Strauss. Near-optimal sparse Fourier representations via sampling. *STOC*, 2002.

[GHI$^+$13] Badih Ghazi, Haitham Hassanieh, Piotr Indyk, Dina Katabi, Eric Price, and Lixin Shi. Sample-optimal average-case sparse Fourier Transform in two dimensions. In *51st Annual Allerton Conference on Communication, Control, and Computing, Allerton 2013, Allerton Park & Retreat Center, Monticello, IL, USA, October 2-4, 2013*, pages 1258–1265, 2013.

[GL89] O. Goldreich and L. Levin. A hard-core predicate for all one-way functions. *STOC*, pages 25–32, 1989.

[GLPS10] A. C. Gilbert, Y. Li, E. Porat, and M. J. Strauss. Approximate sparse recovery: optimizing time and measurements. In *STOC*, pages 475–484, 2010.

[GMS05] A. Gilbert, M. Muthukrishnan, and M. Strauss. Improved time bounds for near-optimal space Fourier representations. *SPIE Conference, Wavelets*, 2005.

[HAKI12] H. Hassanieh, F. Adib, D. Katabi, and P. Indyk. Faster gps via the sparse fourier transform. *MOBICOM*, 2012.

[HIKP12a] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Near-optimal algorithm for sparse Fourier transform. *STOC*, 2012.

[HIKP12b] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Simple and practical algorithm for sparse Fourier transform. *SODA*, 2012.

[HKPV13] Sabine Heider, Stefan Kunis, Daniel Potts, and Michael Veit. A sparse Prony FFT. *SAMPTA*, 2013.

[IK14] Piotr Indyk and Michael Kapralov. Sample-optimal Fourier sampling in any fixed dimension. *FOCS*, 2014.

[IKP14] Piotr Indyk, Michael Kapralov, and Eric Price. (Nearly) sample-optimal sparse Fourier Transform. *SODA*, 2014.

[Iwe10] M. A. Iwen. Combinatorial sublinear-time Fourier algorithms. *Foundations of Computational Mathematics*, 10:303–338, 2010.

[Kap16] Michael Kapralov. Sparse Fourier Transform in any constant dimension with nearly-optimal sample complexity in sublinear time (available as an arxiv report at `http://arxiv.org/abs/1604.00845`). *STOC*, 2016.

[KM91] E. Kushilevitz and Y. Mansour. Learning decision trees using the Fourier spectrum. *STOC*, 1991.

[LWC12] D. Lawlor, Y. Wang, and A. Christlieb. Adaptive sub-linear time fourier algorithms. *arXiv:1207.6368*, 2012.

[Man92] Y. Mansour. Randomized interpolation and approximation of sparse polynomials. *ICALP*, 1992.

[PR13] Sameer Pawar and Kannan Ramchandran. Computing a k-sparse n-length Discrete Fourier Transform using at most $4k$ samples and $O(k \log k)$ complexity. *ISIT*, 2013.

[Pri11] Eric Price. Efficient sketches for the set query problem. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 41–56, 2011.

[PS15] Eric Price and Zhao Song. A robust sparse Fourier Transform in the continuous setting. *FOCS*, 2015.

[RV08] M. Rudelson and R. Vershynin. On sparse reconstruction from Fourier and Gaussian measurements. *CPAM*, 61(8):1025–1171, 2008.