DIMACS Technical Report 2004-49
October 2004

Improved Time Bounds for Near-Optimal Sparse Fourier Representations

by

Anna C. Gilbert[1,2]
Dept. of Mathematics
University of Michigan
Ann Arbor, MI 48109.
annacg@umich.edu

S. Muthukrishnan[3,4]
Dept. of Computer Science
Rutgers University
New Brunswick, New Jersey 08903.
muthu@cs.rutgers.edu

Martin J. Strauss[5,6]
Depts. of Mathematics and EECS
University of Michigan
Ann Arbor, MI 48109.
martinjs@umich.edu

[1]Permanent Member
[2]Part of this work done while the first author was at AT&T Labs.
[3]Permanent Member
[4]Supported in part by NSF ITR 0220280.
[5]Permanent Member
[6]Part of this work done while the third author was at AT&T Labs.

ABSTRACT

We study the problem of finding a Fourier representation $\mathbf{R}$ of B terms for a given discrete signal $\mathbf{A}$ of length N. The Fast Fourier Transform (FFT) can find the optimal N-term representation in $O(N \log N)$ time, but our goal is to get *sublinear* algorithms when $B < N$ (typically, $B \ll N$).

Suppose $\|\mathbf{A}\|_2 \leq M \|\mathbf{A} - \mathbf{R}_{\mathrm{opt}}\|_2$, where $\mathbf{R}_{\mathrm{opt}}$ is the optimal output. The previously best known algorithms output $\mathbf{R}$ such that $\|\mathbf{A} - \mathbf{R}\|_2^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{R}_{\mathrm{opt}}\|_2^2$ with probability at least $1 - \delta$ in time[1] $\mathrm{poly}(B, \log(1/\delta), \log N, \log M, 1/\epsilon)$. Although this is sublinear in the input size, the dominating quantity is the polynomial factor in B which, for published algorithms, is greater than the bottleneck at B^2 that we identify below. Our experience with these algorithms shows that this is serious limitation in theory and in practice. Our algorithm beats this B^2 bottleneck.

Our main result is a significantly improved algorithm for this problem and the d-dimensional analog. Our algorithm outputs an $\mathbf{R}$ with the same approximation guarantees but it runs in time

$$B \cdot \mathrm{poly}(d, \log(1/\delta), \log N, \log M, 1/\epsilon).$$

We need two crucial ideas to achieve this bound: bulk sampling and estimation for multipoint polynomial evaluation using an unevenly-spaced Fourier tranform, and construction and use of arithmetic-progression independent random variables.

[1]The expression poly() denotes a polynomial function of the input.

1 Introduction

Fourier analysis consists of studying the spectra of functions in terms of their Fourier coefficients or frequencies in the functions and it has many applications from partial differential equations to discrete polynomial evaluation to list decoding to hard-core cryptographic predicates. In typical computational applications of Fourier analysis, we are interested only in a small number B of the coefficients. The large coefficients capture the major time-invariant wave-like features of the signal, while the smaller ones contribute little information about the signal. The largest few Fourier coefficients are useful in data compression, feature extraction, finding approximate periods and data mining. The problem of finding the B largest Fourier coefficients of a signal is a fundamental task in computational Fourier analysis. We address the problem of how to estimate these coefficients quickly and accurately.

Let us denote the optimal B-term Fourier representation of a signal $\mathbf{A}$ of length N by $\mathbf{R}_{\text{opt}}$ and assume that $\|\mathbf{A}\|_2 \leq M\|\mathbf{A} - \mathbf{R}_{\text{opt}}\| \geq 1/M$. Our main result in this paper is an algorithm that uses at most

$$B \cdot \text{poly}(\log(1/\delta), \log N, \log M, 1/\epsilon)$$

space and time and outputs a representation $\mathbf{R}$ such that $\|\mathbf{A} - \mathbf{R}\|_2^2 \leq (1+\epsilon)\|\mathbf{A} - \mathbf{R}_{\text{opt}}\|_2^2$, with probability at least $1 - \delta$. Since time B is needed to just output the coefficients, this is optimal in parameter B. We also extend our algorithms to the multidimensional case, paying a factor $d^{O(1)}$ for a d-dimensional problem. Previously known results [1, 6, 10] give similar bounds except for the dependence on B, which is linear in our algorithm, and at least quadratic in the other algorithms. This represents a much-needed, significant improvement.

There are three previously published papers on this problem; unfortunately, there are some missing links in citations. In the breakthrough paper [10], the author studies a variation of our problem and presents an algorithm to which one can immediately reduce our problem for the case N a power of 2 (but not for other N). In [6], the authors, unaware of [10], give an algorithm for any N, power-of-2 or not, with cost polynomial in $(B\log(N))$. Recently, in [1], the authors, independently from [6], present an algorithm with cost polynomial in $(B\log(N))$ for all values of N. The motivating applications in these three papers were quite different: [10] focused on learning, [6] focused on FFT approximation and [1] focused on list decoding for producing hard-core predicates in cryptography. Unfortunately, the dependence on B in all of these papers is quite high. In [10, 6], the cost is "polynomial in B"; a close look at the cost[2] reveals it to be at least $B^{\geq 2}$. In [1], the cost is at least $B^{\geq 5.5}$ (the heart of the procedure is in Section 7.2.4 where this complexity emerges). The polynomial in other factors has degree at most 2 and is reasonable.

Previous papers [10, 6, 1] focused on learning, complexity theory and sampling complexity respectively. In contrast, our focus is on practical applications of Fourier methods. There are a number of applications, e.g., spectral methods for differential equations [8], fast approximation correlation of signals [3], and deconvolving blurred signals [7], where the best B

[2] While the analyses of the algorithms in [10, 6] are not optimal in B, one could optimize their analyses and show that the algorithms do, in fact, take time $\Omega(B^2)$.

Fourier coefficients suffice and currently, the full DFT is used instead. However, the full DFT is a computational bottleneck in these applications. This motivated us to consider sampling algorithms for estimating the B best fourier terms more efficiently. We find in practice that if $B = 2$ and $N = 65,000$, a popular, optimized FFT package FFTW [5] and a naive $O(B^2)$ time implementation of our main result here perform about the same [13]; with larger N, sampling methods will be a win over FFTW. This shows that, even for quite reasonable values of N, one needs to consider sampling algorithms in addition to exact algorithms like FFTW. For larger values of B, the breakeven point for N where the naive implementation of our sampling algorithm performs about the same as FFTW will be larger; for example, for $B = 30$ and N around 4 million, FFTW, our quadratic-in-B algorithm, and our near-linear-in-B algorithm take about the same time. This shows that our near-linear-in-B algorithm surpasses previous algorithms (both FFTW and quadratic-in-B algorithms) even for moderate values of B and N. (We expect comparable performance for certain instances of the multidimensional algorithm. For example, we expect to handle the three-dimensional $127 \times 128 \times 131$ problem in roughly the same time as the one-dimensional problem of length $127 \cdot 128 \cdot 131 \approx 2$ million, and we expect that three-dimensional problems of this size and larger will become commonplace once the bottleneck of DFT is overcome.) For larger N *or* with an optimized implementation of our sampling algorithm *or* with growing dimension of the DFT calculation, we expect our methods to perform relatively better. We are hopeful that our algorithms will be a general replacement for the full FFT algorithm in many such applications.

Our algorithm, at a high level, proceeds in a greedy fashion. Given a signal $\mathbf{A}$, we set the representation $\mathbf{R} = 0$ and consider the residual $\mathbf{A} - \mathbf{R}$. We make progress on the residual by repeatedly

- IDENTIFYING a set of "significant" frequencies in the spectrum of $\mathbf{A} - \mathbf{R}$ and
- ESTIMATING the Fourier coefficients of these "significant" frequencies.

Once we have estimated the significant coefficients, we add their contribution to the representation $\mathbf{R}$, subtract their contribution from the signal, and iteratively analyze the residual signal $\mathbf{A} - \mathbf{R}$. This framework is similar to [10, 6, 1] although the specific steps differ substantively, and are the achievements of this paper.

From a technical view, there is a $\Omega(B^2)$ bottleneck in the overall framework. A B-term superposition $\mathbf{A}$ may have only N/B points in time with non-zero value. It follows that, if one samples randomly, one needs to sample approximately B positions and do work $\Omega(B)$ in order to learn anything. It is then critical that this $\Omega(B)$ work not be repeated to learn information about each of the $\Omega(B)$ frequencies in $\mathbf{A}$ or each $\Omega(B)$ frequencies in an intermediate representation $\mathbf{R}$. We break this $\Omega(B^2)$ bottleneck by showing:

- how to obtain $\Omega(B)$ samples from an intermediate representation (which may contain, say, $B/2$ frequencies) in $B\text{polylog}(N)$ work,
- how to identify all (at most $O(B)$) significant frequencies in an intermediate signal in total work $B\text{polylog}(N)$, and

- how to estimate up to $O(B)$ Fourier coefficients in $\mathbf{A}$ at once with work $B\text{polylog}(N)$.

Note that it takes work approximately B to obtain a *single* sample from a $(B/2)$-term intermediate representation, to identify a *single* significant frequency, or to estimate a *single* coefficient to sufficent accuracy. It follows that a straightforward algorithm for any of these steps would cost $\Omega(B^2)$.

In all three cases, we exploit the "bulk" computation of multiple values which is far more efficient than computing each value independently. To compute coefficients in bulk, we need special structure in the samples we make. Specifically, we make samples only on arithmetic progressions, so that the kth sample is $a + bk$, for random a and b. (We call these *ap-independent*.) Note that the points $(a + bk) \mod N$ are pairwise independent if N is prime, but not in the important case of N a power of 2, or other N. We achieve two broad goals for ap-independence. We show that ap-independent samples are sufficiently random that our algorithm is correct and that ap-independent samples are sufficiently structured that a set of about B ap-independent random variables can be reused about B times with total work about $B\text{polylog}(N) \ll B^2$.

The three tasks above are iterated as we find more and more frequencies and get better and better approximations to the coefficients for frequencies we have already found. Note that each iteration may find just a single significant frequency; a naive overall upper bound would then be B iterations of B^2 work, for a total cost of $O(B^3)$. Below, without substantially modifying the algorithm, we give a bound of approximately $\log(MN/\epsilon)$ iterations.

1.1 Linear versus Quadratic Algorithms

Certainly a near-linear algorithm is quantitatively better than a quadratic algorithm. In this section, we briefly argue that a near-linear-in-B algorithm is *structurally* better than a quadratic-in-B algorithm. We consider the many DFT applications in which time linear in N is needed, *e.g.,* for data acquisition. Then the $(N \log(N))$-time FFT algorithm becomes a bottleneck, at least in theory, and we want to consider an approximate algorithm that takes at most time N. Thus, for a near-linear-in-B algorithm, we can make B as large as $N/\text{polylog}(N)$; for a quadratic-in-B algorithm, we can only make B as large as $\sqrt{N}/\text{polylog}(N)$. We now consider the structural effects in three applications: convolutions, coding rate, and denoising.

To compute the convolution of two vectors x and y, we need to multiply their spectra. In the worst case, the non-zeros in $\widehat{x}$ correspond to (approximate) zeros in $\widehat{y}$ and vice versa, in which case our approximate algorithms give no useful results. If the spectra are random, however, then we might hope to get non-zeros in $\widehat{x}$ to correspond with non-zeros in $\widehat{y}$ with high probability. If $B = \sqrt{N}/\text{polylog}(N)$, then we are unlikely to find any collisions, and we get no information about the convolution. On the other hand, if $B = N/\text{polylog}(N)$, then we expect to get $N/\text{polylog}(N)$ collisions, which, depending on the context, might result in a useful approximation to the convolution.

Next, consider coding by choosing Fourier basis functions to have non-zero coefficients, where we require a decoding algorithm that runs in time linear in N. A near-linear-in-B

algorithm lets us code $\log\binom{N}{B} \approx N/\mathrm{polylog}(N)$ bits; a quadratic-in-B algorithm only lets us code $\log\binom{N}{B} \approx \sqrt{N}/\mathrm{polylog}(N)$ bits—quadratically worse, as expected. But, in coding applications, it is more natural to measure the *rate* of the code—the number of coded bits divided by the length of the codeword, N—and constant rate is often desired. Thus, by improving a quadratic-in-B algorithm to a linear one, the acheivable rate improves from around $1/\sqrt{N}$ to $1/\mathrm{polylog}(N)$, an exponential improvement.

Finally, consider the following *denoising* problem. There is a true signal ψ_ω. We observe the signal corrupted by additive Gaussian white noise with expected magnitude σ. What is the threshold for σ below which we can determine ω reliably? With high probability, The largest Fourier coefficient of the noise has square magnitude around $\log(N)\sigma^2$, so, even if we had unlimited time, we can recover ω iff $1 \geq \log(N)\sigma^2$, or $\sigma^2 \leq 1/\log(N)$. A dual formulation of our sampling algorithms can recover coefficients with approximately $1/B$ of the total energy (*i.e.*, square of L^2 norm), and the total energy is dominated by the noise energy, $\sigma^2 N$. Thus we need $1 \geq (1/B)\sigma^2 N$ in order to find a coefficient with energy 1 from a signal plus noise with energy $\sigma^2 N$; this means $\sigma^2 \leq B/N$. In a quadratic-in-B algorithm, $B \approx \sqrt{N}$, so we can only tolerate $\sigma^2 \approx 1/\sqrt{N}$. In a near-linear-in-$B$ algorithm, $B \approx N/\mathrm{polylog}(N)$, so we can tolerate $\sigma^2 \approx 1/\mathrm{polylog}(N)$, exponentially better, and much closer to the information-theoretic limit of $1/\log(N)$.

2 Preliminaries

Notation. Let $\mathbf{A} = (\mathbf{A}(0), \ldots, \mathbf{A}(N-1))$ be a signal indexed by t, regarded as an integer mod N. The complex conjugate of a number x is denoted by $\overline{x}$ and the dot product $\langle \mathbf{A}, \mathbf{B}\rangle$ of vectors $\mathbf{A}$ and $\mathbf{B}$ is $\sum_t \mathbf{A}(t)\overline{\mathbf{B}(t)}$. Recall that the functions $\frac{1}{\sqrt{N}} e^{2\pi i\omega t/N}$, $\omega \in \mathbb{Z}_N$, form an orthonormal basis for $\mathbb{Z}/N$. We can represent $\mathbf{A}$ as a linear combination of basis functions

$$\mathbf{A}(t) = \frac{1}{\sqrt{N}} \sum_{\omega=0}^{N-1} \widehat{\mathbf{A}}(\omega) e^{2\pi i\omega t/N}$$

where

$$\widehat{\mathbf{A}}(\omega) = \left\langle \mathbf{A}, \frac{1}{\sqrt{N}} e^{2\pi i\omega t/N} \right\rangle = \frac{1}{\sqrt{N}} \sum_t \mathbf{A}(t) e^{-2\pi i\omega t/N}$$

is the ωth Fourier coefficient of $\mathbf{A}$. The vector $\widehat{\mathbf{A}}$ is the *spectrum* of $\mathbf{A}$. The *energy* of $\mathbf{A}$ is $\|\mathbf{A}\|_2^2$, defined by $\|\mathbf{A}\|_2^2 = \sum_t |\mathbf{A}(t)|^2$. Parseval's equality says that $\sum_t |\mathbf{F}(t)|^2 = \sum_\omega |\widehat{\mathbf{F}}(\omega)|^2$. We refer to $|\widehat{\mathbf{A}}(\omega)|^2$ as the energy of the Fourier coefficient $\widehat{\mathbf{A}}(\omega)$ (or the energy of ω) and, similarly, the energy of a set of Fourier coefficients is the sum of the squares of their magnitudes. We write $\mathbf{F} \star \mathbf{G}$ to denote the convolution, $(\mathbf{F} \star \mathbf{G})(t) = \sum_s \mathbf{F}(s)\mathbf{G}(t-s)$. It follows that $\widehat{\mathbf{F} \star \mathbf{G}} = \sqrt{N}\widehat{\mathbf{F}}\widehat{\mathbf{G}}$. We denote by χ_S the signal that equals 1 on a set S and zero elsewhere. The index to χ_S may be time or frequency; this is made clear from context. The *support* $\mathrm{supp}(\mathbf{F})$ of a vector $\mathbf{F}$ is the set of t for which $\mathbf{F}(t) \neq 0$. For a sparse representation

$\mathbf{R}$, We will also write $\widehat{\mathrm{supp}}(\mathbf{R})$ to be the set of ω for which $\widehat{\mathbf{R}}(\omega) \neq 0$. For more background on Fourier analysis, see [12].

Define $\mathbf{H}_{K,N}$ by $\mathbf{H}_{K,N}(t) = \frac{\sqrt{N}}{K}\chi_{[0,K)}$. Then $\widehat{\mathbf{H}}_{K,N}(\omega) = \frac{\sin(K\pi\omega/N)}{K\sin(\pi\omega/N)}$, for $\omega \neq 0$ and $\widehat{\mathbf{H}}_{K,N}(0) = 1$. If $K \leq N$, this is called the "Dirichlet kernel" and the energy $\|\mathbf{H}_{K,N}\|^2$ is N/K. We will sometimes write $\mathbf{H}_K$ or $\mathbf{H}$ if N or both K and N are clear from context.

Permutation of Spectra. We define a transformation $\mathcal{P}_{\theta,\sigma}$ as follows. For a given signal $\mathbf{A}$, define $(\mathcal{P}_{\theta,\sigma}\mathbf{A})(t)$ by $(\mathcal{P}_{\theta,\sigma}\mathbf{A})(t) = e^{-2\pi i\theta t/(\sigma N)}\mathbf{A}(t/\sigma)$. First note that one can sample from $\mathcal{P}_{\theta,\sigma}\mathbf{A}$ with approximately the same cost as sampling from $\mathbf{A}$. Next, by elementary facts from Fourier analysis, we have $\widehat{\mathcal{P}_{\theta,\sigma}\mathbf{A}}(\omega) = \widehat{\mathbf{A}}(\theta + \sigma t)$. Since $t \mapsto \theta + \sigma t$ is invertible mod N if σ is, $\mathcal{P}_{\theta,\sigma}$ is a spectral permutation. (A number is invertible mod N if and only if it is relatively prime to N.)

Precision. We assume that all signal entries are bounded by some number, M. Similarly, our output will be accurate only to $\pm 1/M$, additively. Thus we need $2\log(M)$ bits to process inputs and outputs, and our algorithm will be allowed time polylogarithmic in M in the bit model.

Randomization. Our algorithms are randomized. That is, for all inputs $\mathbf{A}$ and 3/4 of the random choices of our algorithm, the algorithm succeeds. The success probability 1/4 ("significant") or 3/4 ("high") can be boosted to as close to 1 as desired ("overwhelming") using standard inexpensive techniques. We sometimes omit details.

Non-adaptivity. We have the following non-adaptive access to $\mathbf{A}$. We toss coins, then, based on the coins, compute a sequence $T \subseteq [0, N)$ of indices, learn $\mathbf{A}(T) = \{(t, \mathbf{A}(t)) : t \in T\}$ and run a deterministic algorithm on T, $\mathbf{A}(T)$, and the coin flips without further access to $\mathbf{A}$. Thus we say that the sampling is *non-adaptive*. However, for convenience, we will present the algorithm in an adaptive way. More specifically, we will present an algorithm that flips coins as it needs them, and such that the algorithm's actions (but not sample locations) depends on previously-computed values, including coin flips and sample values. Our goal is to bound the time used by the algorithm, which implies a bound on the number of samples made. In practice, one can save factors of $\log(N)$ in time by sampling adaptively or by adaptively deciding to make fewer samples, but this is not the focus of this paper.

Algorithm parameters. To aid the reader in parsing the statements of the lemmas, we establish some parameter setting constraints at the outset. The signal length, N, may be an arbitrary integer, but some aspects of our algorithm are more efficient or elegant if N is a power of 2, so we also include techniques for N a power of 2 as a special case. During greedy pursuit, we will attempt to find up to K frequencies at once; we assume K is a power of 2 equal to $B/\mathrm{poly}(\epsilon)$. We will build up an intermediate representation of L terms, where L is $B\log(M)/\mathrm{poly}(\epsilon)$ and can be assumed to be a power of 2 by taking additional dummy frequencies with zero coefficients.

A frequency ω is η-significant in $\mathbf{A}$, for $\eta > 0$, if $|\widehat{\mathbf{A}}(\omega)|^2 \geq \eta\|\mathbf{A}\|^2$. An L-term representation $\mathbf{R}$ is a linear combination of L frequencies,

$$\mathbf{R}(t) = \frac{1}{\sqrt{N}}\sum_{l=0}^{L-1} c_l e^{-2\pi i\omega_l t/N}.$$

We store $\mathbf{R}$ as a set of pairs $\{(c_l, \omega_l) | l = 0, \ldots, L-1\}$.

2.1 More on the Dirichlet Kernel

In this section we give a useful technical property of the Dirichlet kernel. It involves the Euler totient function, $\phi(N)$, which gives the number of positive integers less than N and relatively prime to N. We include a brief sketch of the following lemma for completeness.

Lemma 1 *Let $\phi(N)$ be the number of positive integers less than N that are relatively prime to N. Then $N/\phi(N) \leq O(\log(N))$.*

Proof: Consider the inverse ratio

$$\phi(N)/N = \prod_{\substack{p \text{ prime} \\ p | N}} (1 - 1/p).$$

There are at most $\log(N)$ primes dividing N, and each factor $(1-1/p)$ gets smaller as p gets smaller. Thus, if p_n is the n'th prime dividing N, if we replace p_n above with $n \leq p_n$, we have

$$\frac{\phi(N)}{N} \geq \prod_{n \leq \log(N)} (1 - 1/n) \approx e^{-\sum_{n\leq \log(N)} \frac{1}{n}} \approx e^{-\log\log(N)} = \frac{1}{\log(N)}.$$

∎

Although the bound of Lemma 1 is acceptable for our purposes, we note that our algorithm will pay a cost factor of $N/\phi(N)$—and, more importantly, some versions in d dimensions will pay $(N/\phi(N))^d$—so we will want to bound $N/\phi(N)$ tightly as possible. Note that, if N is a power of 2 then $\phi(N) = N/2$ and if N is prime then $\phi(N) = N-1$. In general, we can bound $N/\phi(N)$ by $\log^{O(1)} \log(N)$, since the prime number theorem says that the n'th prime is $\Theta(n \log(n))$. It follows that

$$\begin{aligned}
\prod_{\substack{p \text{ prime} \\ p | N}} (1 - 1/p) &\geq \prod_{n \leq \log(N)} (1 - \Theta(1)/(n \log(n))) \\
&\geq e^{-\sum_{n \leq \log(N)} \frac{\Theta(1)}{n \log(n)}} \\
&\geq e^{-\Theta(\log\log\log(N))} \\
&= \frac{1}{\log^{\Theta(1)} \log(N)},
\end{aligned}$$

so that $N/\phi(N) \leq \log^{O(1)} \log(N)$.

It follows that at least the fraction $\phi(N)/N \geq 1/\log(N)$ of the integers mod N are invertible. Thus we can efficiently pick a random invertible element mod N by picking a random element and rejecting it if it is not invertible, as determined efficiently by Euclid's algorithm.

Next we give a technical lemma about the Dirichlet kernel. The lemma says that if we sample $|\widehat{\mathbf{H}}_k(\omega)|^2$ from ω uniformly picked from a certain type of easily-constructible set we get not much more than the value, $1/K$, that we would get if we sampled ω uniformly from all $\mathbb{Z}_N$. This will be motivated below, where it is used.

Lemma 2 *Let N be any number and fix $K \le N$. For some constant c specified below, let p_0 be any number with $|p_0| \le cN/K$ and fix $p \ne p_0$ modulo N. Let b be a random invertible number mod N. Then*

$$E_b\left[|\widehat{\mathbf{H}}_K(b(p-p_0)+p_0)|^2 \middle| \gcd(b,N)=1\right] \le O\left(\frac{N}{\phi(N)K}\right).$$

Proof: By symmetry of $\widehat{\mathbf{H}}_K$, we may assume $p_0 \ge 0$. Let $g = \gcd(p-p_0, N)$ and let x be p_0 mod g; it follows that the distribution on $b(p-p_0)+p_0$ is the same as the distribution on $bg+x$, where b is a random invertible element. It is easy to see that $|\widehat{\mathbf{H}}_K(\omega)| \le h(\omega)$, where the envelope $h : \mathbb{Z}_N \to \mathbb{R}$ is

$$h(\omega) = \begin{cases} 1, & |\omega| < \frac{N}{2K} \\ \frac{4N}{K|\omega|}, & \frac{N}{2K} \le |\omega| \le N/2. \end{cases}$$

Thus it suffices to show that

$$E_b\left[|h(b(p-p_0)+p_0)|^2 \middle| \gcd(b,N)=1\right] \le O\left(\frac{N}{\phi(N)K}\right).$$

Note that $E_b[|h(b)|^2] \le O(1/K)$.

We now show that $h(bg)^2 \le O(h(bg+x)^2)$ for invertible b, so that we may assume $x=0$. We may assume that $0 \le bg \le bg+x < N/2+N/K$, since, otherwise, by the unimodularity of h, it is easy to see that $h(bg) \le h(bg+x)$. We consider two cases. First suppose $bg < N/K$. Then $bg+x \le O(N/K)$, so, from the definition of $h()$, $h(bg+x)^2 \ge \Omega(1)$. Since $h()$ is always at most 1, it follows that $h(bg) \le 1 \le O(h(bg+x))$. Now suppose $bg \ge N/K$. Then $bg+x \le O(bg)$. Again, it follows from the definition of $h()$ that $h(bg) \le O(h(bg+x))$.

To bound $E\left[h(bg)^2 \middle| \gcd(b,N)=1\right]$, proceed as follows. We have

$$\begin{aligned} E\left[h(bg)^2 \middle| \gcd(b,N)=1\right] &= \frac{1}{\phi(N)} \sum_{b \in \mathbb{Z}_N^*} h(bg)^2 \\ &\le \frac{1}{\phi(N)} \sum_{b \in \mathbb{Z}_N \setminus (N/g)\mathbb{Z}_N} h(bg)^2 \\ &= \frac{g}{\phi(N)} \sum_{0<b<N/g} h(bg)^2 \\ &= \frac{g}{\phi(N)} \sum_{-N/(2g) \le b < N/(2g), b \ne 0} h(bg)^2, \end{aligned}$$

since, by periodicity, $h((b+jN/g)g) = h(bg)$, for integers j. Since $h(x)^2$ decreases as $|x|$ increases, we have $h(bg)^2 \leq \frac{1}{g}\sum_{0\leq x<g} h(bg-x)^2$, if $0 < bg \leq N/2$. Thus we have, using symmetry of h and double counting $h(N/2)$ when 2 divides N/g,

$$\begin{aligned}
E\left[h(bg)^2\middle|\ \gcd(b,N)=1\right] &\leq \frac{g}{\phi(N)} \sum_{-N/(2g)\leq b<N/(2g),b\neq 0} h(bg)^2 \\
&\leq \frac{2g}{\phi(N)}\left(\sum_{0<b\leq N/(2g)} h(bg)^2\right) \\
&\leq \frac{2g}{\phi(N)}\left(\sum_{0<b\leq N/(2g)} \frac{1}{g}\sum_{0\leq x<g} h(bg-x)^2\right) \\
&\leq \frac{1}{\phi(N)}\left(\left(\sum_{0\leq b<N} h(b)^2\right) - h(0)^2 + h(\lfloor N/2\rfloor)^2\right),
\end{aligned}$$

where the last term, $h\left(\lfloor N/2\rfloor\right)^2$, is only needed if 2 divides N/g. By definition of h, we have $h(0)^2 \geq h\left(\lfloor N/2\rfloor\right)^2$, so that

$$\begin{aligned}
E\left[h(bg)^2\middle|\ \gcd(b,N)=1\right] &\leq \frac{1}{\phi(N)}\left(\left(\sum_{0\leq b<N} h(b)^2\right) - h(0)^2 + h(\lfloor N/2\rfloor)^2\right) \\
&\leq \frac{1}{\phi(N)}\sum_{0\leq b<N} h(b)^2 \\
&\leq O\left(\frac{N}{\phi(N)}\cdot\frac{1}{K}\right).
\end{aligned}$$

∎

We note without proof that, if $p_0 = 0$ and N and K take special values, then we get the tighter conclusion

$$E_b\left[|\widehat{\mathbf{H}}_K(b(p-p_0)+p_0)|^2\middle|\ \gcd(b,N)=1\right] \leq \frac{1}{K}.$$

This holds, for example, if N is a prime or if N and K are powers of 2. The proof follows from the inclusion/exclusion principle and the observation that $\mathbf{H}_{K,N}(bg) = \mathbf{H}_{K,N/g}(b)$ provided $K \leq N/g$. This gives a tighter result in Lemma 4.

3 AP-Independence

To estimate the expectation $E[Y]$ of a random variable Y, we can form a collection $\{X_k\}$ of independent and identically-distributed copies of Y. We can then define $X = \frac{1}{K}\sum_{k=1}^{K} X_k$.

Then $E[X] = E[Y]$ and $\text{var}(X) = \frac{1}{K}\text{var}(Y)$; the reduced variance will allow us to estimate $E[X]$ from a sample of X. It is known that the X_k's need only be pairwise independent for this variance argument to work. We will now claim that the argument holds for a particular construction of X_k's with somewhat less than pairwise independence.

Definition 3 Fix N and let ϕ be a complex-valued function on $\mathbb{Z}_N$. Fix $K \leq N$. Then a sequence $(X_k : 0 \leq k < K)$ of random variables is called *N-arithmetic-progression-independent* (briefly, *ap-independent*) ϕ^{-1}-distributed, if the joint distribution on the sequence can be obtained by writing $X_k = \phi(t_k)$, where a is a random integer mod N, b is a random invertible integer mod N, and $t_k = a + bk$. ∎

Note that, if N is a prime, then the family $\{a+bk : 0 \leq k < K\}$ for random a and random b, is pairwise independent, so the family $X_k = \phi(a + bk)$ of random variables is pairwise independent, as usual. Our construction differs in some important respects. For example, b is restricted to be invertible mod N, so the values $a + bk$ and $a + bk'$ are guaranteed to be different for $k \neq k'$. It follows that $a + bk$ and $a + bk'$ are guaranteed to be *dependent*, and so X_k and $X_{k'}$ are dependent, except for trivial ϕ. Nevertheless, we show that ap-independent random variables are sufficiently independent for the desired variance reduction.

Lemma 4 *Fix K and N with $K \leq N$. Let ϕ be a complex-valued function of $\mathbb{Z}_N$ and let (X_k) be a sequence of K ap-independent ϕ^{-1}-distributed random variables. Let $X = \frac{1}{K}\sum_k X_k$ and let Y be distributed independently and identical to the X_k's. Then $E[X] = E[Y]$, and* $\text{var}(X) \leq O\left(\frac{\log(N)}{K}\right)\text{var}(Y)$. *If N and K are powers of 2, then* $\text{var}(X) \leq \frac{1}{K}\text{var}(Y)$.

Proof: Note that each t_k is uniformly distributed, so the statement about expectation follows from the linearity of expectation. Note that $E[Y] = \frac{1}{\sqrt{N}}\widehat{\phi}(0)$. From the definition of variance and Parseval's equality,

$$\text{var}(Y) = \frac{1}{N}\left(\sum_t |\phi(t)|^2 - |\widehat{\phi}(0)|^2\right) = \frac{1}{N}\sum_{\omega \neq 0} |\widehat{\phi}(\omega)|^2.$$

Suppose (X_k) is defined on the arithmetic progression $a + bk$, so that $X_k = \phi(a + bk)$. Then $\text{supp}(\mathbf{H}_K) = \{0, 1, 2 \ldots, K - 1\}$, so $\text{supp}(\mathcal{P}_{0,-1/b}\mathbf{H}_K) = \{0, -b, -2b, \ldots, -(K - 1)b\}$, and so

$$\begin{aligned}
(\phi \star \mathcal{P}_{0,-1/b}\mathbf{H}_K)(a) &= \sum_{r+s=a} (\phi(r)\mathcal{P}_{0,-1/b}\mathbf{H}_K(s) \\
&= \phi(a)\mathbf{H}_K(0) + \phi(a + b)\mathbf{H}_K(-b) + \phi(a + 2b)\mathbf{H}_K(-2b) + \cdots \\
&\qquad +\phi(a + (K - 1)b)\mathbf{H}_K(-(K - 1)b) \\
&= \frac{\sqrt{N}}{K} \sum_{0 \leq k < K} X_k,
\end{aligned}$$

since $\mathbf{H}_K$ is equal to $\frac{\sqrt{N}}{K}$ on its support. Thus $X = (\phi \star \mathcal{P}_{0,-1/b}\mathbf{H}_K)(a)/\sqrt{N}$. So, conditioned on b and taking variance in a,

$$\mathrm{var}_a(X|b) = \frac{1}{N^2}\sum_{\omega\neq 0}\left|(\phi \star \mathcal{P}_{0,-1/b}\mathbf{H}_K)\widehat{\ }(\omega)\right|^2 = \frac{1}{N}\sum_{\omega\neq 0}\left|\widehat{\phi}(\omega)\widehat{\mathbf{H}}_K(b\omega)\right|^2.$$

We now take expectation with respect to b. Using Lemma 2 with $p=\omega$ and $p_0=0$, we have as desired:

$$\mathrm{var}(X) \le \frac{1}{N}\sum_{\omega\neq 0}\left|\widehat{\phi}(\omega)\right|^2 \max_{\omega\neq 0} E_b\left[\left|\widehat{\mathbf{H}}_K(b\omega)\right|^2\right] \le O\left(\frac{\log(N)}{KN}\right)\sum_{\omega\neq 0}\left|\widehat{\phi}(\omega)\right|^2.$$

∎

In the sequel, we will substitute AP-independent random variables for pairwise independent random variables. Lemma 2, even for the special case of $p_0 = 0$, insures that the variance reduction succeeds. The more general Lemma 2 for $p_0 \neq 0$ will be used in a different context.

4 Identification of significant frequencies

The first step in our greedy algorithm is the Identification step. At some intermediate stage of our calculation on original signal $\mathbf{A}$, we have a representation in progress, denoted $\mathbf{R}$, and we want to analyze the residual $\mathbf{A}-\mathbf{R}$. We show how to identify a set Λ of frequencies ω that are η-significant in $\mathbf{A}-\mathbf{R}$, where η is $\mathrm{poly}(\epsilon)/B$.

Lemma 5 (Identification) *There's a sampling algorithm that takes input* $(\beta, \delta, \eta, L, N, \mathbf{R})$ *and oracle* $\mathbf{A}$ *and runs in time* $(L + 1/\eta)\log(1/\beta)\log(1/\delta)(\log(N)+\log(1/\eta))^{O(1)}$*, and, with probability at least* $1-\delta$*, produces a set, of size at most* $O((1/\eta)\log(1/\eta))$*, containing all frequencies* ω *which are* η*-significant in the residual and whose energy is not too small,* $|\widehat{\mathbf{A}-\mathbf{R}}(\omega)|^2 \ge \beta(\|\mathbf{A}\|^2 + \|\mathbf{R}\|^2)$.

We prove this lemma in steps. First we define mathematically a randomized set of frequencies to output and prove that it is correct (Section 4.1). Then we will argue, by proving an additional lemma (Section 4.2), that the randomized set can be computed efficiently by an algorithm whose inputs are correct (with probability 1) and whose inputs and intermediate values are infinite-precision complex numbers. Finally, we will argue that our algorithm works on high-probability inputs and with limited precision. (Section 4.3).

4.1 Mathematical description of Identification algorithm

The Identification step consists of

- Isolation. Generate a set $\{\mathbf{F}_k : k < K\}$ of K new signals from $\mathbf{A} - \mathbf{R}$ where $K \leq O(1/\eta)$ is sufficiently large, so that each ω that is η-significant in $\mathbf{A} - \mathbf{R}$ is likely to be $(1-\gamma)$-significant in some $\mathbf{F}_k$ for some small constant γ.
- Group Testing. Assuming N is a power of 2 (the general case is similar but requires an additional trick):
  - Projection. Project each $\mathbf{F}_k$ onto the space of even frequencies and the space of odd frequencies.
  - Norm Estimation Estimate the norm of each projection, thereby learning the least significant bit of ω.

  Repeat for other bits of ω, from least to most significant.

We use the Dirichlet kernel $\mathbf{H}_K$, for appropriate K, to isolate frequencies. We will first permute the spectrum of a given signal, then filter with $\mathbf{H}_K$, and then undo the permutation.

Definition 6 Given signal $\mathbf{A}$ of length N, and given parameter K a power of 2, a K-shattering of $\mathbf{A}$ is a collection $\left\{\frac{1}{\sqrt{N}}(\mathcal{P}_{kN/K,\sigma}\mathbf{H}_K) \star \mathbf{A}\right\}_k$, where σ is a random number invertible mod N. ∎

The next lemma guarantees that a K-shattering isolates significant frequencies. That is, each desired frequency is isolated in some element of the shattering with probability $\Omega(1)$.

Lemma 7 *Let γ be any positive constant. Let ω_0 be an η-significant frequency in $\mathbf{A}$, so that $|\widehat{\mathbf{A}}(\omega_0)|^2 \geq \eta \left\|\mathbf{A}\right\|^2$. Then, for some $K \leq O(\log(N)/(\gamma\eta))$, in a K-shattering $\{\mathbf{F}_k\}_k$ of $\mathbf{A}$, with probability $\Omega(1)$, there exists $k \leq K$ such that $|\widehat{\mathbf{F}}_k(\omega_0)|^2 \geq (1-\gamma)\left\|\mathbf{F}_k\right\|^2$.*

Proof: Note that there is some k such that

$$|\sigma\omega_0 + kN/K| \leq N/(2K). \tag{1}$$

Then, by the properties of the Dirichlet kernel, $|(\widehat{\mathcal{P}_{kN/K,\sigma}\mathbf{H}_K})(\omega_0)| = |\widehat{\mathbf{H}}_K(\sigma\omega_0 + kN/K)| \geq 2/\pi$, whence, for some constant c, $|\widehat{\mathbf{F}}_k(\omega_0)|^2 = |\widehat{\mathbf{H}}_K(\sigma\omega_0 + kN/K)|^2|\widehat{\mathbf{A}}(\omega_0)|^2 \geq c\eta\left\|\mathbf{A}\right\|^2$. Next, for each k,

$$\left\|\mathbf{F}_k\right\|^2 - |\widehat{\mathbf{F}}_k(\omega_0)|^2 = \sum_{\omega\neq\omega_0}|\widehat{\mathbf{F}}_k(\omega)|^2 = \sum_{\omega\neq\omega_0}|\widehat{\mathbf{A}}(\omega)|^2|\widehat{\mathbf{H}}_K(\sigma\omega + kN/K)|^2.$$

Lemma 2 implies that $E_\sigma\left[|\widehat{\mathbf{H}}_K(\sigma\omega + kN/K)|^2\middle|\,(1)\text{ holds}\right] \leq \frac{1}{4}c\gamma\eta$, so that

$$E\left[\left\|\mathbf{F}_k\right\|^2 - |\widehat{\mathbf{F}}_k(\omega_0)|^2\middle|\,(1)\text{ holds}\right] \leq \frac{1}{4}c\gamma\eta\left\|\mathbf{A}\right\|^2.$$

By Markov, it follows that, for each k, with probability at least 3/4, if (1) holds, then $\left\|\mathbf{F}_k\right\|^2 - |\widehat{\mathbf{F}}_k(\omega_0)|^2 \leq c\gamma\eta\left\|\mathbf{A}\right\|^2$. In that case, $|\widehat{\mathbf{F}}_k(\omega_0)|^2 \geq (1-\gamma)\left\|\mathbf{F}_k\right\|^2$, as desired. ∎

Now, we show how to project the signal approximately onto the space of even and odd frequencies. We also generalize this to other pairs of subspaces that we will need.

Definition 8 Define the filter pair $\mathbf{G}_n^{\pm}$ by $\mathbf{G}_n^{\pm} = \left(\frac{\sqrt{N}}{2}\right)\left(\delta_0 \pm \delta_{\lfloor N/2^{n+1}\rfloor}\right)$. ∎

The next Lemma shows how these filters can be used to find the $(n+1)$'s least significant bit of ω, provided we know the n least significant bits (by induction) and a few of the most significant bits, which we can assume by trying all possibilities exhaustively. If we know the values of any bit positions in an otherwise unknown ω_0, we can modulate the relevant signal (*i.e.*, translate the spectrum), so that all of the known bit positions become zeros, which we assume below. This does not change any of the unknown bit positions.

Lemma 9 (Projection) *For all $\gamma > 0$ and all n, if $0 \le \omega_0 < N\gamma/\pi$ and $\omega_0 \bmod 2^n = 0$, then*

$$\begin{cases} 1-\gamma \le |\widehat{\mathbf{G}_n^{+}}(\omega_0)|^2 \le 1 \quad \text{and} \quad |\widehat{\mathbf{G}_n^{-}}(\omega_0)| \le \gamma & \omega_0 = 0 \bmod 2^{n+1} \\ 1-\gamma \le |\widehat{\mathbf{G}_n^{-}}(\omega_0)|^2 \le 1 \quad \text{and} \quad |\widehat{\mathbf{G}_n^{+}}(\omega_0)| \le \gamma & \omega_0 = 2^n \bmod 2^{n+1} \end{cases}$$

Proof: For N a power of 2, the result is known from [10]. We sketch that algorithm as an ideal case. We then generalize the result to other N.

First assume N is a power of 2. Consider the filters $\mathbf{F}_0^{\pm} = (\sqrt{N}/2)(\delta_0 \pm \delta_{N/2})$. It's easy to see that

$$\widehat{\mathbf{F}_0^{+}}(\omega) = \begin{cases} 1, & \omega \text{ even} \\ 0, & \omega \text{ odd}, \end{cases}$$

and, similarly, $\widehat{\mathbf{F}_0^{-}}(\omega)$ is 0 or 1, respectively, depending on whether ω is even or odd. More generally, define $\mathbf{F}_n^{\pm} = (\sqrt{N}/2)(\delta_0 \pm \delta_{N/2^{n+1}})$. It's easy to see that $\widehat{\mathbf{F}_n^{+}}(\omega) = 1$ if $\omega = 0 \bmod 2^{n+1}$ and $\widehat{\mathbf{F}_n^{+}}(\omega) = 0$ if $\omega = 2^n \bmod 2^{n+1}$; $\widehat{\mathbf{F}_n^{-}}(\omega) = 1 - \widehat{\mathbf{F}_n^{+}}(\omega)$ if 2^n divides ω.

Now consider general N. We cannot use the ideal filters $(\sqrt{N}/2)(\delta_0 \pm \delta_{N/2})$, because $N/2$ is not necessarily an integer. Instead, we will define $\mathbf{G}_0^{\pm} = (\sqrt{N}/2)(\delta_0 \pm \delta_{\lfloor N/2\rfloor})$. Then

$$\widehat{\mathbf{G}_0^{+}}(\omega) = \frac{1 + e^{2\pi i\omega\lfloor N/2\rfloor/N}}{2}.$$

Put $\Delta = N/2 - \lfloor N/2\rfloor$, so $\Delta = 0$ or $\Delta = 1/2$. Thus

$$\widehat{\mathbf{G}_0^{+}}(\omega) = \frac{1 + e^{-2\pi i\omega\Delta/N}(-1)^{\omega}}{2}.$$

Now, suppose $0 \le \omega_0 < N\gamma/\pi$. Then $\omega_0\Delta/N \le \gamma/(2\pi)$. Thus, if ω_0 is even, $\widehat{\mathbf{G}_0^{+}}(\omega_0) = (1 + e^{-2\pi i\omega_0\Delta/N})/2 \approx 1 + \pi i\omega_0\Delta/N$ so that $|\widehat{\mathbf{G}_0^{+}} - 1| \le \gamma$ and, similarly, $|\widehat{\mathbf{G}_0^{-}}| \le \gamma$, provided γ is small enough to force $\pi\omega/N \le \gamma$ to be small enough that $|1 - e^{-2\pi i\omega_0\Delta/N}| \le 4\pi\omega_0\Delta/N$. Similarly, if ω_0 is odd, then $|\widehat{\mathbf{G}_0^{+}}| \le \gamma$ and $|\widehat{\mathbf{G}_0^{-}} - 1| \le \gamma$. It follows that, for sufficiently small $\gamma \geq \Omega(1)$, the filters $\mathbf{G}_0^{\pm}$ behave similarly to $\mathbf{F}_0^{\pm}$, provided $0 \leq \omega_0 < N\gamma/\pi$; *i.e.*, provided the most significant $\log(\pi/\gamma)$ bits of ω_0 are zero. Similar considerations hold for $\mathbf{G}_k^{\pm}$, $k > 0$. In general, we will perturb $t \approx N/2^n$ by $\Delta \leq 1/2$ to get an integer $\lfloor N/2^n \rfloor$ or $\lceil N/2^n \rceil$; we then need $|\omega| \leq N\gamma/\pi$ so that the perturbation $2\pi i\omega\Delta/N$ is at most γ, so that $(e^{2\pi i\Delta t/N} - 1)/2 \approx \gamma/2 \leq \gamma$. Finally, $|\widehat{\mathbf{G}_n^{\pm}}(\omega)|^2 \leq 1$ for all ω by the triangle inequality. ∎

Finally, we show how to estimate norms (equivalently, energies), by sampling. We cannot estimate norms reliably (for example, if the signal consists of a single spike, our sampling algorithm cannot find the spike to learn its height). But we can get a certain one-sided estimate that suffices for us. A similar lemma appeared in [10].

Definition 10 For integers J and K, and signal $\mathbf{A}$, define the estimator $\|\mathbf{A}\|^2_{\sim J,K}$ (or $\|\mathbf{A}\|^2_{\sim}$, if J and K are understood) as follows. Choose JK points t at random, regarded as J groups of K points. Take the mean over each group of $N|\mathbf{A}(t)|^2$. Take the median of the means. Finally, divide by 4. ∎

Lemma 11 (Norm Estimation) *For all constants $\alpha > 0$ and all $\delta > 0$ there's a constant $\beta > 0$, and integer $J \leq (\log(1/\delta))$ and $K \leq O(1/\alpha^2)$ such that the following hold probability at least $1 - \delta$:*

- $\|\mathbf{A}\|^2_{\sim J,K} \leq \|\mathbf{A}\|^2$ *(for any $\mathbf{A}$);*
- *if $|\widehat{\mathbf{A}}(\omega)|^2 \geq \alpha \|\mathbf{A}\|^2$ for some ω, then $\|\mathbf{A}\|^2_{\sim J,K} \geq \beta \|\mathbf{A}\|^2$.*

Proof: Let $X = N|\mathbf{A}(t)|^2$ for random t and let Y be the average of K independent copies of X. Then $E[Y] = E[X] = \|\mathbf{A}\|^2$. First consider the upper bound. Since $X \geq 0$, by the Markov inequality, it follows that $\Pr(Y > 4E[Y]) \leq 1/4$.

Now consider the lower bound. Write $\psi = \widehat{\mathbf{A}}(\omega)\psi_\omega$ and $\nu = \mathbf{A} - \psi$, where $\langle \psi, \nu \rangle = 0$. Assuming $\alpha \leq 1$, define ν' by

$$\nu'(t) = \begin{cases} \nu(t), & |\nu(t)|^2 \leq \frac{2}{\alpha}|\psi(t)|^2; \\ 0, & \text{otherwise.} \end{cases}$$

Let $X'(t) = N|\psi(t) + \nu'(t)|^2$ and let Y' be the average of K independent copies of X'. Then, for all t, we have $X'(t) \leq X(t)$; similarly, $Y' \leq Y$ identically. Thus it suffices to show that $\Pr(Y' \leq \beta \|\psi\|^2)$ is small. To do this, it suffices to show that $E[X'] \geq \Omega(\|\psi\|^2)$ and $E[|X'|^2] \leq O(\|\psi\|^2)$. Then, by taking K large enough, it follows, say, that $E[|Y'|^2] \leq \frac{1}{4}\|\psi\|^2$. We can then choose β appropriately to get our result.

First, consider $E[X']$. Let $T = \{t : |\nu(t)|^2 > \frac{2}{\alpha}|\psi(t)|^2\}$. Suppose that $|T| \leq \alpha N/5$. Then

$$\begin{aligned} E[X'] &= \|\psi + \nu - \mathbf{A}\chi_T\|^2 \\ &\geq \|\psi\|^2 - 2|\langle \psi, \nu - \mathbf{A}\chi_T \rangle| \\ &\geq \|\psi\|^2 - 2|\langle \psi, \mathbf{A}\chi_T \rangle|, \end{aligned}$$

since $\langle \psi, \nu \rangle = 0$. Thus

$$\begin{aligned} E[X'] &\geq \|\psi\|^2 - 2|\langle \psi, \mathbf{A}\chi_T \rangle| \\ &\geq \|\psi\|^2 - 2|\langle \psi\chi_T, \mathbf{A} \rangle| \\ &\geq \|\psi\|^2 - 2\,\|\psi\chi_T\|\,\|\mathbf{A}\| \\ &\geq \|\psi\|^2 - 2\sqrt{\frac{\alpha}{5}}\,\|\psi\|\,\|\mathbf{A}\|\,, \end{aligned}$$

because $|\psi(t)|$ is constant and only $\alpha N/5$ of the N positions contribute to $\|\psi\|^2$. Thus

$$\begin{aligned} E[X'] &\geq \|\psi\|^2 - 2\sqrt{\frac{\alpha}{5}}\,\|\psi\|\,\|\mathbf{A}\| \\ &\geq \left(1 - \frac{2}{\sqrt{5}}\right)\|\psi\|^2\,, \end{aligned}$$

since $\|\mathbf{A}\| \geq \frac{1}{\sqrt{\alpha}}\|\psi\|$.

On the other hand, if $|T| > \alpha N/5$, then

$$\begin{aligned} E[X'] &\geq \sum_{t\in T} |\mathbf{A}(t)|^2 \\ &\geq \sum_{t\in T} \left(\sqrt{\frac{2}{\alpha}} - 1\right)^2 |\psi(t)|^2 \\ &\geq \frac{\alpha}{5}\left(\sqrt{\frac{2}{\alpha}} - 1\right)^2 \|\psi\|^2 \\ &\geq \Omega(\|\psi\|^2) \end{aligned}$$

for sufficiently small α.

Next, consider the variance. Since $0 \leq X' \leq \left(1 + \sqrt{\frac{2}{\alpha}}\right)^2 \|\psi\|^2 \approx \frac{2}{\alpha}\|\psi\|^2$, it follows that $E[|X'|^2] \leq O\left(\frac{1}{\alpha^2}\|\psi\|^4\right)$, as desired. We can lower the variance acceptably by taking $K \leq O(1/\alpha^2)$ sufficiently large. In that case, with probability 3/4, we get an output that is at least $\Omega(\|\psi\|^2) \geq \Omega(\alpha\,\|\mathbf{A}\|^2)$. So we can take sufficiently small $\beta \geq \Omega(\alpha)$.

Thus, if we take J copies of Y, we expect at most $J/4$ of the copies to fail (where the meaning of "failure" depends on whether we are interested in a lower or upper bound). The probability that, in actuality, $J/2$ copies fail is exponentially small in J. This is at most δ for sufficiently large $J \leq O(\log(1/\delta))$. ∎

We now show how to assemble the previous steps to find a set that includes the significant frequencies in a signal. Given a signal, we form a K-shattering by convolving with a scaled, translated, and dilated Dirichlet kernel. For each new signal in the shattering, we assume that it contains exactly one dominant frequency, ω (this is a high probability event) and we then exhaustively try all possibilities for the most significant few bits of ω (this step is unnecessary if N is a power of 2). For each of these settings, we project the signal onto the space of even and odd frequencies, and estimate the norm of the results, thereby learning the least significant bit. Note that we are presented with two subspaces, say, A and B, such that one of them, say, A, is dominated by a single frequency with large norm and the other has much smaller norm. By the nature of our one-sided approximation guarantee, our estimate of the energy on A will not be too small and our estimate of the energy on B will not be too large; this suffices. As remarked above, knowing the least significant bit of ω, we may assume that bit is 0. This puts us in position to learn the next least significant bit. We repeat for other bits, from least to most significant. We output the set of frequencies we find this way.

Now we consider the settings of the parameters. We are given a parameter η and we want to find each frequency that is at least η-significant in the residual. We start with any reasonable constant α and its associated β in Lemma 11. Ultimately, we will apply Lemma 11 to each of two subspaces, A and B, to determine which has larger norm (without loss of generality, assume the energy E_A of A is larger than the energy E_B of B). Because Lemma 11 gives only a (one-sided) approximation, we need to insure that E_B is smaller than E_A by roughly the factor β. After applying Lemma 7, in each element of the shattering, E_A essentially represents $|\widehat{\mathbf{F}}(\omega_0)|^2$ and E_B represents $\|\mathbf{F}\|^2 - |\widehat{\mathbf{F}}(\omega_0)|^2$. Subsequently, in Lemma 9, E_A is decreased by a constant factor $(1-\gamma)$ and E_B is increased additively by a small constant multiple γE_A of E_A, where γ is the constant of Lemma 9. So we need to pick a value of γ in Lemma 9 to control the additive increase γE_A and then pick a value of K in Lemma 7 to accommodate *both* the additive increase γE_A and gap E_A/E_B required by Lemma 11. The dependence on η is still inverse linear, as claimed. To accomodate the increased size of the shattering, we need the failure probability of each component to be smaller than the overall failure probability, but, because the dependence on δ is logarithmic, the cost blowup is modest—the overall cost remains inverse linear in η, ignoring factors of $\log(1/\eta)$. We omit the details.

4.2 Efficiency of the Identification algorithm

We proved in the previous subsection the mathematical correctness of the Identification step. Now, we demonstrate its claimed efficiency, ignoring failure probability and precision. Note Lemmas 9 and Lemma 11 requires just $O(1)$ samples from their respective input signals. So we need just $O(1)$ samples from each signal in the shattering. To build a K-shattering of $\mathbf{A}-\mathbf{R}$, we need to sample from $\mathbf{A}-\mathbf{R}$ at $O(K)$ places on an arithmetic progression. We can sample from $\mathbf{A}$ trivially in $O(K)$ places in time $O(K)$. We do, however, need to sample from $\mathbf{A}-\mathbf{R}$ at intermediate steps of our algorithm, where $\mathbf{R}$ is an L-term representation. The naive approach requires time $\Omega(KL)$ but the following lemma gives us the efficient time $(L+K)\log^{O(1)}(N)$. Later we consider the task of building a shattering from samples; this task also can be done naively in time K^2 but we give an algorithm with cost nearly linear in K.

Lemma 12 (Bulk Sampling from a Representation) *Let* $\mathbf{R}$ *be an* L*-term representation and let* $t_k = a + bk$ *for* $0 \le k < K$ *be an arithmetic progression. In time* $(K + L)\log^{O(1)}(N)\log(1/\beta)$*, we can compute* $\mathbf{R}(t_k)$ *to precision* $\pm\beta\|\mathbf{R}\|$*, additively.*

Proof: Since $t_k = a + bk$, we can expand $\mathbf{R}(t_k)$ as

$$\frac{1}{\sqrt{N}}\sum_\ell (c_\ell e^{2\pi i a\omega_\ell/N})e^{2\pi i\omega_\ell bk/N} = \sum_\ell d_\ell f_k(b\omega_\ell), \quad \text{where } d_\ell = \frac{1}{\sqrt{N}}c_\ell e^{2\pi i a\omega_\ell/N}\text{and} f_k(\omega) = e^{2\pi i\omega k/N}.$$

Fix R to be a power of 2 at least $2\pi K$, and expand f_k in a Taylor series around rN/R for some r, getting, for $\Delta_{\ell,r} = b\omega_\ell - rN/R$,

$$\begin{aligned}\sum_\ell d_\ell f_k(b\omega_\ell) &= \sum_\ell d_\ell\left(f_k(rN/R) + f_k'(rN/R)\Delta_{\ell,r} + \frac{1}{2!}f_k''(rN/R)\Delta_{\ell,r}^2 + \cdots\right)\\ &= \sum_\ell d_\ell\left(e^{2\pi ikr/R} + \frac{2\pi ik}{N}e^{2\pi ikr/R}\Delta_{\ell,r} + \frac{1}{2!}\left(\frac{2\pi ik}{N}\right)^2 e^{2\pi ikr/R}\Delta_{\ell,r}^2 + \cdots\right)\end{aligned}$$

For each $n \ge 0$, let $a_{r,n}$ be the sum of $d_\ell\Delta_{\ell,r}^n$ over ℓ's such that $r = s$ minimizes $|b\omega_\ell - sN/R|$; that is, sum over ℓ's such that, for the given r, rN/R is the closest root of unity to $b\omega_\ell$. Take the empty sum to be zero. Thus we have

$$\sum_\ell d_\ell f_k(b\omega) = \sum_r a_{r,0}e^{2\pi ikr/R} + \frac{2\pi ik}{N}\sum_r a_{r,1}e^{2\pi ikr/R} + \frac{1}{2!}\left(\frac{2\pi ik}{N}\right)^2\sum_r a_{r,2}e^{2\pi ikr/R} + \cdots.$$

For each n, the sum $\sum_r a_{r,n}e^{2\pi ikr/R}$ can be computed, simultaneously for all k, using the FFT algorithm. We then multiply by $\frac{1}{n!}\left(\frac{2\pi ik}{N}\right)^n$ and sum over the first few n.

To analyze the rate of decay, consider the n'th term. For each ω_ℓ and the closest rN/R, we have $|\Delta_{\ell,r}| \le \frac{1}{2}\frac{N}{R}$. Since $|d_\ell| \le \|\mathbf{R}\|$, it follows that $|d_\ell\Delta_{\ell,r}^n| \le \|\mathbf{R}\|\left(\frac{N}{2R}\right)^n$. Since $k < K$ and there are at most K terms in the sum defining $a_{r,n}$, the n'th term is at most

$$\frac{K\|\mathbf{R}\|}{n!}\left(\frac{\pi K}{R}\right)^n \le \frac{K\|\mathbf{R}\|}{n!}\left(\frac{1}{2}\right)^n, \tag{2}$$

which decays exponentially in n, so the n'th tail also decays exponentially in n. It follows that only $\log(N/\beta)$ terms are needed to make the error less than $\beta\|\mathbf{R}\|$, as desired. ∎

The following lemma tells us that we can produce $O(1)$ samples from each element of the K-shattering in time $K\log^{O(1)}(N)$.

Lemma 13 *There's a sampling algorithm that takes oracle* $\mathbf{A}$ *and input* (β, t, K, N)*, where* $\beta > 0$ *and* $t \in \mathbb{Z}_N$*, such that the algorithm runs in time* $O(K\log^{O(1)}(N)\log(1/\beta))$*, and produces the vector* $(\mathbf{F}_k(t))_k$ *to precision* $\pm\beta\|\mathbf{A}\|$*, where* $(\mathbf{F}_k)_k$ *is a* K*-shattering of* $\mathbf{A}$.

Proof: Pick invertible σ at random mod N. Note that a single sample from $\mathbf{F}_k = \frac{1}{\sqrt{N}}\mathbf{A} \star \mathcal{P}_{kN/K,\sigma}\mathbf{H}_K$ at t involves K samples from $\mathbf{A}$, on an arithmetic progression with difference σ^{-1}, whose zero'th term is t. We need to modulate the samples, then sum them. Formally, for $k < K$, we have

$$\begin{aligned}\mathbf{F}_k(t) &= \frac{1}{\sqrt{N}}\left(\mathbf{A} \star \mathcal{P}_{kN/K,\sigma}\mathbf{H}_K\right)(t) = \frac{1}{K}\sum_{0\le\ell'<N}\left(\mathcal{P}_{kN/K,\sigma}\chi_{[0,K)}\right)(\ell')\mathbf{A}(t-\ell') \\ &= \frac{1}{K}\sum_{0\le\ell<N}\left(\mathcal{P}_{kN/K,\sigma}\chi_{[0,K)}\right)(\sigma\ell)\mathbf{A}(t-\sigma\ell) = \frac{1}{K}\sum_{0\le\ell<K} e^{2\pi i\ell k/K}\mathbf{A}(t-\sigma\ell).\end{aligned}$$

For each t and for fixed σ, the values $\mathbf{F}_k(t)$ comprise $1/K$ times the DFT of the values $\mathbf{A}(t-\sigma\ell)$, so they can be computed efficiently by the (usual, equally-spaced) FFT algorithm. ∎

Now, we simply apply Lemma 9 and Lemma 11 to all K of elements of the shattering in time $K\log^{O(1)}(N)$.

4.3 Precision and Failure Probability

To verify that our Identification algorithm has the advertised precision we consider the following analysis. At top level, we can abandon coefficients smaller than $\beta(\|\mathbf{A}\|^2 + \|\mathbf{R}\|^2)$, whereas our inputs are all bounded by $\|\mathbf{A}\|^2 + \|\mathbf{R}\|^2$. One can check that Lemma 12 requires precision $\pm\beta\|\mathbf{A}\|/N^{O(1)}$ in the input specification of $\mathbf{A}$ and $\mathbf{R}$ to achieve precision $\pm\beta\|\mathbf{A}\|$ or $\pm\beta\|\mathbf{R}\|$ out. Lemma 11 requires precision $\pm\|\mathbf{A}\|/N^{O(1)}$ in the input to achieve precsion $\pm O(\|\mathbf{A}\|^2)$ in the output. Lemma 9 needs the input $\mathbf{A}$ good to $\pm\|\mathbf{A}\|/N^{O(1)}$ in order that the L^2 input precision error be small, which will guarantee that neither $\widehat{\mathbf{A}}(\omega)$ nor $\|\mathbf{A}\|^2$ changes much from imprecision, so that, if ω dominates in the ideal signal, then it also dominates in the imprecise signal. Similarly, Lemma 7 requires input precision just $\pm\|\mathbf{A}\|/N^{O(1)}$. Thus, in using each of these lemmas, we need to supply $O(\log(N))$ more bits of precision than we expect out of the lemma. Thus we can perform computations to $O(\log(N/\beta))$ bits, since the depth of calls to the lemmas is bounded. (Bottom up, we sample from the representation, build a K-shattering, filter odd or even frequencies, then estimate norms.)

Finally, we consider failure probability. Sampling from $\mathbf{R}$ is deterministic. We need $\log(N)$ calls to Lemma 11, each on even/odd-filtered elements of a K-shattering. In order that all calls to Lemma 11 succeed with probability δ, we need each call to succeed with probability $\delta/\log(N)$. A K-shattering will isolate each of $R \le O(1/\eta)$ significant frequencies with probability $q \ge \Omega(1)$. We make $O(\log(R/\delta)/q) \le O(\log(1/(\eta\delta)))$ independent K-shatterings; then each such frequency is missed in all repetitions with probability at most $(1-q)^{O(\log(R/\delta)/q)} < \delta/R$. It follows that all R frequencies are found with probability at least $1-\delta$. Since the time dependence on the failure probability δ of any of these lemmas is just $O(\log(\delta))$, the overall time dependence blows up by at most logarithmic factors to achieve overall failure probability δ.

Thus, we have proven

Lemma 14 (Identification) *There's a sampling algorithm that takes input* $(\beta, \delta, \eta, L, N, \mathbf{R})$ *and oracle* $\mathbf{A}$*, where* $\beta > 0$*,* L *and* N *are powers of 2,* $\mathbf{A}$ *is a signal of length* N*,* $\eta > 0$*, and* $\mathbf{R}$ *is a formal representation of at most* L *terms, such that the algorithm runs in time* $(L + 1/\eta)\log(1/\beta)\log(1/\delta)(\log(N) + \log(1/\eta))^{O(1)}$*, and, with probability at least* $1 - \delta$*, produces a set, of size at most* $O((1/\eta)\log(1/\eta))$*, containing all frequencies* ω *for which* $|\widehat{\mathbf{A} - \mathbf{R}}(\omega)|^2 \geq \eta \|\mathbf{A} - \mathbf{R}\|^2$ *and* $|\widehat{\mathbf{A} - \mathbf{R}}(\omega)|^2 \geq \beta(\|\mathbf{A}\|^2 + \|\mathbf{R}\|^2)$*.*

5 Estimation of significant frequencies

In the previous step we generate a set of L significant frequencies. In the Estimation step, we estimate the contribution each significant frequency makes to the signal; i.e., we estimate its Fourier coefficient. We first show how to estimate a given Fourier coefficient using ap-independent random variables. These allow us to estimate several coefficients (not just one) with bulk sampling. In what follows, we define an estimator, show that the estimator approximates Fourier coefficients, then show how to compute the estimator efficiently in bulk.

Definition 15 Given frequency ω and parameters J and K, define the random variable $\widetilde{\widetilde{\mathbf{A}}}_{J,K}(\omega)$ as:

$$\widetilde{\widetilde{\mathbf{A}}}_{J,K}(\omega) = \mathrm{median}_{j \leq J} \mathrm{mean}_{k \leq K} \sqrt{N}\mathbf{A}(t_{j,k})e^{-2\pi i\omega t_{j,k}},$$

where, for each j, we independently put $t_{j,k} = a_j + b_j k$, for a_j random modulo N and b_j is a random invertible element modulo N, and the median of the set of complex numbers is performed by taking medians separately in the real and imaginary directions. ∎

Lemma 16 *For any signal* $\mathbf{A}$*, any frequency* ω*, and any parameters* $\eta > 0$ *and* $\delta > 0$*, there are a* $J \leq O(\log(1/\delta))$ *and a* $K \leq O(1/\eta)$*, such that* $\left\|\widetilde{\widetilde{\mathbf{A}}}_{J,K}(\omega) - \widehat{\mathbf{A}}(\omega)\right\|^2 \leq \eta \|\mathbf{A}\|^2$ *with probability at least* $1 - \delta$*.*

Proof: (Sketch.) Let $X = N\mathbf{A}(t)\overline{\psi_\omega(t)}$. Then $E[X] = \widehat{\mathbf{A}}(\omega)$ and $\mathrm{var}(X) \leq \|\mathbf{A}\|^2$. Use Lemma 4 to drive down the variance to $O(\eta \|\mathbf{A}\|^2)$. Then take a median of $J \leq O(\log 1/\delta)$ trials of this random variable. The standard Chernoff inequality guarantees our result. ∎

At this point, we have not discussed the computation of $\widetilde{\widetilde{\mathbf{A}}}_{J,K}(\omega)$, though it is straightforward to compute each $\widetilde{\widetilde{\mathbf{A}}}_{J,K}(\omega)$ in time $JK = \Theta(J/\eta)$. We now show that we can estimate $\Theta(K)$ Fourier coefficients at once, in time just $J\log(K)$ per coefficient. This lemma shows how to perform a major step of our algorithm in time linear in B instead of quadratic in B as the parameter K depends linearly on B at intermediate steps of our algorithm.

Lemma 17 (Bulk Estimation) *There exists a sampling algorithm that takes oracle* $\mathbf{A}$ *and takes inputs* (β, K, L, N) *and* L *frequencies* ω_l*, where* $\beta > 0$*, such that the algorithm has time* *cost* $(L{+}K)\log^{O(1)}(N)\log(1/\beta)$*, and the algorithm outputs a sample from the distribution on the vector* $\left(\widetilde{\widetilde{\mathbf{A}}}_{1,K}(\omega_\ell)\right)_\ell$ *of specified Fourier coefficient estimates to within precision* $\pm\beta\left\|\mathbf{A}\right\|$*, additively.*

Proof: Write $z_\ell = e^{-2\pi i\omega_\ell/N}$. Since $t_k = a + bk$, we have $e^{-2\pi i\omega_\ell t_k/N} = z_\ell^{a+bk}$. From the definition of $\widetilde{\widetilde{\mathbf{A}}}_{1,K}(\omega_\ell)$, we need to evaluate

$$\frac{\sqrt{N}}{K}\begin{pmatrix} z_0^a & & & & \\ & z_1^a & & 0 & \\ & & z_2^a & & \\ & 0 & & \ddots & \\ & & & & z_{L-1}^a \end{pmatrix}\left(\begin{pmatrix} 1 & (z_0^b) & (z_0^b)^2 & \cdots & (z_0^b)^{K-1} \\ 1 & (z_1^b) & (z_1^b)^2 & \cdots & (z_1^b)^{K-1} \\ \vdots & & & \ddots & \vdots \\ 1 & (z_{L-1}^b) & (z_{L-1}^b)^2 & \cdots & (z_{L-1}^b)^{K-1} \end{pmatrix}\begin{pmatrix} \mathbf{A}(t_0) \\ \mathbf{A}(t_1) \\ \vdots \\ \mathbf{A}(t_{K-1}) \end{pmatrix}\right).$$

Multiplying by the diagonal matrix $\mathrm{diag}(z_\ell^a)$ is straightforward to do in time L. The other matrix-vector product is the multipoint polynomial evaluation problem: Evaluate a given degree-K polynomial $p(z) = \sum_k \left(z_k^a\mathbf{A}(t_k)\right)z^k$ at given points points $z = z_0^b$, $z = z_1^b$, etc. We will need to compute the matrix-vector product to precision $\log(N/\beta)$ bits, so that each value is correct to $\pm\beta\left\|\mathbf{A}\right\|$.

Note that the straightforward matrix-vector product takes time LK. Classic algorithms for the multipoint polynomial evaluation problem, such as in [2], require $K\log^2(K)$ multiplications (for $K = L$), but, in general, each multiplication needs to be carried out to $\Theta(K)$ bits. For example, if $p(z) = z^K$ is evaluated at $z = 2$, then the result, 2^K, requires K bits of precision.

Fix a constant multiple R of K to be specified below.[3] We need to compute an expression of the form $f(\omega) = \sum_{0\le k<K} a_k e^{-2\pi i\omega k/N}$, for ω among the L values $b\omega_\ell$. Expand f in a Taylor series about rN/R for some r, getting, for $\Delta_r = \omega - rN/R$,

$$\begin{aligned} f(\omega) &= f(rN/R) + f'(rN/R)\Delta_r + \frac{1}{2!}f''(rN/R)\Delta_r^2 + \cdots \\ &= \sum_{0\le k<K} a_k e^{-2\pi irk/R} + \left(\frac{-2\pi i}{N}\right)\sum_{0\le k<K} a_k k e^{-2\pi irk/R}\Delta_r \\ &\qquad + \frac{1}{2!}\left(\frac{-2\pi i}{N}\right)^2\sum_{0\le k<K} a_k k^2 e^{-2\pi irk/R}\Delta_r^2\cdots \end{aligned}$$

As in Lemma 12, the terms drop off exponentially, so few terms are needed.

For each n, the expression $\sum_{0\le k<R} a_k k^n e^{-2\pi irk/R}$ can be evaluated simultaneously for all r as follows. Define $a_k = 0$ for $k \le K < R$, so we can equivalently compute $\sum_{0\le k<R} a_k k^n e^{-2\pi irk/R}$.

[3]Our algorithm, which relies on the FFT, will be faster in practice if K and R are powers of 2. One can assume that K is a power of 2 by extending the matrix with more columns and padding the column vector $\mathbf{A}(t_k)$ with zeros.

This is precisely the DFT of the sequence $a_k k^n$ of length R, so we can compute this efficiently via the FFT algorithm. For each ℓ, we can then find $r = s$ that minimizes $|\omega_\ell - sN/R|$. We then approximate $f(\omega_\ell)$ by a Taylor series at rN/R for that r. ∎

We can call this lemma J times and take a median to compute $\widetilde{\widehat{\mathbf{A}}}_{J,K}(\omega_k)$ for $J > 1$.

In the sequel, we will want a somewhat better estimate of coefficients than is guaranteed by a single call to Lemma 17. We want to bound the coefficient error in terms of the possibly much smaller quantity $\|\mathbf{A} - \mathbf{R}^*\|^2$, where $\mathbf{R}^*$ is the representation over Λ with optimal coefficients. We do this iteratively.

Lemma 18 (Iterative Bulk Estimation) *There exists a sampling algorithm that takes oracle $\mathbf{A}$ and takes inputs $(\beta, \delta, \epsilon, L, N)$ and L frequencies ω_l, where $\beta > 0$, such that the algorithm has time cost $L \log(1/\delta) \log^{O(1)}(N) \log(1/\beta)/\epsilon$, and, with probability at least $1 - \delta$, the algorithm outputs an L-term representation $\mathbf{R}$ such that, for each $\ell < L$, $|\widehat{\mathbf{A} - \mathbf{R}}(\omega_\ell)|^2 \leq (\epsilon/L) \|\mathbf{A} - \mathbf{R}^*\|^2$, where $\mathbf{R}^*$ is the optimal L-term representation, and where we assume $\|\mathbf{A} - \mathbf{R}^*\|^2 \geq \beta \|\mathbf{A}\|^2$.*

Proof: Fix $T \leq O(\log(1/\beta)/\epsilon)$, specified more below. Set $\mathbf{R}_0 = 0$. For $t < T$, repeatedly call Lemma 17 to estimate coefficients for all L terms in $\mathbf{A} - \mathbf{R}_t$, and let $\mathbf{R}_{t+1}$ be the result. We estimate each coefficient in $\mathbf{A} - \mathbf{R}_t$ so that $|\widetilde{\widehat{\mathbf{A} - \mathbf{R}}}_t(\omega) - \widehat{\mathbf{A} - \mathbf{R}}_t(\omega)|^2 \leq \frac{\epsilon}{4L} \|\mathbf{A} - \mathbf{R}_t\|^2$, so that the total square error in the L terms is at most $\frac{\epsilon}{4} \|\mathbf{A} - \mathbf{R}_t\|^2$ (and the total square error in the other $N - L$ terms is $\|\mathbf{A} - \mathbf{R}^*\|^2$). As long as $\|\mathbf{A} - \mathbf{R}_t\|^2 > (1 + \epsilon/2) \|\mathbf{A} - \mathbf{R}^*\|^2$, then the square error at the $(t+1)$'st iteration is

$$
\begin{aligned}
\|\mathbf{A} - \mathbf{R}_{t+1}\|^2 &\leq \|\mathbf{A} - \mathbf{R}^*\|^2 + \frac{\epsilon}{4} \|\mathbf{A} - \mathbf{R}_t\|^2 \\
&\leq \left(\frac{1}{1 + \epsilon/2} + \frac{\epsilon}{4} \right) \|\mathbf{A} - \mathbf{R}_t\|^2 \\
&\leq (1 - \epsilon/4) \|\mathbf{A} - \mathbf{R}_t\|^2 .
\end{aligned}
$$

If $\|\mathbf{A} - \mathbf{R}_t\|^2 > (1 + \epsilon/2) \|\mathbf{A} - \mathbf{R}^*\|^2$ for all $t < T$, then, by construction of T, we have $\|\mathbf{A} - \mathbf{R}_T\|^2 \leq (1 - \epsilon/4)^T \|\mathbf{A}\|^2 \leq \beta \|\mathbf{A}\|^2 < \|\mathbf{A} - \mathbf{R}^*\|^2$, a contradiction. So $\|\mathbf{A} - \mathbf{R}_t\|^2 \leq (1 + \epsilon/2) \|\mathbf{A} - \mathbf{R}^*\|^2$ for some $t < T$. On subsequent iterations, if $\|\mathbf{A} - \mathbf{R}_t\|^2 \leq (1 + \epsilon/2) \|\mathbf{A} - \mathbf{R}^*\|^2$, then

$$
\|\mathbf{A} - \mathbf{R}_{t+1}\|^2 < (1 + \epsilon/4)(1 + \epsilon/2) \|\mathbf{A} - \mathbf{R}^*\|^2 \leq (1 + \epsilon) \|\mathbf{A} - \mathbf{R}^*\|^2 ;
$$

if $\|\mathbf{A} - \mathbf{R}_t\|^2 > (1+\epsilon/2) \|\mathbf{A} - \mathbf{R}^*\|^2$, then $\|\mathbf{A} - \mathbf{R}_{t+1}\|^2 < \|\mathbf{A} - \mathbf{R}_t\|^2$. It follows by induction that $\|\mathbf{A} - \mathbf{R}_{t+1}\|^2 \leq (1 + \epsilon) \|\mathbf{A} - \mathbf{R}^*\|^2$, as desired. On each each iteration after iteration t, each coefficient has error at most $\frac{\epsilon}{4L}(1 + \epsilon) \|\mathbf{A} - \mathbf{R}^*\|^2 \leq \frac{\epsilon}{L} \|\mathbf{A} - \mathbf{R}^*\|^2$, as desired. ∎

6 Iteration upon Identification and Estimation

Above we showed, for any η and L, given a signal $\mathbf{A}$ as oracle, and a representation $\mathbf{R}$ of at most L terms we can, in time $(L/\epsilon + 1/\eta)(\log(N) + \log(1/\eta))^{O(1)} \log(1/\delta)\log(1/\beta)$, find a list Λ containing all frequencies ω with $|\widehat{\mathbf{A}-\mathbf{R}}(\omega)|^2 \geq \eta \|\mathbf{A}-\mathbf{R}\|^2$ and $|\widehat{\mathbf{A}-\mathbf{R}}(\omega)|^2 \geq \beta(\|\mathbf{A}\|^2 + \|\mathbf{R}\|^2)$ and, for each $\omega \in \Lambda$, estimate $\widehat{\mathbf{A}-\mathbf{R}}(\omega)$ as $\widetilde{\widehat{\mathbf{A}-\mathbf{R}}}(\omega)$ with

$$\left|\widetilde{\widehat{\mathbf{A}-\mathbf{R}}}(\omega) - \widehat{\mathbf{A}-\mathbf{R}}(\omega)\right|^2 \leq (\epsilon/L)\|\mathbf{A}-\mathbf{A}_\Lambda\|^2,$$

where $\mathbf{A}_\Lambda$ is the projection of $\mathbf{A}$ onto the frequencies in Λ. We now show how to use this repeatedly to recover an approximate representation for the signal.

In the sequel, L and $1/\eta$ can be as large as B, so a single round of identification and estimation may already take time linear in B. On the other hand, it is possible that, for some intermediate representation $\mathbf{R}$ with $\|\mathbf{A}-\mathbf{R}\|^2 \gg \|\mathbf{A}-\mathbf{R}_{\rm opt}\|^2$, a single iteration of greedy pursuit only produces a single new frequency. So a straightforward bound on the number of iterations would be B, giving a runtime bound of B^2. Instead, we will bound the number of interations independently of B (ignoring log factors), by considering the decrease in $\|\mathbf{A}-\mathbf{R}\|^2$ rather than counting the number of new frequencies.

Our goal will be a representation $\mathbf{R}$ such that $\|\mathbf{A}-\mathbf{R}\|^2$ is bounded by either $(1+\epsilon)\|\mathbf{A}-\mathbf{R}_{\rm opt}\|^2$ or $1/M^2$. Intuitively, the bound of $(1+\epsilon)\|\mathbf{A}-\mathbf{R}_{\rm opt}\|^2$ holds in the noisy case and $1/M^2$ holds when $\mathbf{A}$ is an exact superposition. In the latter case, note that the coefficients in $\mathbf{R}_{\rm opt}$ may be irrational even if the input values in $\mathbf{A}$ are integers, so a scheme that outputs an answer correct to $\pm 1/M$ (in L_2 norm, or, by replacing M with MN, in L_∞ norm) in time $\log^{O(1)}(MN)$ is considered "exact output."

Next we give our algorithm. A "formal representation" is a list of frequencies and corresponding coefficients. The algorithm will use positive constants c_0, c_1, c_2 and our analysis will use additional positive constants c_3, etc., described below.

Algorithm 19 Input: $M, N, \epsilon, B, \delta$
Oracle: $\mathbf{A}$, representing a signal of length N
Output: B-term representation, $\mathbf{R}$, such that, with probability at least $1-\delta$, we have $\|\mathbf{A}-\mathbf{R}\|^2 \leq (1+\epsilon)\|\mathbf{A}-\mathbf{R}_{\rm opt}\|^2$ or $\|\mathbf{A}-\mathbf{R}\|^2 \leq 1/M^2$.

$\mathbf{R}_0 = 0$, formally, and $\Lambda_0 = \emptyset$
$T = c_0 \log(M/\epsilon)/\epsilon^2$
for($t = 0$; $t \leq T$; t^{++}){

Find a list L of ω with $\left|\widehat{\mathbf{A}-\mathbf{R}_t}(\omega)\right|^2 \geq c_1 \frac{\epsilon^2}{B}\|\mathbf{A}-\mathbf{R}_t\|^2$.

Put $\Lambda_{t+1} = L \cup \Lambda_t$.

Estimate coefficients for $\omega \in \Lambda_{t+1}$ so that $|\widetilde{\widehat{\mathbf{A}-\mathbf{R}_{t+1}}}(\omega)|^2 \leq \frac{c_2\epsilon^2}{|\Lambda_{t+1}|+B}\left\|\mathbf{A}-\mathbf{A}_{\Lambda_{t+1}}\right\|^2$

}
Output $\mathbf{R}_T^B$, the top B terms in $\mathbf{R}_T$.

■

We now proceed with analysis. First we show that we can assume the signal is noisy. Starting with a signal, $\mathbf{A}$, add a spike (*i.e.*, noise), ν, with $\|\nu\| = \frac{\epsilon}{36M}$, in a random location, and with sign chosen so that, for any $(N/4)$-term representation $\mathbf{R}$, $\|\mathbf{A}+\nu-\mathbf{R}\| \geq \frac{\epsilon}{144M}$. Then find a $(1+\epsilon)$-factor approximation $\mathbf{R}$ to $\mathbf{A}+\nu$ and return it as an representation for $\mathbf{A}$. It is straightforward to check that $\|\mathbf{A}-\mathbf{R}\|^2$ is bounded by either $(1+\epsilon)\|\mathbf{A}-\mathbf{R}_{\text{opt}}\|^2$ or $1/M^2$, as desired. Since our sampling algorithm sees ν only with small probability, our algorithm won't change as a result of adding ν; this is a fiction for analysis only. So, henceforth, we assume that, for any $(N/4)$-term representation $\mathbf{R}$, $\|\mathbf{A}-\mathbf{R}\| \geq \frac{\epsilon}{144M}$; we show how to get a relative-error approximation to $\mathbf{A}$.

We will use Lemma 5 to find the frequencies in L. Note that Lemma 5 finds frequencies whose energy in $\mathbf{A}-\mathbf{R}_t$ is large compared with $\|\mathbf{A}-\mathbf{R}_t\|^2$ *and* large compared with $\|\mathbf{A}\|^2 + \|\mathbf{R}_t\|^2$. But, since we are assuming $\|\mathbf{A}-\mathbf{R}_t\| \geq \frac{\epsilon}{144M}$, if $\left|\widehat{\mathbf{A}-\mathbf{R}}_t(\omega)\right|^2$ is large compared with $\|\mathbf{A}-\mathbf{R}_t\|^2$, then it is automatically large compared with $\left(\frac{\epsilon}{144M}\right)^2$. Also, by our invariant, $\|\mathbf{A}\|^2 = \Theta\left(\|\mathbf{A}\|^2 + \|\mathbf{R}_t\|^2\right)$, so $\|\mathbf{A}\|^2 + \|\mathbf{R}\|^2 \leq O(M^2)$. By choosing suitably small $\beta \geq \Omega(\epsilon^2/M^4)$, we conclude that $\left|\widehat{\mathbf{A}-\mathbf{R}}_t(\omega)\right|^2 \geq \beta\left(\|\mathbf{A}\|^2 + \|\mathbf{R}_t\|^2\right)$.

Similarly, we will use Lemma 18 to estimate coefficients. Formally, Lemmata 17 and 18 require perfectly precise samples from $\mathbf{A}-\mathbf{R}_t$; it's easy to see that, since $\|\mathbf{A}-\mathbf{R}\| \geq \frac{\epsilon}{144M}$, just $O(\log(MN/\epsilon))$ bits of precision suffice for the conclusion.

We now give an informal discussion of correctness. Similar proofs have appeared, *e.g.*, in [1, 6, 10] but some care is needed to keep the overall time cost approximately linear in B rather than quadratic in B, by showing that the number of rounds is roughly independent of B and that the previous lemmata for finding frequencies and estimating coefficients can be used with parameters leading to time linear in B.

Fix an optimal representation, $\mathbf{R}_{\text{opt}} = \sum_{\omega\in\Lambda_{\text{opt}}} c_\omega\psi_\omega$. Consider the relative improvement $\frac{\|\mathbf{A}-\mathbf{R}_{t+1}\|}{\|\mathbf{A}-\mathbf{R}_t\|}$. For appropriate constant c_3, this ratio cannot be less than $(1-c_3\epsilon^2)$ for all t, since, otherwise, $\|\mathbf{A}-\mathbf{R}_T\| \leq (1-c_3\epsilon^2)^T\|\mathbf{A}-\mathbf{R}_0\| \leq \epsilon/(144M)$, a contradiction, if $T \leq O(\log(M/\epsilon)/\epsilon^2)$ is sufficiently large. (Note that, crucially, T is independent of B.) So suppose $\|\mathbf{A}-\mathbf{R}_{j+1}\| > (1-c_3\epsilon^2)\|\mathbf{A}-\mathbf{R}_j\|$.

Define $\Lambda_{\text{opt}'} = \{\omega \in \Lambda_{\text{opt}} : |c_\omega|^2 \geq (c_4\epsilon/B)\|\mathbf{A}-\mathbf{R}_{\text{opt}}\|^2\}$, for some constant c_4. Note that, if $\mathbf{R}_{\text{opt}'} = \sum_{\omega\in\Lambda_{\text{opt}'}} c_\omega\psi_\omega$, then $\|\mathbf{A}-\mathbf{R}_{\text{opt}'}\|^2 \leq (1+c_4\epsilon)\|\mathbf{A}-\mathbf{R}_{\text{opt}}\|^2$. Thus we can bound $\|\mathbf{A}-\mathbf{R}\|^2$ in terms of $\|\mathbf{A}-\mathbf{R}_{\text{opt}'}\|^2$ instead of $\|\mathbf{A}-\mathbf{R}_{\text{opt}}\|^2$.

We next claim that $\Lambda_{\text{opt}'} \subseteq \Lambda_{j+1} \subseteq \Lambda_T$. First, toward a contradiction, suppose $\|\mathbf{A}-\mathbf{R}_j\|^2 > 2\|\mathbf{A}-\mathbf{R}_{\text{opt}'}\|^2$. Then, ignoring the coefficient approximation error in $\mathbf{R}_j$ (the total square error is much less than $\|\mathbf{A}-\mathbf{R}_j\|^2$), we conclude that $\Lambda_{\text{opt}'}\setminus\Lambda_j$ has energy at least $\|\mathbf{A}-\mathbf{R}_j\|^2 - \|\mathbf{A}-\mathbf{R}_{\text{opt}'}\|^2 \geq (1/2)\|\mathbf{A}-\mathbf{R}_j\|^2$. An iteration of greedy pursuit finds frequencies ω and good approximations to coefficients c_ω if $|c_\omega|^2 \geq (c_1\epsilon^2/B)\left\|\mathbf{A}-\mathbf{R}_j\right\|^2$; it follows that most of the energy in $\Lambda_{\mathrm{opt}'} \setminus \Lambda_j$ is found, so that $\left\|\mathbf{A}-\mathbf{R}_{j+1}\right\|^2 \leq (3/4)\left\|\mathbf{A}-\mathbf{R}_j\right\|^2$, contradicting the definition of j. So suppose $\left\|\mathbf{A}-\mathbf{R}_j\right\|^2 \leq 2\left\|\mathbf{A}-\mathbf{R}_{\mathrm{opt}'}\right\|^2$. Then $\omega \in \Lambda_{\mathrm{opt}'}$ satisfies $|c_\omega|^2 \geq (c_4\epsilon/B)\left\|\mathbf{A}-\mathbf{R}_{\mathrm{opt}}\right\|^2 \geq (c_4\epsilon/(2B))\left\|\mathbf{A}-\mathbf{R}_j\right\|^2$, so we recover such ω on the next iteration of greedy pursuit.

Our estimates of coefficients are good enough to be ignored from the perspective of overall error. So our output $\mathbf{R}$ is suboptimal because it has a suboptimal choice of frequencies (which can be due to tiny errors in approximation of coefficients). Suppose $\omega \in \Lambda_{\mathrm{opt}'}$ is displaced by $\omega_* \in \Lambda_T^B$, and suppose $\widetilde{c}_{\omega_*}$ is the coefficient of ω_* in our output. Then the error attributed to this exchange is $E_x = |\widetilde{c}_{\omega_*} - c_{\omega_*}|^2 + |c_\omega|^2 - |c_{\omega_*}|^2$. By goodness of approximation, $|\widetilde{c}_{\omega_*} - c_{\omega_*}|^2 \leq (c_2\epsilon^2/B)\left\|\mathbf{A}-\mathbf{R}_{\mathrm{opt}}\right\|^2$. Next, because $\omega \in \Lambda_{\mathrm{opt}'}$, it follows that $|\widetilde{c}_{\omega_*} - c_{\omega_*}|^2 \leq (c_2\epsilon/c_4)|c_\omega|^2$. Thus it follows, if constants are chosen properly, that $E_x \leq (1+\epsilon)|c_\omega|^2 - |c_{\omega_*}|^2$, and we need to show that E_x is small compared with the unavoidable error, $|c_{\omega_*}|^2$. Because we chose ω_* over ω, we have $|\widetilde{c}_{\omega_*}|^2 \geq |\widetilde{c}_\omega|^2$, which is at least $(|c_\omega| - |c_\omega - \widetilde{c}_\omega|)^2$ by the triangle inequality. By goodness of approximation and because $\omega \in \Lambda_{\mathrm{opt}'}$, this is at least $(|c_\omega| - \sqrt{c_2\epsilon/c_4}|c_\omega|)^2 = (1 - c_5\sqrt{\epsilon})|c_\omega|^2$. It follows that $E_x \leq c_6\sqrt{\epsilon}|c_{\omega_*}|^2$. By replacing ϵ by ϵ^2/c_6^2, we conclude that each displacement of an optimal frequency by a suboptimal one increases its contribution to the error by the factor $(1+\epsilon)$, so the combined contributions blow up the error by at most the same factor. We conclude that $\left\|\mathbf{A}-\mathbf{R}_T^B\right\|^2 \leq (1+\epsilon)\left\|\mathbf{A}-\mathbf{R}_{\mathrm{opt}}\right\|^2$; *i.e.*, the algorithm is correct.

7 Higher Dimensional Versions

In this section, we show how to generalize our algorithm to more than one dimension. Suppose that the time and frequency domain is $N_1 \times N_2 \times \cdots \times N_d$, where $N = \prod_j N_j$. Let $\mathcal{G}$ denote this additive group. For some values of the N_j's, we can directly reduce the multidimensional version to the one-dimensional version (*e.g.*, all N_j's co-prime); in other cases, the generalization is more involved.

In higher dimensions, the basis functions are of the form $\psi_{\omega_1,\omega_2,\ldots,\omega_d}$, defined by

$$\psi_{\omega_1,\omega_2,\ldots,\omega_d}(t_1,t_2,\ldots,t_d) = \frac{1}{\sqrt{N}} e^{2\pi i(\omega_1 t_1 + \omega_2 t_2 + \cdots + \omega_d t_d)/N}.$$

In general, the algorithm above for one-dimensional signals can be generalized to higher dimensions in a straightforward way, but the result will not be efficient. In several places we give up a constant factor in one dimension that becomes the factor $2^{O(d)}$ in higher dimensions. While this may be acceptable for $d \leq 3$, other aspects are much worse. For example, the multidimensional Taylor expansion involves the n'th derivative in d dimensions, which is an object of size d^n. (Even if one consolidates equal partial derivatives like $\frac{\partial^2}{\partial x \partial y}$ and $\frac{\partial^2}{\partial y \partial x}$, the n'th derivative in d dimensions has size $\binom{n+d-1}{d-1} \approx n^d/d!$, which is potentially much larger than $2^{O(d)}$, since n may be as large as $\log(MN/\epsilon)$, which may be much larger than a constant.

7.1 Co-prime Case

If all the N_j's are relatively prime, we can reduce the d-dimensional problem of size $N_1 \times N_2 \times \cdots \times N_d$ to a one-dimensional problem of length N. This is because the group $\mathbb{Z}_{N_1} \times \mathbb{Z}_{N_2}$ is isomorphic to $\mathbb{Z}_{N_1 N_2}$. Algorithmically, we proceed as follows. Choose a line that exhausts the time domain, say $(t_1, t_2, \ldots, t_d) = t(1, 1, 1, \ldots, 1)$, for $0 \le t < N$. It follows that any multidimensional basis function $\psi_{(\omega_1, \omega_2, \ldots, \omega_d)}$ transforms to the one dimensional basis function $\psi_{(\omega_1 + \omega_2 + \cdots + \omega_d)}$ on this line. The map $\omega \mapsto (\omega_1 + \omega_2 + \cdots + \omega_d)$ is defined by $\omega_j = \omega \bmod N_j$; the inverse map is given by the Chinese Remainder Theorem and has a fast implementation.

Thus, given a multidimensional signal $\mathbf{A}$, we can construct a one dimensional signal $\mathbf{B}(t) = \mathbf{A}(t \bmod N_1, t \bmod N_2, \ldots, t \bmod N_j)$. Find a B-term one-dimensional representation for $\mathbf{B}$. With the result, do one round of B instances of the Chinese Remainder Theorem to lift each term in the one-dimensional representation to its multidimensional equivalent.

This also covers a common practical case. Often the data is available on the continuous domain $[0, 1)^d$ and the signal is known to have a good representation over frequencies at most N' where each frequency is close to an integer multiple of $1/N'$, for some large enough N'. We can then just pick d co-prime values around N' for the values N_j and model the problem discretely.

7.2 General—a Constant Factor Approximation

In this section, we give an algorithm for the general multidimensional problem in time

$$(d \log(N) \log(M) \log(1/\delta)/\epsilon)^{O(1)}.$$

First we give an overview. We'll need somewhat stronger variants of some of the above lemmas, which we'll prove in this section. We'll assume that there are at least $\log(B)$ dimensions by introducing extra dummy dimensions of length N, if necessary. Then the overall algorithm proceeds as follows, similar to [10].

We assume that the signal has a B-term representation with energy at least $\epsilon \|\mathbf{A}\|^2$. Otherwise, zero is an acceptable output and, while we may output something slightly worse than zero, our ability to estimate coefficients is good enough that coefficient estimation error can be ignored. It follows that, using Lemma 25 (a stronger version of Lemma 11), we can get an estimate of the signal energy. For this discussion, assume $\|\mathbf{A}\|^2 = 1$.

Fix an optimal representation, $\mathbf{R}_{\text{opt}}$. As in [10], we will track a collection of orthogonal subspaces. In each subspace X, we may assume that the terms in $\mathbf{R}_{\text{opt}}$ that lie in X have total energy at least $\Omega(\epsilon)$ times the energy of X. This is because, if we discard subspaces with lower SNR, we will have discarded at most, say, $(\epsilon/2)E_X$ energy of the optimal representation, where E_X is the energy of X. Summing over all such (disjoint) subspaces, we discard $\sum_X (\epsilon/2)E_X = (\epsilon/2)\sum_X E_X \le (\epsilon/2)$. Since $\|\mathbf{A} - \mathbf{R}_{\text{opt}}\|^2 \ge \epsilon$, we can still recover at least $\frac{1}{2}\|\mathbf{A} - \mathbf{R}_{\text{opt}}\|^2$, which suffices for geometric improvement.

Similarly, we want to keep the subspaces with energy at least ϵ/B, since they may contain useful signal. If a subspace has less energy, our estimate of the energy will tell us the energy is less, so we'll discard it, and keep only at most B/ϵ subspaces. If a subspace has sufficient energy *and* has some portion $\mathbf{R}_{\text{opt}}$ with proportional energy, then Lemma 25 will insure that we keep the subspace. (We may or may not discard a subspace with high energy but no large element of $\mathbf{R}_{\text{opt}}$.)

To analyze a subspace, we'll break it into two halves (say, even and odd frequencies along some dimension) and estimate the norm of each half. Breaking the space in half entails just constant blowup. By Lemma 25, to analyze a subspace X, we need a number of samples proportional to the energy E_X in X, namely, BE_X samples, up to factors involving only the logarithm of B, etc.

We'll want to analyze several subspaces in bulk. We may assume all subspaces have the same energy within a factor of 2, by rounding energies up to a factor of 2 and repeating for each power of 2. Each subspace corresponds to the settings of some of the bits of the identity of a frequency in some one dimension j, all of the bits set in some other dimensions, and none of the bits set in the remaining dimensions. In dimension j, we will exhaustively try all settings of a few of the most significant bits. We will then learn the other bits adaptively, from least significant to most significant. We may assume that all of the subspaces are working on the same dimension j, have the same setting for the most significant bits, and the same *number*, k, of least significant bits specified. Our immediate task, then, is to learn the $k+1$'s least significant bit in each subspace.

Suppose each of these subspaces has energy E_X. Then there are at most $1/E_X$ of these subspaces. We will show how to get one sample from each of the subspaces in cost roughly $1/E_X$. Then we can get the required BE_X samples in total cost roughly B. By "making a sample from a subspace X," we mean that we will project the signal in such a way that subspace X passes with no attenuation and everything orthogonal to X gets attenuation factor ϵE_X or smaller. It follows that we can ignore everything but X in the result.

If the specified bits in the subspaces indicated that the subspaces were spaced regularly, then we could make $1/E_X$ regularly-spaced samples and do an FFT. Instead, in general, we need to do an unequally-spaced discrete Fourier transform, which can be done efficiently by generalizing Lemma 12 (below).

Once we've identified frequencies, we need to estimate their coefficients. We do this by generalizing Lemma 2 properly.

In the remaining sections, we prove the necessary lemmas.

7.2.1 Variance Reduction

It turns out that we *can* make true pairwise-independent samples from a construction that supports quick bulk operations. Although the construction in this section is simpler than ap-independence in the sense that no new proof of variance reduction is required, the algorithms for efficient bulk operations are more involved and build on the algorithms for the ap-independent case, given previously.

Definition 20 Given two sets A and B from a group, $\mathcal{G}$, let $A+B$ denote $\{a+b : a \in A, b \in B\}$. An *edge* in $\mathcal{G}$ is any two-point subset of $\mathcal{G}$. A k-dimensional *parallelepiped* is the sum of k edges. (The elements are also called *corners.*) A random parallelepiped is the sum of random edges (with replacement). ∎

Note that the dimension of the parallelepiped may be greater or less than the dimension of $\mathcal{G}$. For example, if $\mathcal{G}$ is one-dimensional and edge A is $\{3,5\}$ and edge B is $\{6,17\}$, then $A+B$ is $\{3+6, 3+17, 5+6, 5+17\} = \{9, 11, 20, 22\}$.

The following lemma is immediate:

Lemma 21 *The 2^k corners of a random k-dimensional parallelepiped are pairwise independent.*

Also note that we can subtract off one element from each edge, essentially moving one corner of the parallelepiped to the origin. For example, the above parallelepiped is $3+6+\{0+0, 0+11, 2+0, 2+11\}$. Thus we can take each edge to be a two-point set of the form $\{0,a\}$, and, if the a's are random, then the 2^k-1 corners other than the origin are pairwise independent.

7.2.2 Multivariate Multipoint Polynomial Evaluation

We can view bulk operations at the corners of a parallelepiped as multivariate polynomial evaluation or interpolation. For example, the above parallelepiped corresponds immediately to the univariate polynomial $ax^9+bx^{11}+cx^{20}+dx^{22}$. Putting $u=x^{5-3}$ and $v=x^{17-6}$ (where the exponents are read from the edges), we can rewrite the polynomial as $x^9(a+bu+cv+duv)$. To evaluate this polynomial at $2^k=2^2$ different values for x, we evaluate x^9 at these points by repeated squaring, evaluate the k intermediate variables u and v, then, as we show next, evaluate the (2^k)-term k-variable multilinear polynomial at 2^k points in time $2^k \cdot k^{O(1)}$, quadratically faster than the naive 2^{2k}-time algorithm. This suffices as a generalization of Lemma 12; a similar technique generalizes Lemma 17. We omit the details.

First we need a fast algorithm for the Hadamard transform; *i.e.*, a Fourier transform on the Boolean cube.

Definition 22 The order 2^k Hadamard transform is the k-fold tensor product of $\begin{pmatrix} +1 & +1 \\ +1 & -1 \end{pmatrix}$. Thus the (r,c) position is given by (-1) raised to the dot product of the binary expansions of r and c. ∎

The tensor product definition immediately suggests an algorithm with cost $k2^k$ for evaluating the Hadamard transform, which amounts to evaluating a k-variate multilinear polynomial on the corners of the Boolean cube. We now show how to evaluate the polynomial on a parallelepiped.

Lemma 23 *We can evaluate a k-variate multilinear polynomial at 2^k places on $\mathbb{T}^k$ in time $2^k \cdot k^{O(1)}$.*

Here $\mathbb{T}^k$ is the k-dimensional Torus: the set of k-tuples of unit-norm complex numbers.

Proof: (Sketch.) Denote the polynomial and variables by $p(x_1, \ldots, x_k)$ and the set of points as S.

Note that if the points in S were given by $((-1)^{\alpha_1}, (-1)^{\alpha_2}, \ldots, (-1)^{\alpha_k})$ for $\alpha_j \in \{0,1\}$ ("grid points"), then the evaluation would be a Boolean Fourier transform (Hadamard transform), for which there are efficient algorithms. Instead, we'll evaluate p and a few derivatives of p at these points, then expand using Newton's theorem for finding q given p and a such that $q(x-a) = p(x)$. (This is a degenerate form of Taylor's theorem; a Taylor expansion of a degree-k multilinear p at $x = a$ has just k terms and we'll have time to find them all.)

Unfortunately, a straightforward generalization of Lemma 17 is too expensive, since the n'th derivative in d dimensions is an object of size d^n or, after exploiting symmetry, size $\binom{d+n-1}{d-1}$. Instead we do the following.

Choose a bijection between the points in S and the grid points. Find the values of the first two derivatives in the first two dimensions, $p, p_{x_1} = \frac{\partial p}{\partial x_1}, p_{x_2} = \frac{\partial p}{\partial x_2}$, and $p_{x_1 x_2} = \frac{\partial^2 p}{\partial x_1 \partial x_2}$ at each grid point. For each order of derivative, the values for all 2^k grid points can be found simultaneously in time $O(k2^k)$ by the fast Hadamard transform algorithm. Use p_{x_1} and p_{x_2} to find the value of the first derivative at each grid point g in the direction of its assigned point, $s(g)$, projected down to $\pi(s(g))$ in the plane of the first two dimensions. Each such value is the dot product of the total derivative (p_{x_1}, p_{x_2}) with the unit vector pointing from g to $\pi(s(g))$. Similarly, find all higher directional derivatives in the direction $g \to \pi(s(g))$. Then discard the axis-aligned derivatives, keeping just the first n derivatives in the appropriate direction. For each order of derivative, compute the coefficients of a polynomial taking the given values at the the grid points. This interpolation step is an inverse fast Hadamard transform and can also be done in time $k2^k$.

In general, after j steps, our inductive hypothesis is that we have up to n polynomials, each one associated with an order of a directional derivative, where the direction at each grid point g is toward the projection of the assigned point $s(g)$ onto the span of the first j dimensions. (Note that, because the original polynomial was multilinear, all derivatives beyond the j'th are zero. So there are $j+1$ polynomials, not $k+1$ polynomials.) Each polynomial, when evaluated at a grid point, gives us the value of the directional derivative; the direction itself is encoded in the bijection $s : \mathcal{G} \to S$. Take a symbolic derivative of each polynomial in the $(j+1)$'st dimension; this gives $2(j+1)$ polynomials. Evaluate each of the $2(j+1)$ polynomials at all the grid points in bulk, in total time $O(jk2^k)$. For each order of derivative, at each grid point, we now have a total derivative in the two-dimensional space spanned by $g \to s(g)$ projected into the first j dimensions and the $(j+1)$'st dimension. Take a linear combination to get the value of the directional derivative in the direction of $g \to s(g)$ projected onto the first $j+1$ dimensions. Finally, recover the coefficients of a polynomial corresponding to those values. This gives the inductive hypothesis at $j+1$. ∎

The total cost of this algorithm is $O(k^3 \cdot 2^k)$, since there are k iterations (one for each dimension), $j \le k$ derivatives, and each Hadamard transform takes time $O(k2^k)$. More precisely, the time is dominated by $O(k^2)$ fast Hadamard transforms, where, in our algorithm, $2^k \approx B$. Note that our one-dimensional algorithm Lemma 17 required n FFT's, where n is the number of terms in the Taylor expansion, roughly $\log(MN/\epsilon)$. It is conceivable that the algorithm of this section is better even in the one-dimensional setting, since $\log^2(B)$ may be less than $\log(MN/\epsilon)$.

7.2.3 Norm Estimation

Next we turn to norm estimation. We use essentially the same estimator from Definition 10, but with different parameter settings.

Definition 24 For integers J and K, and signal $\mathbf{A}$, define the estimator $\|\mathbf{A}\|^2_{\sim J,K}$ (or $\|\mathbf{A}\|^2_{\sim}$, if J and K are understood) as follows. Choose JK points t at random, regarded as J groups of K points, such that the points in each group are the corners of a random parallelepiped and all the parallelepipeds are fully independent of each other. Take the mean over each group of $N|\mathbf{A}(t)|^2$. Take the median of the means. Finally, divide by 4. ∎

Lemma 25 (Norm Estimation) *For all constants $\alpha > 0$ and all $\delta > 0$ there's a constant $\beta > 0$, and integer $J \le (\log(1/\delta))$ and $K \le O(B/\alpha^2)$ such that the following hold probability at least $1-\delta$:*

- $\|\mathbf{A}\|^2_{\sim} \le \|\mathbf{A}\|^2$ *(for any $\mathbf{A}$);*
- *if $\|\psi\|^2 \ge \alpha \|\mathbf{A}\|^2$ for some B-term superposition ψ, then $\|\mathbf{A}\|^2_{\sim} \ge \beta \|\mathbf{A}\|^2$.*

Proof: We modify the proof of Lemma 11. The upper bound is similar, so we focus on the lower bound.

Write $\mathbf{A} = \psi + \nu$, where $\langle \psi, \nu \rangle = 0$. Assuming $\alpha \le 1$, define ν' by

$$\nu'(t) = \begin{cases} \nu(t), & |\nu(t)|^2 \le \frac{2B}{\alpha N} \|\psi\|^2\,; \\ 0, & \text{otherwise.} \end{cases}$$

Let $X'(t) = N|\psi(t) + \nu'(t)|^2$. As above, it suffices to show that $E[X'] \ge \Omega(\|\psi\|^2)$ and $E[|X'|^2] \le O(B\|\psi\|^2)$. Then, by taking $K \le O(B)$ large enough, it follows, say, that $E[|Y'|^2] \le \frac{1}{4}\|\psi\|^2$. We can then choose β appropriately to get our result.

First, consider $E[X']$. Let $T = \{t : |\nu(t)|^2 > \frac{2B}{\alpha N}\|\psi\|^2\}$. Suppose that $|T| \le \alpha N/(5B)$. Then

$$\begin{aligned} E[X'] &= \|\psi + \nu - \mathbf{A}\chi_T\|^2 \\ &\ge \|\psi\|^2 - 2|\langle \psi, \nu - \mathbf{A}\chi_T\rangle| \\ &\ge \|\psi\|^2 - 2|\langle \psi, \mathbf{A}\chi_T\rangle|, \end{aligned}$$

since $\langle \psi, \nu \rangle = 0$. Thus

$$\begin{aligned} E[X'] &\ge \|\psi\|^2 - 2|\langle \psi, \mathbf{A}\chi_T\rangle| \\ &\ge \|\psi\|^2 - 2|\langle \psi\chi_T, \mathbf{A}\rangle| \\ &\ge \|\psi\|^2 - 2\|\psi\chi_T\|\,\|\mathbf{A}\| \\ &\ge \|\psi\|^2 - 2\sqrt{\frac{\alpha}{5}}\,\|\psi\|\,\|\mathbf{A}\|, \end{aligned}$$

because $|\psi(t)|^2$ is at most $\frac{B}{N}\|\psi\|^2$, so at most $\alpha N/(5B)$ positions from at least N/B contribute. Thus

$$\begin{aligned} E[X'] &\geq \|\psi\|^2 - 2\sqrt{\frac{\alpha}{5}}\,\|\psi\|\,\|\mathbf{A}\| \\ &\geq \left(1-\frac{2}{5}\right)\|\psi\|^2, \end{aligned}$$

since $\|\mathbf{A}\| \geq \frac{1}{\sqrt{\alpha}}\|\psi\|$.

On the other hand, if $|T| > \alpha N/(5B)$, then

$$\begin{aligned} E[X'] &\geq \sum_{t\in T}|\mathbf{A}(t)|^2 \\ &\geq \sum_{t\in T}\left(\sqrt{\frac{2B}{\alpha N}}-1\right)^2\|\psi\|^2 \\ &\geq \frac{\alpha N}{5B}\left(\sqrt{\frac{2B}{\alpha N}}-1\right)^2\|\psi\|^2 \\ &\geq \Omega(\|\psi\|^2) \end{aligned}$$

for sufficiently small α.

Next, consider the variance. We have

$$\begin{aligned} E[|X'|^2] &= \frac{1}{N}\sum_t (N|\psi(t)+\nu'(t)|^2)^2 \\ &\leq O(N)\sum_t\left(|\psi(t)|^4+|\nu'(t)|^4\right), \end{aligned}$$

using the inequality $|ab| \leq |a|^2+|b|^2$. By the Cauchy-Schwarz inequality, $\max_t|\psi(t)|^2 \leq \frac{B}{N}\|\psi\|^2$ and, by construction, $\max_t|\nu'(t)|^2 \leq \frac{2B}{\alpha N}\|\psi\|^2$. It follows as in Lemma 11 that

$$E[|X'|^2] \leq O\left(\frac{B}{\alpha^2}\|\psi\|^4\right).$$

By making $K \leq O(B/\alpha^2)$ large enough, we can guarantee that each group succeeds with probability at least 3/4.

As in Lemma 11, if we take J copies of Y, we expect at most $J/4$ of the copies to fail (where the meaning of "failure" depends on whether we are interested in a lower or upper bound). The probability that, in actuality, $J/2$ copies fail is exponentially small in J. This is at most δ for sufficiently large $J \leq O(\log(1/\delta))$. ∎

Note that a direct reduction to Lemma 11 will not work. That would proceed as follows. Some element of ψ has energy at least $(1/B)\|\psi\|^2 \approx \|\mathbf{A}\|^2$, so, putting $\alpha = 1/B$, we then make $K \approx B^2$. This would have unacceptable cost B^2 instead of B.

8 Conclusion

We provided a sampling algorithm that yields, with high probability, a B-term Fourier representation $\mathbf{R}$ for any input signal $\mathbf{A}$ of length N, with the guarantee that $\|\mathbf{A}-\mathbf{R}\|_2^2$ is within a factor $(1+\epsilon)$ of the best possible B-term Fourier representation. The algorithm samples

$$B \cdot \text{poly}(\log N, \log \|\mathbf{A}\|, 1/\epsilon)$$

positions non-adaptively and spends time and space linear in this quantity. Preliminary implementations of our algorithm indicate that, for small B and $\epsilon = 1$, our algorithm is more efficient than an optimized, publicly-available FFT package for N approximately 1 million.

The overall structure of this algorithm follows previous work [6, 10, 1], but we have to apply two key ideas: bulk estimation of multipoint polynomial evaluation using an unequally-spaced Fourier tranform, and use of arithmetic-progression independent random variables to enable the iterative algorithm. As a result we improve the $B^{\geq 4}$ factor in previous results to being linear in B. More recent work [11, 9] solves the multipoint polynomial evaluation problem generally in limited precision, *e.g.*, using *displacement rank*. We give a simple, self-contained solution for our special case of z_ℓ's on the unit circle.

Acknowledgments

For helpful discussions, we thank: Ingrid Daubechies, Björn Enquist, and Jing Zou. Emmanuel Candès [4] privately communicated to us his recent independent, unpublished discovery of some of the techniques for unequally-spaced Fourier transforms that we use for bulk coefficient estimation.

References

[1] A. Akavia, S. Goldwasser and S. Safra. Proving hard-core predicates by list decoding. Proc. IEEE FOCS, 2003, 146–157.

[2] A. Aho, J. Hopcroft, and J. Ullman. *Design and analysis of algorithms.* Addison-Wesley, 1972.

[3] G. Box, G. Jenkins, and G. Reinsel. *Time Series Analysis: Forecasting and Control.* Prentice Hall, 1994.

[4] E. Candès. *private communication.*

[5] FFTW. `http://www.fftw.org`.

[6] A. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, and M. J. Strauss. Near optimal sparse Fourier representation via sampling. Proc. ACM STOC, 2002.

[7] J. Goodman. *Introduction to Fourier Optics.* McGraw-Hill, 1996.

[8] D. Gottlieb and S. Orszag. Numerical Analysis of Spectral Methods: Theory and Applications. CMBS-NSF Regional Conference Series in Applied Mathematics 26, SIAM, 1977.

[9] E. Kaltofen and L. Yagati. Improved Sparse Multivariate Polynomial Interpolation Algorithms. International Symposium on Symbolic and Algebraic Computation. Proc. ISSAC 1988, Springer-Verlag LNCS 358, 1988, pp. 467–474.

[10] Y. Mansour. Randomized interpolation and approximation of sparse polynomials. *SIAM J. Computing*, 24-2, 1995.

[11] V. Olshevsky. Fast algorithms for structured matrices: Theory and applications. Contemporary Mathematics, Vol 323, American Math Soc., RI, 2003.

[12] H. J. Weaver. *Applications of Discrete and Continuous Fourier Analysis*. Wiley, 1983.

[13] Jing Zou. *personal communication.*