

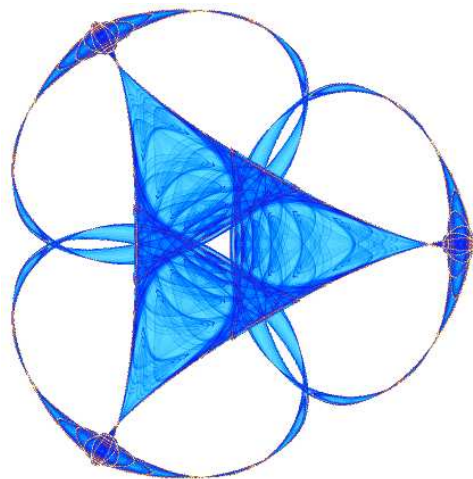
**IMPROVED APPROXIMATION GUARANTEES FOR
SUBLINEAR-TIME FOURIER ALGORITHMS**

By

M.A. Iwen

IMA Preprint Series # 2337

(September 2010)



INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS

UNIVERSITY OF MINNESOTA
400 Lind Hall
207 Church Street S.E.
Minneapolis, Minnesota 55455-0436

Phone: 612-624-6066 Fax: 612-626-7370

URL: <http://www.ima.umn.edu>

IMPROVED APPROXIMATION GUARANTEES FOR SUBLINEAR-TIME FOURIER ALGORITHMS

M. A. IWEN
 DUKE UNIVERSITY, BOX 90320
 DURHAM, NC 27708-0320
 EMAIL: MARKIWEN@MATH.DUKE.EDU

ABSTRACT. In this paper modified variants of the sparse Fourier transform algorithms from [14] are presented which improve on the approximation error bounds of the original algorithms. In addition, simple methods for extending the improved sparse Fourier transforms to higher dimensional settings are developed. As a consequence, approximate Fourier transforms are obtained which will identify a near-optimal k -term Fourier series for any given input function, $f : [0, 2\pi]^D \rightarrow \mathbb{C}$, in $O(k^2 \cdot D^4)$ time (neglecting logarithmic factors). Faster randomized Fourier algorithm variants with runtime complexities that scale linearly in the sparsity parameter k are also presented.

1. INTRODUCTION

This paper develops fast methods for finding near-optimal nonlinear approximations to the Fourier transform of a given function $f : [0, 2\pi]^D \rightarrow \mathbb{C}$. Suppose that f is a bandlimited function so that $\hat{f} \in \mathbb{C}^{N^D}$, where N^D is large. An optimal k -term trigonometric approximation to f is given by

$$(1) \quad f_k^{\text{opt}}(\vec{x}) = \sum_{j=1}^k \hat{f}(\vec{\omega}_j) e^{i\vec{\omega}_j \cdot \vec{x}}$$

where $\vec{\omega}_1, \dots, \vec{\omega}_{N^D} \in [1 - N/2, N/2]^D \cap \mathbb{Z}^D$ are ordered by the magnitudes of their Fourier coefficients so that

$$|\hat{f}(\vec{\omega}_1)| \geq |\hat{f}(\vec{\omega}_2)| \geq \dots \geq |\hat{f}(\vec{\omega}_{N^D})|.$$

The optimal k -term approximation error is then $\|f - f_k^{\text{opt}}\|_2 = \|\hat{f} - \hat{f}_k^{\text{opt}}\|_2$. Suppose $k \in \mathbb{N}$ is given. The goal of this paper is to develop Fourier approximation schemes that are guaranteed to always return a near-optimal trigonometric polynomial, $y_k : [0, 2\pi]^D \rightarrow \mathbb{C}^{N^D}$, having $\|f - y_k\|_2 \approx \|f - f_k^{\text{opt}}\|_2$. Furthermore, we require that the developed schemes are fast, with runtime complexities that scale polylogarithmically in N^D and at most quadratically in k . Such Fourier algorithms will then be able to accurately approximate the Fourier transform of an arbitrarily given function (i.e., with no a priori assumptions regarding “smoothness”) much more quickly than a standard Fast Fourier Transform (FFT) methods [8, 3] whenever $N^D \gg k$ is large. More specifically, the developed schemes will lead to Fourier approximation algorithms with runtime complexities that scale *polynomially* in D , as opposed to *exponentially*.

The Fourier approximation techniques developed in this paper are improvements of the techniques introduced in [14]. As an example, suppose for simplicity that $f : [0, 2\pi] \rightarrow \mathbb{C}$ is a bandlimited function of only one variable so that $\hat{f} \in \mathbb{C}^N$. Furthermore, let $k < N$ be given. The main theorem in [14] implicitly proves that $O(k^2 \log^4 N)$ function evaluations and runtime are sufficient to produce a sparse approximation, \hat{y}_k , to \hat{f} satisfying

$$\|\hat{f} - \hat{y}_k\|_2 \leq \|\hat{f} - \hat{f}_k^{\text{opt}}\|_2 + 3\sqrt{k} \|\hat{f} - \hat{f}_k^{\text{opt}}\|_1,$$

where f_k^{opt} is defined as in Equation 1. This error bound is unsatisfying for several reasons. Principally, if many of the Fourier coefficients of f are roughly the same magnitude the approximation error above can actually increase with k , the number of nonzero terms in the sparse approximation \hat{y}_k . If nothing else, we would like to improve these error guarantees so that additional computational effort can always be counted on to yield better sparse Fourier approximations.

Let $p, q \in [1, \infty)$. We will say that $\vec{y} \in \mathbb{C}^N$ satisfies an $l^p, l^q/k^{1/q-1/p}$ error bound with respect to $\hat{f} \in \mathbb{C}^N$ if

$$(2) \quad \|\hat{f} - \vec{y}\|_p \leq \|\hat{f} - \hat{f}_k^{\text{opt}}\|_p + \frac{\|\hat{f} - \hat{f}_k^{\text{opt}}\|_q}{k^{1/q-1/p}}.$$

More generally, we will refer to any error bound of the form given in Equation 2 as an *instance optimal error bound* for \hat{f} . In this paper the result discussed in the previous paragraph is improved by showing that $O(k^2 \log^4 N)$ function samples and runtime are sufficient to produce a sparse approximation satisfying an $l^2, l^1 / \sqrt{k}$ error bound with respect to the Fourier transform of any N -bandlimited function $f : [0, 2\pi] \rightarrow \mathbb{C}$. This decreases the “ $\sqrt{k} \|\hat{f} - \hat{f}_k^{\text{opt}}\|_1$ ” term in the previous error bound [14] by a multiplicative factor of k . Furthermore, faster randomized methods are also presented which are capable of achieving the same type of approximation errors typically achieved by slower algorithms based on the restricted isometry property [6, 11] with high probability, despite utilizing a similar number of function samples.

1.1. Results and Related Work. Over the past few years, results concerning matrices with the Restricted Isometry Property (RIP) have allowed methods to be developed which can accurately approximate the Fourier transform of a function despite being given access to only a very small number of samples. Informally, an $m \times N$ matrix \mathcal{M} has the RIP of order $k \in \mathbb{N}$ if it acts as a near isometry for all vectors, $\vec{x} \in \mathbb{C}^N$, which contain at most k nonzero entries. Particularly important for our purposes is that RIP matrices of order $2k$ serve as good measurement matrices for sparsely approximating vectors in \mathbb{C}^N . Suppose \mathcal{M} is an $m \times N$ matrix with the RIP of order $2k$. Then, for any $\vec{x} \in \mathbb{C}^N$, a variety of computational methods including l^1 -minimization [4, 5, 6], Orthogonal Matching Pursuit [27, 17], Regularized Orthogonal Matching Pursuit [20, 21], Iterative Hard Thresholding [2], etc., will take $\mathcal{M}\vec{x}$ as input and subsequently output another vector, $\vec{y} \in \mathbb{C}^N$, satisfying an instance optimal error bound with respect to \vec{x} (e.g., an $l^2, l^1 / \sqrt{k}$ error bound). Hence, any linear operator satisfying an appropriate RIP condition can serve as an efficient measurement operator capable of capturing sufficient information about any input vector in order to allow it to be accurately approximated.

The most pertinent RIP result to approximate Fourier recovery as considered here states that a rectangular matrix constructed by randomly selecting a small set of rows from an $N \times N$ inverse discrete Fourier transform matrix will have the RIP with high probability. The following theorem was proven in [26] and subsequently generalized and improved in [24].

Theorem 1. (See [26]). *Suppose we select m rows uniformly at random from the rescaled $N \times N$ Inverse Discrete Fourier Transform (IDFT) matrix $\frac{1}{\sqrt{m}}\Psi^{-1}$, where*

$$(\Psi^{-1})_{i,j} = \frac{e^{\frac{2\pi i \cdot i \cdot j}{N}}}{\sqrt{N}},$$

and form the $m \times N$ submatrix \mathcal{M} . If m is $\Omega(k \cdot \log N \cdot \log^2 k \cdot \log(k \log N))$ then \mathcal{M} will have the RIP of order k with high probability.

Let Ψ be the $N \times N$ Discrete Fourier Transform (DFT) matrix defined by $\Psi_{i,j} = \frac{1}{N} \cdot e^{\frac{2\pi i \cdot i \cdot j}{N}}$, $f : [0, 2\pi] \rightarrow \mathbb{C}$ be a given function, and $\vec{f} \in \mathbb{C}^N$ be the vector of N equally spaced samples from f on $[0, 2\pi]$. In this case Theorem 1 tells us that collecting the m function samples determined by $\mathcal{M}\Psi\vec{f}$ will be sufficient to accurately approximate the discrete Fourier transform of \vec{f} with high probability. More precisely, if $\mathcal{M}\hat{\vec{f}} = \mathcal{M}\Psi\vec{f}$ is input to a recovery algorithm known as CoSaMP [19] the following theorem holds.

Theorem 2. (See [19]). *Suppose that \mathcal{M} is a $m \times N$ measurement matrix formed by selecting $m = \Theta(k \cdot \log^4 N)$ rows from the $N \times N$ IDFT matrix, Ψ^{-1} , uniformly at random. Furthermore, assume that \mathcal{M} satisfies the RIP of order $2k^1$. Fix precision parameter $\eta \in \mathbb{R}$ and let $\vec{U} = \mathcal{M}\Psi\vec{f}$ be measurements collected for any given $\vec{f} \in \mathbb{C}^N$. Then, when executed with \vec{U} as input, CoSaMP will output a $2k$ -sparse vector, $\vec{y} \in \mathbb{C}^N$, satisfying*

$$\left\| \hat{\vec{f}} - \vec{y} \right\|_2 \leq \text{Const} \cdot \max \left\{ \eta, \frac{1}{\sqrt{k}} \cdot \left\| \hat{\vec{f}} - \hat{\vec{f}}_k^{\text{opt}} \right\|_1 \right\},$$

where $\hat{\vec{f}}_k^{\text{opt}}$ is a best possible k -term approximation for $\hat{\vec{f}} = \Psi\vec{f}$. The required runtime is $O(N \log N \cdot \log(\|\hat{\vec{f}}\|_2 / \eta))$.

¹Note that this is true with high probability by Theorem 1.

Fourier Result	w.h.p./D	Runtime	Function Samples	Error Guarantee
Theorem 6	D	$O(N \cdot k \cdot \log^2 N)$	$O(k^2 \cdot \log^2 N)$	$l^2, l^1 / \sqrt{k}$
CoSaMP [19]	\approx D	$O(N \cdot \log N)$	$O(k \cdot \log^4 N)$ [26, 11]	$l^2, l^1 / \sqrt{k} + \eta$
Corollary 3	w.h.p.	$O(N \cdot \log N)$	$O(k \cdot \log^2 N)$	$l^2, l^1 / \sqrt{k}$
Theorem 7	D	$O(k^2 \cdot \log^4 N)$	$O(k^2 \cdot \log^4 N)$	$l^2, l^1 / \sqrt{k}$
Sparse Fourier [13]	w.h.p.	$O(k \cdot \log^{O(1)}(N))$	$O(k \cdot \log^{O(1)}(N))$	$l^2, l^2 + \eta$
Corollary 4	w.h.p.	$O(k \cdot \log^5 N)$	$O(k \cdot \log^4 N)$	$l^2, l^1 / \sqrt{k}$
Optimal Algorithm	D	$\Omega(k)$	$\Omega(k)$ [7]	$l^2, l^1 / \sqrt{k}$

TABLE 1. Sparse Fourier Approximation Algorithms with Robust Recovery Guarantees

In effect, Theorem 2 promises that CoSaMP will locate $2k$ of the dominant entries in \hat{f} if given access to $\Theta(k \cdot \log^4 N)$ samples from f . If \hat{f} contains $2k$ significant frequencies whose Fourier coefficients collectively dominate all others combined, then these most significant frequencies will be found and their Fourier coefficients will be well approximated. If \hat{f} has no dominant set of $2k$ entries then CoSaMP will return a sparse representation which is guaranteed only to be trivially bounded. However, in such cases sparse Fourier approximation is a generally hopeless task anyways and a bounded, albeit poor, sparse representation is the best one can expect. In any case, as long as the random function samples correspond to a matrix with the RIP, CoSaMP will output a vector satisfying an instance optimal error bound with respect to \hat{f} . However, the required runtime will always be $\Omega(N)$. More generally, all existing Fourier recovery methods based on RIP conditions have superlinear runtime complexity in N .

Other existing Fourier algorithms for approximating $\hat{f} \in \mathbb{C}^N$ given sampling access to $\vec{f} \in \mathbb{C}^N$ work by utilizing random sampling techniques [12, 13]. These approaches simultaneously obtain both instance optimal error guarantees, and runtime complexities that scale sublinearly in N . However, they generally also require more function samples than recovery algorithms which utilize matrices satisfying the RIP. A variant of the following Fourier sampling theorem, concerning the sparse approximation of \hat{f} provided sampling access to $f : [0, 2\pi] \rightarrow \mathbb{C}$, is proven in [13].

Theorem 3. (See [13]). *Fix precision parameters $\eta, \tau \in \mathbb{R}^+$ and probability parameter $\lambda \in (0, 1)$. There exists a randomized sampling algorithm which, when given sampling access to an input signal $\vec{f} \in \mathbb{C}^N$, outputs a k -sparse representation \vec{y} for \hat{f} satisfying*

$$\left\| \hat{f} - \vec{y} \right\|_2 \leq \sqrt{1 + \tau} \cdot \max \left\{ \eta, \left\| \hat{f} - \hat{f}_k^{\text{opt}} \right\|_2 \right\}$$

with probability at least $1 - \lambda$. Here \hat{f}_k^{opt} is a best possible k -sparse representation for \hat{f} . Both the runtime and sampling complexities are bounded above by

$$k \cdot \left(\log \left(\frac{1}{\lambda} \right), \log \left(\frac{1}{\eta} \right), \log \|A\|_2, \log N, \frac{1}{\tau} \right)^{O(1)}.$$

It is important to note that the probabilistic guarantee of recovering an accurate sparse representation provided by Theorem 3 is a nonuniform *per signal* guarantee. In contrast, Fourier approximation procedures which rely on RIP matrices provide uniform probability guarantees *for all* possible input vectors. If a set of sample positions corresponds to an $N \times N$ IDFT submatrix with the RIP property, those sample positions will allow the accurate Fourier approximation of all possible input vectors $\vec{f} \in \mathbb{C}^N$.

In this paper several Fourier algorithms are developed which obtain instance optimal approximation guarantees while also improving on various aspects of the previously mentioned approaches. See Table 1 for a comparison of the results obtained herein with Theorems 2 and 3 when applied to the problem of approximating the Fourier transform, $\hat{f} \in \mathbb{C}^N$, of an N -bandwidth function $f : [0, 2\pi] \rightarrow \mathbb{C}$. The first column of Table 1 lists the Fourier results considered, while the second column lists whether the recovery algorithm in question guarantees an instance optimal

output Deterministically (D), or With High Probability (w.h.p.) per signal. Note that CoSaMP² has an “ $\approx D$ ” listed in its second column. This denotes that the RIP results utilized in Theorem 2 provide a uniform probability guarantee, although no explicit constructions of RIP matrices satisfying these bounds are currently known. The third and fourth columns of Table 1 contain the sampling and runtime complexities of the algorithms, respectively. For simplicity some of the bounds were simplified by ignoring precision parameters, etc.³. Finally, the fifth column of Table 1 lists the instance optimal approximation guarantees achievable by each algorithm when budgeted the number of samples and time listed in the third and fourth columns. The “ $+\eta$ ” in the CoSaMP and Sparse Fourier rows remind us that their error bounds are good up to an additive precision parameter.

The last row of Table 1 lists lower bounds for the runtime and sampling complexity of any algorithm guaranteed to achieve an instance optimal $l^2, l^1 / \sqrt{k}$ Fourier approximation error (see [7]). Note that all six approaches have sampling complexities containing additional multiplicative logarithmic factors of N beyond the stated lower sampling bound⁴. The lowest overall sampling complexity is achieved by Corollary 3, although, it is achieved at the expense of a weak nonuniform “w.h.p.” approximation probability guarantee. Similarly, Corollary 4 improves on the previous sampling complexity of the sparse Fourier algorithm in [13] while at least matching its runtime complexity⁵. Finally, to the best of the author’s knowledge, Theorem 7 obtains the best available runtime of any existing deterministic Fourier approximation algorithm which is guaranteed to achieve an instance optimal error guarantee.

The remainder of this paper is organized as follows: In Section 2 the notation utilized throughout the remainder of the paper is established. Next, in Section 3, a number theoretic matrix construction is presented and analyzed. Section 3.1 explains how random submatrices of the presented number theoretic matrices can yield nonuniform probabilistic approximation guarantees, while Section 3.2 outlines a useful relationship between these matrices and the Fourier transform of a periodic function. In Section 4 the matrices defined in Section 3 are used to construct Fourier approximation algorithms with runtime complexities that scale superlinearly in N (i.e., Theorem 6 and Corollary 3 are proven). Next, in Section 5, the algorithms of Section 4 are modified into algorithms with runtime complexities that scale sublinearly in N (i.e., Theorem 7 and Corollary 4 are proven). In Section 6 a simple strategy is given for extending the results of the previous two sections to higher dimensional Fourier transforms. Finally, a short conclusion is presented in Section 7.

2. NOTATION AND SETUP

Below we will consider any function whose domain, I , is both ordered and countable to be a vector. Let $\vec{x}: I \rightarrow \mathbb{C}$. In this case we will say that $\vec{x} \in \mathbb{C}^{|I|}$, and that $x_i = \vec{x}(i) \in \mathbb{C}$ for all $i \in I$. We will denote the l^p norm of any such vector, \vec{x} , by

$$\|\vec{x}\|_p = \left(\sum_{i \in I} |x_i|^p \right)^{\frac{1}{p}}, \quad \text{for } p \in [1, \infty).$$

If \vec{x} is an infinite vector (i.e., if I is countably infinite), we will say that $\vec{x} \in l^p$ if $\|\vec{x}\|_p$ is finite. Without loss of generality, we will assume that a given $\vec{x} \in \mathbb{C}^N$ is indexed by $I = [0, N) \cap \mathbb{Z}$ unless indicated otherwise. The vector $\vec{1}_N \in \mathbb{C}^N$ will always denote the vector of N ones, and $\vec{0}_N \in \mathbb{C}^N$ will always denote the vector of N zeros.

For any given $\vec{x} \in \mathbb{C}^{|I|}$ and subset $S \subseteq I$, we will let $\vec{x}_S \in \mathbb{C}^{|I|}$ be equal to \vec{x} on S and be zero everywhere else. Thus,

$$(\vec{x}_S)_i = \begin{cases} x_i & \text{if } i \in S, \\ 0 & \text{otherwise} \end{cases}.$$

Furthermore, for a given integer $k < |I|$, we will let $S_k^{\text{opt}} \subset I$ be the first k element subset of I in lexicographical order with the property that $|x_s| \geq |x_t|$ for all $s \in S_k^{\text{opt}}$ and $t \in I - S_k^{\text{opt}}$. Thus, S_k^{opt} contains the indexes of the k of the largest magnitude entries in \vec{x} . Finally, we will define \vec{x}_k^{opt} to be $\vec{x}_{S_k^{\text{opt}}}$, a best k -term approximation to \vec{x} .

²We used CoSaMP as a representative for all RIP based recovery algorithms because, for the purposes of Table 1 at least, it matches the currently best achievable runtime, sampling, and error bound performance characteristics of all the other previously mentioned RIP-based methods in the Fourier setting.

³The $O(N \cdot \log N)$ runtime listed for Corollary 3 will hold if k is $O(N / \log^2 N)$. More generally, the runtime will always be $O(N \cdot \log^3 N)$.

⁴Both the sampling and runtime complexities of the Sparse Fourier algorithm presented in [13] scale like $\Omega(k \cdot \log^5 N)$.

⁵It must be remembered, however, that the algorithm presented in [13] enjoys a stronger approximation error guarantee up to its additive/multiplicative precision parameters.

In this paper we will be considering methods for approximating the Fourier series of an arbitrarily given periodic function, $f : [0, 2\pi]^D \rightarrow \mathbb{C}$. Following convention, we will denote the Fourier transform of f by $\hat{f} : \mathbb{Z}^D \rightarrow \mathbb{C}$, where

$$\hat{f}(\vec{\omega}) = \frac{1}{(2\pi)^D} \int_{\vec{x} \in [0, 2\pi]^D} e^{-i\vec{\omega} \cdot \vec{x}} f(\vec{x}) d\vec{x} \text{ for all } \vec{\omega} \in \mathbb{Z}^D.$$

Note that \hat{f} can be considered an infinite vector indexed by \mathbb{Z}^D . We also have the inverse relationship

$$f(\vec{x}) = \sum_{\vec{\omega} \in \mathbb{Z}^D} \hat{f}(\vec{\omega}) e^{i\vec{\omega} \cdot \vec{x}} \text{ for all } \vec{x} \in [0, 2\pi]^D.$$

Thus, we learn f in the process of approximating its Fourier transform.

Call each $\vec{\omega} \in \mathbb{Z}^D$ a *Fourier mode* or *frequency*, and $\hat{f}(\vec{\omega})$ its corresponding *Fourier coefficient*. Ultimately, we will restrict our attention to the Fourier modes of f inside some finite bandwidth. We will do this by identifying, and then estimating the Fourier coefficients of, the most energetic Fourier modes in $\left(-\left\lceil \frac{N}{2} \right\rceil, \left\lfloor \frac{N}{2} \right\rfloor\right]^D \cap \mathbb{Z}^D$ for a given bandwidth value $N \in \mathbb{N}$. Toward this end, define the vector $\vec{f} \in \mathbb{C}^{N^D}$ by

$$\hat{f}_{\vec{\omega}} = \hat{f}(\vec{\omega}) \text{ for all } \vec{\omega} \in \left(-\left\lceil \frac{N}{2} \right\rceil, \left\lfloor \frac{N}{2} \right\rfloor\right]^D \cap \mathbb{Z}^D.$$

Similarly, define $\tilde{f} : \mathbb{Z}^D \rightarrow \mathbb{C}$ to be the Fourier transform of the related optimal bandlimited approximation to f . More precisely, let

$$\tilde{f}(\vec{\omega}) = \hat{f}_{\left(-\left\lceil \frac{N}{2} \right\rceil, \left\lfloor \frac{N}{2} \right\rfloor\right]^D \cap \mathbb{Z}^D} = \begin{cases} \hat{f}(\vec{\omega}) & \text{if } \vec{\omega} \in \left(-\left\lceil \frac{N}{2} \right\rceil, \left\lfloor \frac{N}{2} \right\rfloor\right]^D \cap \mathbb{Z}^D, \\ 0 & \text{otherwise} \end{cases}$$

for all $\omega \in \mathbb{Z}^D$. We will approximate f by approximating \tilde{f} . However, in order to do so we must first construct a special class of matrices.

3. A SPECIALIZED MEASUREMENT MATRIX CONSTRUCTION

We consider $m \times N$ measurement matrices, $\mathcal{M}_{s_1, K}$, constructed as follows. Select K pairwise relatively prime integers beginning with a given $s_1 \in \mathbb{N}$ and denote them by

$$(3) \quad s_1 < \dots < s_K.$$

Produce a row $r_{j,h}$, where $j \in [1, K] \cap \mathbb{N}$ and $h \in [0, s_j] \cap \mathbb{N}$, in $\mathcal{M}_{s_1, K}$ for each possible residue of each s_j integer. The n^{th} entry of each $r_{j,h}$ row, $n \in [0, N] \cap \mathbb{N}$, is given by

$$(4) \quad \begin{aligned} (r_{j,h})_n &= \delta((n-h) \bmod s_j) \\ &= \begin{cases} 1 & \text{if } n \equiv h \pmod{s_j} \\ 0 & \text{otherwise} \end{cases}. \end{aligned}$$

We then set

$$(5) \quad \mathcal{M}_{s_1, K} = \begin{pmatrix} r_{1,0} \\ r_{1,1} \\ \vdots \\ r_{1,s_1-1} \\ \vdots \\ r_{K,s_K-1} \end{pmatrix}.$$

The result is an $(m = \sum_{j=1}^K s_j) \times N$ matrix with binary entries. See Figure 1 for an example measurement matrix.

The matrices constructed above using relatively prime integers have many useful properties. As we shall see later in Section 4, these properties cumulatively allow the accurate recovery of Fourier sparse signals. We require two additional definitions before we may continue. Let $n \in [0, N] \cap \mathbb{N}$. We define $\mathcal{M}_{s_1, K, n}$ to be the $K \times N$ matrix created

$$\begin{array}{l}
\mathbf{n} \in [0, N) \cap \mathbb{N} \\
\mathbf{n} \equiv 0 \pmod{2} \\
\mathbf{n} \equiv 1 \pmod{2} \\
\mathbf{n} \equiv 0 \pmod{3} \\
\mathbf{n} \equiv 1 \pmod{3} \\
\mathbf{n} \equiv 2 \pmod{3} \\
\vdots \\
\mathbf{n} \equiv 1 \pmod{5} \\
\vdots
\end{array}
\begin{array}{c}
\mathbf{0} \quad \mathbf{1} \quad \mathbf{2} \quad \mathbf{3} \quad \mathbf{4} \quad \mathbf{5} \quad \mathbf{6} \quad \dots \\
\left(\begin{array}{cccccccc}
1 & 0 & 1 & 0 & 1 & 0 & 1 & \dots \\
0 & 1 & 0 & 1 & 0 & 1 & 0 & \dots \\
1 & 0 & 0 & 1 & 0 & 0 & 1 & \dots \\
0 & 1 & 0 & 0 & 1 & 0 & 0 & \dots \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & \dots \\
& & & & \vdots & & & \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & \dots \\
& & & & \vdots & & &
\end{array} \right)
\end{array}$$

FIGURE 1. An Example Matrix created using $s_1 = 2, s_2 = 3, s_3 = 5, \dots$

by selecting the K rows of $\mathcal{M}_{s_1, K}$ with nonzero entries in the n^{th} column. Furthermore, we define $\mathcal{M}'_{s_1, K, n}$ to be the $K \times (N-1)$ matrix created by deleting the n^{th} column of $\mathcal{M}_{s_1, K, n}$. Thus, we have

$$(6) \quad \mathcal{M}_{s_1, K, n} = \begin{pmatrix} r_{1, n \bmod s_1} \\ r_{2, n \bmod s_2} \\ \vdots \\ r_{K, n \bmod s_K} \end{pmatrix}$$

and

$$(7) \quad \mathcal{M}'_{s_1, K, n} = \begin{pmatrix} (r_{1, n \bmod s_1})_0 & (r_{1, n \bmod s_1})_1 & \dots & (r_{1, n \bmod s_1})_{n-1} & (r_{1, n \bmod s_1})_{n+1} & \dots & (r_{1, n \bmod s_1})_{N-1} \\ (r_{2, n \bmod s_2})_0 & (r_{2, n \bmod s_2})_1 & \dots & (r_{2, n \bmod s_2})_{n-1} & (r_{2, n \bmod s_2})_{n+1} & \dots & (r_{2, n \bmod s_2})_{N-1} \\ \vdots & & & & & & \\ (r_{K, n \bmod s_K})_0 & (r_{K, n \bmod s_K})_1 & \dots & (r_{K, n \bmod s_K})_{n-1} & (r_{K, n \bmod s_K})_{n+1} & \dots & (r_{K, n \bmod s_K})_{N-1} \end{pmatrix}.$$

We have the following two lemmas.

Lemma 1. *Let $n, \bar{k} \in [0, N) \cap \mathbb{N}$ and $\vec{x} \in \mathbb{C}^{N-1}$. Then, at most $\bar{k} \lceil \log_{s_1} N \rceil$ of the K entries of $\mathcal{M}'_{s_1, K, n} \cdot \vec{x}$ will have magnitude greater than or equal to $\|\vec{x}\|_1 / \bar{k}$.*

Proof:

We have that

$$\left| \left\{ j \mid |(\mathcal{M}'_{s_1, K, n} \cdot \vec{x})_j| \geq \frac{\|\vec{x}\|_1}{\bar{k}} \right\} \right| \leq \frac{\bar{k}}{\|\vec{x}\|_1} \|\mathcal{M}'_{s_1, K, n} \cdot \vec{x}\|_1 \leq \bar{k} \cdot \|\mathcal{M}'_{s_1, K, n}\|_1$$

by the Markov Inequality. Focusing now on $\mathcal{M}'_{s_1, K, n}$ we can see that

$$(8) \quad \|\mathcal{M}'_{s_1, K, n}\|_1 = \max_{l \in [0, N-1) \cap \mathbb{N}} \sum_{j=1}^K |(\mathcal{M}'_{s_1, K, n})_{j,l}| = \max_{l \in [0, N-1) \cap \mathbb{N}} \sum_{j=1}^K \delta((n-l) \bmod s_j) \leq \lceil \log_{s_1} N \rceil$$

by the Chinese Remainder Theorem (see [22]). The result follows. \square

Lemma 2. *Let $n, \bar{k} \in [0, N) \cap \mathbb{N}$, $S \subset [0, N) \cap \mathbb{N}$ with $|S| \leq \bar{k}$, and $\vec{x} \in \mathbb{C}^{N-1}$. Then, $\mathcal{M}'_{s_1, K, n} \cdot \vec{x}$ and $\mathcal{M}'_{s_1, K, n} \cdot (\vec{x} - \vec{x}_S)$ will differ in at most $\bar{k} \lceil \log_{s_1} N \rceil$ of their K entries.*

Proof:

We have that

$$\left| \left\{ j \mid (\mathcal{M}'_{s_1, K, n} \cdot \vec{x})_j \neq (\mathcal{M}'_{s_1, K, n} \cdot (\vec{x} - \vec{x}_S))_j \right\} \right| = \left| \left\{ j \mid (\mathcal{M}'_{s_1, K, n} \cdot \vec{x}_S)_j \neq 0 \right\} \right| \leq \left| \left\{ j \mid (\mathcal{M}'_{s_1, K, n} \cdot (\vec{\mathbb{1}}_{N-1})_S)_j \geq 1 \right\} \right|$$

since all the entries of $\mathcal{M}'_{s_1, K, n}$ are nonnegative integers. Applying Lemma 1 with $\vec{x} = (\vec{\mathbb{1}}_{N-1})_S$ and $\vec{k} = \left\| (\vec{\mathbb{1}}_{N-1})_S \right\|_1 = |S|$ finishes the proof. \square

Combining these two Lemmas we obtain a general theorem concerning the accuracy with which we can approximate any entry of an arbitrary complex vector $\vec{x} \in \mathbb{C}^N$ using only entries of $\mathcal{M}_{s_1, K} \cdot \vec{x}$.

Theorem 4. *Let $n, k, s_1 \in [0, N) \cap \mathbb{N}$, $\epsilon^{-1} \in \mathbb{N}^+$, $c \in [2, \infty) \cap \mathbb{N}$, and $\vec{x} \in \mathbb{C}^N$. Set $K = c \cdot (k/\epsilon) \lfloor \log_{s_1} N \rfloor + 1$. Then, more than $\frac{c-2}{c} \cdot K$ of the K entries of $\mathcal{M}_{s_1, K, n} \cdot \vec{x}$ will estimate x_n to within $\frac{\epsilon \cdot \left\| \vec{x} - \vec{x}_{(k/\epsilon)}^{\text{opt}} \right\|_1}{k}$ precision.*

Proof:

Define $\vec{y} \in \mathbb{C}^{N-1}$ to be $\vec{y} = (x_0, x_1, \dots, x_{n-1}, x_{n+1}, \dots, x_{N-1})$. We have that

$$\mathcal{M}_{s_1, K, n} \cdot \vec{x} = x_n \cdot \vec{\mathbb{1}}_K + \mathcal{M}'_{s_1, K, n} \cdot \vec{y}.$$

Applying Lemma 2 with $\vec{k} = (k/\epsilon)$ reveals that at most $(k/\epsilon) \lfloor \log_{s_1} N \rfloor$ entries of $\mathcal{M}'_{s_1, K, n} \cdot \vec{y}$ differ from $\mathcal{M}'_{s_1, K, n} \cdot (\vec{y} - \vec{y}_{(k/\epsilon)}^{\text{opt}})$. Of the remaining $K - (k/\epsilon) \lfloor \log_{s_1} N \rfloor$ entries of $\mathcal{M}'_{s_1, K, n} \cdot \vec{y}$, at most $(k/\epsilon) \lfloor \log_{s_1} N \rfloor$ will have magnitudes greater than or equal to $\epsilon \left\| \vec{y} - \vec{y}_{(k/\epsilon)}^{\text{opt}} \right\|_1 / k$ by Lemma 1. Hence, at least

$$K - 2(k/\epsilon) \lfloor \log_{s_1} N \rfloor \geq (c-2)(k/\epsilon) \lfloor \log_{s_1} N \rfloor + 1 > \frac{c-2}{c} \cdot K$$

entries of $\mathcal{M}'_{s_1, K, n} \cdot \vec{y}$ will have a magnitude no greater than

$$\frac{\epsilon \cdot \left\| \vec{y} - \vec{y}_{(k/\epsilon)}^{\text{opt}} \right\|_1}{k} \leq \frac{\epsilon \cdot \left\| \vec{x} - \vec{x}_{(k/\epsilon)}^{\text{opt}} \right\|_1}{k}.$$

The result follows. \square

We will now study the number of rows, $m = \sum_{j=1}^K s_j$, in our measurement matrix under the Theorem 4 assumption that $K = c \cdot (k/\epsilon) \lfloor \log_{s_1} N \rfloor + 1$ for some constant integer $c \in [2, \infty)$ and given values of $s_1 = (k/\epsilon), N \in \mathbb{N}^+$. Given this assumption concerning K , we wish to bound the smallest possible sum, m , resulting from all possible choices of pairwise relatively prime s_j values. We will do this by bounding m for one particular set of s_j values.

Let p_l be the l^{th} prime natural number. Thus, we have

$$(9) \quad p_1 = 2, p_2 = 3, p_3 = 5, p_4 = 7, \dots$$

Next, define $q \in \mathbb{N}$ so that

$$(10) \quad p_{q-1} < (k/\epsilon) \leq p_q.$$

We will use the first K primes no smaller than (k/ϵ) to define our relatively prime s_j values for the purposes of bounding m . Hence, for the remainder of Section 3 we will have

$$(11) \quad s_1 = \frac{k}{\epsilon} \leq p_q < s_2 = p_{q+1} < \dots < s_K = p_{q+K-1}.$$

It follows from results in [15] that

$$(12) \quad m = \sum_{j=1}^K s_j \leq \sum_{j=0}^{K-1} p_{q+j} = \frac{p_{q+K}^2}{2 \ln p_{q+K}} \cdot \left(1 + O\left(\frac{1}{\ln p_{q+K}} \right) \right) - \frac{p_q^2}{2 \ln p_q} \cdot \left(1 + O\left(\frac{1}{\ln p_q} \right) \right).$$

Furthermore, the Prime Number Theorem (see [22]) tells us that

$$q = \frac{k}{\epsilon \cdot \ln(k/\epsilon)} \left(1 + O\left(\frac{1}{\ln(k/\epsilon)} \right) \right)$$

and

$$p_q = \frac{k}{\epsilon} \left(1 + O\left(\frac{\ln \ln(k/\epsilon)}{\ln(k/\epsilon)}\right) \right).$$

Thus, if we use $K = c \cdot (k/\epsilon) \lceil \log_{(k/\epsilon)} N \rceil + 1$ in order to construct $\mathcal{M}_{(k/\epsilon),K}$ we will have

$$q + K = \frac{c \cdot k \lceil \log_{(k/\epsilon)} N \rceil}{\epsilon} \left(1 + O\left(\frac{1}{\ln N}\right) \right).$$

Here we have assumed that $(k/\epsilon) + K$ is less than N . Applying the Prime Number Theorem once more we have that

$$(13) \quad p_{q+K} = \frac{c \cdot k \lceil \log_{(k/\epsilon)} N \rceil \cdot \ln\left(\frac{k \ln N}{\epsilon}\right)}{\epsilon} \left(1 + O\left(\frac{\ln \ln\left(\frac{k \ln N}{\epsilon}\right)}{\ln\left(\frac{k \ln N}{\epsilon}\right)}\right) \right).$$

Utilizing Equation 12 now yields

$$(14) \quad m \leq \sum_{j=0}^{K-1} p_{q+j} = \frac{c^2 \cdot k^2 \lceil \log_{(k/\epsilon)} N \rceil^2 \cdot \ln\left(\frac{k \ln N}{\epsilon}\right)}{2\epsilon^2} \left(1 + O\left(\frac{\ln \ln\left(\frac{k \ln N}{\epsilon}\right)}{\ln\left(\frac{k \ln N}{\epsilon}\right)}\right) \right).$$

Hence, we have an asymptotic upper bound for the number of rows in $\mathcal{M}_{(k/\epsilon),K}$. The next theorem, proven in Appendix A, provides a concrete upper bound.

Theorem 5. *Suppose that $N, k, \epsilon^{-1} \in \mathbb{N} - \{1\}$ with $N > k \geq 2$. Then, if we set $K = c \cdot (k/\epsilon) \lceil \log_{s_1} N \rceil + 1$ for some constant integer $c \in [2, \infty)$, there exists an $m \times N$ measurement matrix, $\mathcal{M}_{s_1, K}$, with a number of rows*

$$m < \frac{3(c + 1.89)^2 \cdot k^2 \lceil \log_{(k/\epsilon)} N \rceil^2}{4 \cdot \epsilon^2} \cdot \ln\left(\frac{(c + 1.89) \cdot k \lceil \log_{(k/\epsilon)} N \rceil}{\epsilon}\right).$$

Tighter upper bounds for the number of rows may be explicitly calculated using Equations 35 – 38 below.

Proof: See Appendix A. \square

Theorems 4 and 5 collectively provide bounds for the number of rows a measurement matrix $\mathcal{M}_{s_1, K}$ may contain and still be able to estimate any entry of a vector $\vec{x} \in \mathbb{C}^N$ to within a precision proportional to $\left\| \vec{x} - \vec{x}_{(k/\epsilon)}^{\text{opt}} \right\|_1$. These bounds are universal in that they pertain to measurement matrices which are guaranteed to provide accurate estimates for all entries of all vectors $\vec{x} \in \mathbb{C}^N$. In the next section we will prove the existence of a small number of $\mathcal{M}_{s_1, K}$ rows which are guaranteed to provide precise estimates for any sufficiently small set of vector entries. We will also briefly consider a randomized matrix construction based on uniformly sampling rows of the deterministic $\mathcal{M}_{s_1, K}$ matrices considered above. These results will ultimately motivate the development of sparse Fourier transforms with reduced sampling requirements.

3.1. Randomized Row Sampling and Existence Results. In this section we will consider submatrices of the $m \times N$ measurement matrices, $\mathcal{M}_{s_1, K}$, discussed above. More specifically, we will be discussing matrices formed by selecting a small number of rows from an $\mathcal{M}_{s_1, K}$ matrix as follows. Let $\tilde{S} = \{s_{j_1}, s_{j_2}, \dots, s_{j_l}\}$ be a subset of the s_j values used to form $\mathcal{M}_{s_1, K}$ (see Equations 3 – 5). We will then define $\mathcal{M}_{\tilde{S}}$ to be the $(\tilde{m} = \sum_{\tilde{h}=1}^l s_{j_{\tilde{h}}}) \times N$ matrix,

$$(15) \quad \mathcal{M}_{\tilde{S}} = \begin{pmatrix} r_{j_1, 0} \\ r_{j_1, 1} \\ \vdots \\ r_{j_1, s_{j_1}-1} \\ \vdots \\ r_{j_l, s_{j_l}-1} \end{pmatrix},$$

with each row defined as per Equation 4. Finally, for $n \in [0, N) \cap \mathbb{N}$, we define $\mathcal{M}_{\tilde{S}, n}$ to be the $l \times N$ matrix,

$$(16) \quad \mathcal{M}_{\tilde{S}, n} = \begin{pmatrix} r_{j_1, n \bmod s_{j_1}} \\ r_{j_2, n \bmod s_{j_2}} \\ \vdots \\ r_{j_l, n \bmod s_{j_l}} \end{pmatrix},$$

along the lines of Equation 6. The following corollary of Theorem 4 demonstrates the existence of small submatrices of $\mathcal{M}_{s_1, K}$ capable of providing accurate approximations to any given subset of a given vector $\vec{x} \in \mathbb{C}^N$.

Corollary 1. *Let $k, N, \epsilon^{-1} \in \mathbb{N}$, $S \subseteq [0, N) \cap \mathbb{N}$, and $\vec{x} \in \mathbb{C}^N$. Set $K = c \cdot (k/\epsilon) \lceil \log_{s_1} N \rceil + 1$ for $s_1 \in \mathbb{N}$ and a constant integer $c \in [4, \infty)$. Form an $m \times N$ measurement matrix $\mathcal{M}_{s_1, K}$ as per Section 3. Then, there exists a subset of $O(\log |S|)$ s_j values for $\mathcal{M}_{s_1, K}$,*

$$\tilde{S} = \left\{ s_{j_1}, s_{j_2}, \dots, s_{j_{\lceil \log_{(c/2)}(|S|+1) \rceil}} \right\},$$

with the following property: For all $n \in S$ we have

$$\min_{s_{j_h} \in \tilde{S}} \left| \left(\mathcal{M}_{\tilde{S}, n} \vec{x} - x_n \cdot \vec{1}_{\lceil \log_{(c/2)}(|S|+1) \rceil} \right)_h \right| \leq \frac{\epsilon \cdot \left\| \vec{x} - \vec{x}_{(k/\epsilon)}^{\text{opt}} \right\|_1}{k}.$$

Proof:

We proceed by induction on the size of $S \subseteq [0, N) \cap \mathbb{N}$. For the base case we assume $|S| = 1$ and apply Theorem 4 with n set to the single element of S . We then define \tilde{S} to be a singleton set containing any one of the s_j rows of $\mathcal{M}_{s_1, K, n}$ which approximates x_n to the guaranteed precision. Now, suppose that the statement of Corollary 1 holds for all subsets $S \subseteq [0, N) \cap \mathbb{N}$ with $|S| \leq a \in \mathbb{N}^+$. Let $S' \subseteq [0, N) \cap \mathbb{N}$ have $|S'| \leq \frac{a \cdot c}{2}$. We will prove that the statement of Corollary 1 holds for S' .

For each $n \in S'$ and $j \in [1, K] \cap \mathbb{N}$ we will count a ‘failure’ if

$$\left| (\mathcal{M}_{s_1, K, n} \vec{x})_j - x_n \right| > \frac{\epsilon \cdot \left\| \vec{x} - \vec{x}_{(k/\epsilon)}^{\text{opt}} \right\|_1}{k}.$$

Theorem 4 tells us that there will be fewer than $(2/c) \cdot K$ ‘failures’ for each element of S' , for a total of fewer than $\frac{2 \cdot |S'|}{c} \cdot K$ collective ‘failures’ for all elements of S' . Clearly, at least one of the K s_j values used to construct $\mathcal{M}_{s_1, K}$ must ‘fail’ for fewer than $\frac{2 \cdot |S'|}{c}$ elements of S' . Let s'_j be the s_j value which ‘fails’ for the smallest number of elements of S' , and let $S'' \subset S'$ contain all the elements of S' for which s'_j ‘fails’. We can see that $|S''| < \frac{2 \cdot |S'|}{c} \leq a$. Our induction hypothesis applied to S'' together with the presence of s'_j yields the desired result. \square

Corollary 1 demonstrates the existence of a small number of s_j values which allow us to estimate every entry of a given vector. However, it is apparently difficult to locate these s_j values efficiently. The following corollary circumvents this difficulty by showing that a small set of randomly selected s_j values will still allow us to estimate all entries of any given vector with high probability. Thus, in practice it suffices to select a random subset of the rows from a $\mathcal{M}_{s_1, K}$ matrix.

Corollary 2. *Let $k, N, \epsilon^{-1} \in \mathbb{N}^+$, $\sigma \in [2/3, 1)$, $S \subseteq [0, N) \cap \mathbb{N}$, and $\vec{x} \in \mathbb{C}^N$. Set $K = c \cdot (k/\epsilon) \lceil \log_{s_1} N \rceil + 1$ for $s_1 \in \mathbb{N}$ and a constant integer $c \in [14, \infty)$. Form an $m \times N$ measurement matrix $\mathcal{M}_{s_1, K}$ as per Section 3. Finally, form a multiset of the s_j values for $\mathcal{M}_{s_1, K}$ by independently choosing*

$$(17) \quad l = \left\lceil 21 \cdot \ln \left(\frac{|S|}{1 - \sigma} \right) \right\rceil$$

s_j values uniformly at random with replacement. Denote this multiset of s_j values by

$$\tilde{S} = \left\{ s_{j_1}, s_{j_2}, \dots, s_{j_l} \right\}.$$

Then, with probability at least σ the resulting random matrix, $\mathcal{M}_{\tilde{S}}$, will have the following property: For all $n \in S$ more than $l/2$ of the $s_{j_n} \in \tilde{S}$ (counted with multiplicity) will have

$$\left| \left(\mathcal{M}_{\tilde{S}, n} \vec{x} - x_n \cdot \vec{\mathbb{1}}_l \right)_h \right| \leq \frac{\epsilon \cdot \left\| \vec{x} - \vec{x}_{(k/\epsilon)}^{\text{opt}} \right\|_1}{k}.$$

Proof: See Appendix B. \square

Notice that Corollary 2 considers selecting a multiset of rows from a $\mathcal{M}_{s_1, K}$ measurement matrix. In other words, some rows of the measurement matrix may be selected multiple times. If this occurs in practice, one should consider any multiply selected rows to be chosen more than once for counting purposes only. For example, during matrix multiplication a multiply selected row should be processed only once in order to avoid duplication of labor. However, the results of these calculations should be considered multiple times for the purposes of estimation (e.g., in the median operations of Algorithm 2).

We will now consider these $m \times N$ matrices, $\mathcal{M}_{s_1, K}$, with respect to the discrete Fourier transform. In particular, we will consider using $\mathcal{M}_{s_1, K}$ to estimate the Fourier transform of a periodic function along the lines of Theorem 4. As we shall see, the special number theoretic nature of our matrix constructions will allow us to estimate Fourier coefficients of any periodic function by using a small number of function samples.

3.2. The Fourier Case. Suppose $f : [0, 2\pi] \rightarrow \mathbb{C}$ is a complex valued function with $\hat{f} \in l^1$. Let P be the least common multiple of $\{N, s_1, \dots, s_K\}$ and form a set of samples from f , $\vec{A} \in \mathbb{C}^P$, with

$$A_p = f\left(p \cdot \frac{2\pi}{P}\right) \text{ for } p \in [0, P) \cap \mathbb{N}.$$

Ultimately, we want to use $\mathcal{M}_{s_1, K} \vec{f}$ in order to estimate the entries of the N -length vector \vec{f} . However, we must first calculate $\mathcal{M}_{s_1, K} \vec{f}$. In the remainder of this section we will discuss how to calculate $\mathcal{M}_{s_1, K} \vec{f} \in \mathbb{C}^m$ while using as few samples from f as possible in the process.

To solve this problem we will use an extended version of our $m \times N$ matrix $\mathcal{M}_{s_1, K}$. This extended matrix, $\mathcal{E}_{s_1, K}$, is the $m \times P$ matrix formed by extending each row $r_{j, h}$ of $\mathcal{M}_{s_1, K}$ as per Equation 4 for all $p \in [0, P)$. We now consider the product of $\mathcal{E}_{s_1, K}$ and the $P \times P$ discrete Fourier transform matrix, $\tilde{\Psi}$, defined by $\tilde{\Psi}_{\omega, p} = \frac{1}{P} \cdot e^{\frac{-2\pi i \cdot \omega \cdot p}{P}}$. For each row $r_{j, h}$ of $\mathcal{E}_{s_1, K}$ and column p of $\tilde{\Psi}$ we have

$$(18) \quad \left(\mathcal{E}_{s_1, K} \cdot \tilde{\Psi} \right)_{r_{j, h}, p} = \frac{1}{P} \sum_{l=0}^{s_j-1} e^{\frac{-2\pi i \cdot p \cdot (h+l \cdot s_j)}{P}} = \frac{e^{\frac{-2\pi i \cdot p \cdot h}{P}}}{P} \sum_{l=0}^{s_j-1} e^{\frac{-2\pi i \cdot p \cdot l}{P/s_j}} = \begin{cases} \frac{e^{\frac{-2\pi i \cdot p \cdot h}{P}}}{s_j} & \text{if } p \equiv 0 \pmod{s_j} \\ 0 & \text{otherwise} \end{cases}.$$

Thus, $\mathcal{E}_{s_1, K} \cdot \tilde{\Psi}$ is highly sparse. In fact, we can see that each $r_{j, h}$ row contains only s_j nonzero entries. Better still, all the rows associated with a given s_j have nonzero column entries in a pattern consistent with a small fast Fourier transform. This aliasing phenomena results in a fast algorithm for computing $\mathcal{E}_{s_1, K} \cdot \tilde{\Psi} \cdot \vec{A}$ (see Algorithm 1). Lemma 3 shows that $\mathcal{E}_{s_1, K} \tilde{\Psi} \vec{A}$ is a good approximation to $\mathcal{M}_{s_1, K} \vec{f} \in \mathbb{C}^m$ for all periodic functions whose Fourier transforms decay quickly enough.

Lemma 3. *Every entry of $\mathcal{E}_{s_1, K} \tilde{\Psi} \vec{A}$ approximates the associated entry of $\mathcal{M}_{s_1, K} \vec{f}$ to within $\left\| \hat{f} - \hat{f} \right\|_1$ precision.*

Proof:

Suppose that N is odd (the case for N even is analogous). Then, for all $j \in [1, K] \cap \mathbb{N}$ and $h \in [0, s_j) \cap \mathbb{N}$, we have that

$$\left| \left(\mathcal{M}_{s_1, K} \vec{f} - \mathcal{E}_{s_1, K} \tilde{\Psi} \vec{A} \right)_{r_{j, h}} \right| = \left| \sum_{l, |h+l \cdot s_j| \leq \frac{N-1}{2}} \hat{f}_{h+l \cdot s_j} - \sum_{\omega \equiv h \pmod{s_j}} \hat{f}(\omega) \right| = \left| \sum_{l, |h+l \cdot s_j| \leq \frac{N-1}{2}} \hat{f}(h+l \cdot s_j) - \sum_{\omega \equiv h \pmod{s_j}} \hat{f}(\omega) \right|.$$

Algorithm 1 FAST MULTIPLY

- 1: **Input:** Function f , integers $k < K < N$, relatively prime s_1, \dots, s_K
 - 2: **Output:** $\mathcal{E}_{s_1, K} \cdot \vec{\Psi} \cdot \vec{A}$
 - 3: **for** j from 1 to K **do**
 - 4: $\vec{A}_{s_j} \leftarrow f(0), f\left(\frac{2\pi}{s_j}\right), \dots, f\left(\frac{2\pi(s_j-1)}{s_j}\right)$
 - 5: $\widehat{\vec{A}}_{s_j} \leftarrow \text{FFT}[\vec{A}_{s_j}]$
 - 6: **end for**
 - 7: Output $(\widehat{\vec{A}}_{s_1}, \widehat{\vec{A}}_{s_2}, \dots, \widehat{\vec{A}}_{s_K})^T$
-

Cancelling all Fourier coefficients for frequencies in $(-\lfloor \frac{N}{2} \rfloor, \lfloor \frac{N}{2} \rfloor] \cap \mathbb{N}$ we get that

$$(19) \quad \left| \left(\mathcal{M}_{s_1, K} \vec{f} - \mathcal{E}_{s_1, K} \vec{\Psi} \vec{A} \right)_{r_{jh}} \right| = \left| \sum_{\omega \equiv h \pmod{s_j}, |\omega| \geq \frac{N+1}{2}} \hat{f}(\omega) \right| \leq \sum_{|\omega| \geq \frac{N+1}{2}} |\hat{f}(\omega)| = \|\hat{f} - \tilde{f}\|_1. \quad \square$$

By inspecting Equation 18 it is not difficult to see that Algorithm 1 utilizes exactly $m - (K - 1)$ samples from f . Considering this in combination with Theorem 5 in Section 3 leads us to the conclusion that Algorithm 1 samples f at $O\left(\frac{k^2 \cdot \lfloor \log_{(k/\epsilon)} N \rfloor^2 \cdot \ln\left(\frac{k \ln N}{\epsilon}\right)}{\epsilon^2}\right)$ distinct values. Similarly, we can see that Algorithm 1 runs in time $O\left(\sum_{j=1}^K s_j \log s_j\right)$ if we calculate the FFTs using a chirp z-transform [23]. Thus, for well chosen s_j values the runtime will be

$$(20) \quad \begin{aligned} O\left(\sum_{j=1}^K s_j \log s_j\right) &= O\left(\sum_{j=0}^{K-1} p_{q+j} \log p_{q+j}\right) = O(p_{q+K}^2) \quad (\text{see [15]}) \\ &= O\left(\frac{k^2 \cdot \lfloor \log_{(k/\epsilon)} N \rfloor^2 \cdot \ln^2\left(\frac{k \ln N}{\epsilon}\right)}{\epsilon^2}\right) \end{aligned}$$

using Equation 13. We will now demonstrate how the specialized $m \times N$ matrices, $\mathcal{M}_{s_1, K}$, along with their extended $m \times P$ counterpart matrices, $\mathcal{E}_{s_1, K}$, considered throughout Sections 3 and 3.2 can be utilized to construct accurate sparse Fourier transform methods.

4. FOURIER RECONSTRUCTION

In this section we develop a sparse Fourier transform based on the measurement matrices considered in the previous section. This sparse Fourier method is entirely dependent on the ability of our developed measurement matrices to accurately estimate any entry of a vector with which they have been multiplied (i.e., Theorem 4). The idea behind the algorithm is simple. We first quickly approximate the product of a Section 3 measurement matrix with the Fourier transform of an input function using Algorithm 1. We then use this product to accurately estimate all Fourier entries, keeping only the largest magnitude estimates for our final sparse Fourier approximation. See Algorithm 2 for pseudo code. Theorem 6 provides error, sampling, and runtime bounds for Algorithm 2.

Theorem 6. *Suppose $f : [0, 2\pi] \rightarrow \mathbb{C}$ has $\hat{f} \in l^1$. Let $N, k, \epsilon^{-1} \in \mathbb{N} - \{1\}$ with $N > (k/\epsilon) \geq 2$. Then, Algorithm 2 will output an $\vec{x}_S \in \mathbb{C}^N$ satisfying*

$$(21) \quad \left\| \vec{f} - \vec{x}_S \right\|_2 \leq \left\| \vec{f} - \vec{f}_k^{\text{opt}} \right\|_2 + \frac{22\epsilon \cdot \left\| \vec{f} - \vec{f}_{(k/\epsilon)}^{\text{opt}} \right\|_1}{\sqrt{k}} + 22\sqrt{k} \cdot \left\| \hat{f} - \tilde{f} \right\|_1.$$

In the process f will be evaluated at less than

$$26.02 \cdot \frac{k^2 \lfloor \log_{(k/\epsilon)} N \rfloor^2}{\epsilon^2} \cdot \ln\left(\frac{5.89 \cdot k \lfloor \log_{(k/\epsilon)} N \rfloor}{\epsilon}\right)$$

Algorithm 2 FOURIER APPROXIMATE 1

- 1: **Input:** $k, N, \epsilon^{-1} \in \mathbb{N} - \{1\}$, Function f , Measurement matrix $\mathcal{M}_{s_1, K}$ with $K = 4 \cdot (k/\epsilon) \lceil \log_{s_1} N \rceil + 1$ (see Section 3)
 - 2: **Output:** \vec{x}_S , an approximation to \vec{f}_k^{opt}
 - 3: Initialize $S \leftarrow \emptyset$, $\vec{x} \leftarrow \vec{0}_N$
 - 4: $\mathcal{E}_{s_1, K} \tilde{\Psi} \vec{A} \leftarrow$ Algorithm 1(f, k, K, N, s_j values for $\mathcal{M}_{s_1, K}$)
 - 5: **for** ω from $1 - \lfloor \frac{N}{2} \rfloor$ to $\lfloor \frac{N}{2} \rfloor$ **do**
 - 6: $\Re\{x_\omega\} \leftarrow$ median of multiset $\left\{ \Re\left\{ \left(\mathcal{E}_{s_1, K, \omega} \tilde{\Psi} \vec{A} \right)_j \right\} \mid 1 \leq j \leq K \right\}$
 - 7: $\Im\{x_\omega\} \leftarrow$ median of multiset $\left\{ \Im\left\{ \left(\mathcal{E}_{s_1, K, \omega} \tilde{\Psi} \vec{A} \right)_j \right\} \mid 1 \leq j \leq K \right\}$
 - 8: **end for**
 - 9: Sort \vec{x} entries by magnitude so that $|x_{\omega_1}| \geq |x_{\omega_2}| \geq |x_{\omega_3}| \geq \dots$
 - 10: $S \leftarrow \{\omega_1, \omega_2, \dots, \omega_{2k}\}$
 - 11: Output \vec{x}_S
-

points in $[0, 2\pi]$. The runtime of lines 5 through 11 is $O(N \cdot (k/\epsilon) \log_{(k/\epsilon)} N)$.

Proof:

Fix $\omega \in \left(-\lfloor \frac{N}{2} \rfloor, \lfloor \frac{N}{2} \rfloor\right] \cap \mathbb{Z}$ and let δ be set to

$$\delta = \frac{\epsilon \cdot \left\| \vec{f} - \vec{f}_{(k/\epsilon)}^{\text{opt}} \right\|_1}{k} + \left\| \hat{f} - \tilde{f} \right\|_1.$$

As a consequence of Theorem 4 and Lemma 3 we can see that more than half of the $K = 4 \cdot (k/\epsilon) \lceil \log_{s_1} N \rceil + 1$ entries of $\mathcal{E}_{s_1, K, \omega} \tilde{\Psi} \vec{A}$ produced in line 4 will satisfy $\left| \left(\mathcal{E}_{s_1, K, \omega} \tilde{\Psi} \vec{A} \right)_j - \hat{f}_\omega \right| \leq \delta$. Therefore, the x_ω value produced by lines 6 and 7 will have

$$(22) \quad |x_\omega - \hat{f}_\omega| \leq \sqrt{2} \cdot \delta.$$

Since Equation 22 holds for all $\omega \in \left(-\lfloor \frac{N}{2} \rfloor, \lfloor \frac{N}{2} \rfloor\right] \cap \mathbb{Z}$ we can begin to bound the approximation error by

$$(23) \quad \begin{aligned} \left\| \vec{f} - \vec{x}_S \right\|_2 &\leq \left\| \vec{f} - \vec{f}_S \right\|_2 + \left\| \vec{f}_S - \vec{x}_S \right\|_2 \leq \left\| \vec{f} - \vec{f}_S \right\|_2 + 2\sqrt{k} \cdot \delta \\ &= \sqrt{\left\| \vec{f} - \vec{f}_k^{\text{opt}} \right\|_2^2 + \sum_{\omega \in S_k^{\text{opt}} - S} |\hat{f}_\omega|^2 - \sum_{\tilde{\omega} \in S - S_k^{\text{opt}}} |\hat{f}_{\tilde{\omega}}|^2} + 2\sqrt{k} \cdot \delta. \end{aligned}$$

In order to make additional progress on Equation 23 we must first consider the possible magnitudes of \vec{f} entries at indices in $S - S_k^{\text{opt}}$ and $S_k^{\text{opt}} - S$.

Suppose $\omega \in S_k^{\text{opt}} - S \neq \emptyset$ and let $\tilde{\omega} \in S - S_k^{\text{opt}}$. Line 10 will only have placed $\tilde{\omega} \in S$ instead of ω if $|x_{\tilde{\omega}}| \geq |x_\omega|$. However, this can only happen if

$$|\hat{f}_{\omega_k}| + \sqrt{2} \cdot \delta \geq |\hat{f}_{\tilde{\omega}}| + \sqrt{2} \cdot \delta \geq |\hat{f}_\omega| - \sqrt{2} \cdot \delta \geq |\hat{f}_{\omega_k}| - \sqrt{2} \cdot \delta.$$

In other words, all elements of $S - S_k^{\text{opt}}$ and $S_k^{\text{opt}} - S$ must index \vec{f} entries with roughly the same magnitude as the k^{th} largest magnitude entry of \vec{f} (up to a δ factor). Furthermore, since $|S| = 2k$ we can see that $|S - S_k^{\text{opt}}| \geq 2 \cdot |S_k^{\text{opt}} - S|$. We are now ready to give Equation 23 further consideration.

If $S_k^{\text{opt}} - S = \emptyset$ we are finished. Otherwise, if $S_k^{\text{opt}} - S \neq \emptyset$, we will have

$$\sum_{\tilde{\omega} \in S - S_k^{\text{opt}}} |\hat{f}_{\tilde{\omega}}|^2 \geq 2 \cdot |S_k^{\text{opt}} - S| \cdot \left(|\hat{f}_{\omega_k}| - 2\sqrt{2} \cdot \delta \right)^2 = \mathbf{A},$$

and

$$\mathbf{B} = |S_k^{\text{opt}} - S| \cdot (|\hat{f}_{\omega_k}| + 2\sqrt{2} \cdot \delta)^2 \geq \sum_{\omega \in S_k^{\text{opt}} - S} |\hat{f}_{\omega}|^2.$$

If $\mathbf{A} \geq \mathbf{B}$ then we are again finished. If $\mathbf{A} < \mathbf{B}$ then

$$|\hat{f}_{\omega_k}|^2 - 12\sqrt{2}\delta \cdot |\hat{f}_{\omega_k}| + 8\delta^2 < 0$$

which can only happen if $|\hat{f}_{\omega_k}| \in ((6\sqrt{2} - 8) \cdot \delta, (6\sqrt{2} + 8) \cdot \delta)$. Therefore, in the worse case we can continue to bound Equation 23 by

$$\left\| \vec{f} - \vec{x}_S \right\|_2 \leq \sqrt{\left\| \vec{f} - \vec{f}_k^{\text{opt}} \right\|_2^2 + k \cdot (8\sqrt{2} + 8)^2 \cdot \delta^2} + 2\sqrt{k} \cdot \delta \leq \left\| \vec{f} - \vec{f}_k^{\text{opt}} \right\|_2 + 22\sqrt{k} \cdot \delta.$$

The error bound in Equation 21 follows.

The upper bound on the number of point evaluations of f follows directly from the application of Theorem 5 with $c = 4$. Finding the largest $2k$ magnitude entries of \vec{x} in lines 9 and 10 can be accomplished in $O(N \cdot \log k)$ time by using a binary search tree (see [16]). Therefore, the runtime of Algorithm 2 will be dominated by the median operations in lines 6 and 7. Each of these medians can be accomplished in $O(K)$ time using a median-of-medians algorithm (e.g., [9]). The stated $O(N \cdot K)$ runtime follows. \square

Note that the overall runtime behavior of Algorithm 2 will be dictated by both Equation 20 and the runtime stated in Theorem 6. However, for most reasonable values of sublinear sparsity (i.e., whenever k/ϵ is $O(N/\log^3 N)$) the total runtime of Algorithm 2 will be $O(N \cdot (k/\epsilon) \log_{(k/\epsilon)} N)$. One strategy for decreasing the runtime of Algorithm 2 is to decrease the number of measurement matrix rows, K , required to accurately estimate each Fourier coefficient. Pursuing this strategy also has the additional benefit of reducing the number of function evaluations required for approximate Fourier reconstruction. However, in exchange for these improvements we will have to sacrifice approximation guarantees for a small probability of outputting a relatively inaccurate answer.

Following the strategy above we will improve the performance of Algorithm 2 by modifying its input measurement matrix. Instead of inputting a $\mathcal{M}_{s_1, K}$ measurement matrix as constructed in Section 3 we will utilize a randomly constructed $\mathcal{M}_{\tilde{s}}$ measurement matrix as described in Section 3.1. Corollary 2 ensures that such a randomly constructed $\mathcal{M}_{\tilde{s}}$ matrix will be likely to have all the properties of $\mathcal{M}_{s_1, K}$ matrices that Algorithm 2 needs. Hence, with high probability we will achieve output from Algorithm 2 with the same approximation error bounds as derived for Theorem 6. Formalizing these ideas we obtain the following Corollary proved in Appendix C.

Corollary 3. *Suppose $f : [0, 2\pi] \rightarrow \mathbb{C}$ has $\hat{f} \in \ell^1$. Let $\sigma \in [2/3, 1)$ and $N, k, \epsilon^{-1} \in \mathbb{N} - \{1\}$ with $N > (k/\epsilon) \geq 2$. Algorithm 2 may be executed using a matrix $\mathcal{M}_{\tilde{s}}$ from Section 3.1 in place of the matrix $\mathcal{M}_{s_1, K}$ from Section 3 to produce an output vector $\vec{x}_S \in \mathbb{C}^N$ which will satisfy Equation 21 with probability at least σ . In the process f will be evaluated at less than*

$$15.89 \cdot \left[21 \cdot \ln \left(\frac{N}{1-\sigma} \right) \right] \cdot \frac{k \lfloor \log_{(k/\epsilon)} N \rfloor}{\epsilon} \cdot \left(\ln \left(\frac{15.89 \cdot k \lfloor \log_{(k/\epsilon)} N \rfloor}{\epsilon} \right) + \ln \ln \left(\frac{15.89 \cdot k \lfloor \log_{(k/\epsilon)} N \rfloor}{\epsilon} \right) \right)$$

points in $[0, 2\pi]$. The runtime of lines 5 through 11 will be $O\left(N \cdot \log\left(\frac{N}{1-\sigma}\right)\right)$.

Proof: See Appendix C. \square

When executed with a random matrix $\mathcal{M}_{\tilde{s}}$ as input the overall runtime complexity of Algorithm 2 will be determined by both the runtime stated in Corollary 3 and the runtime of Algorithm 1. Suppose \tilde{S} is a subset of $O\left(\log\left(\frac{N}{1-\sigma}\right)\right)$ s_j values defined as per Equations 9 – 11. Then, Algorithm 1 will have a runtime complexity of

$$\begin{aligned} O\left(\sum_{s_j \in \tilde{S}} s_j \cdot \log s_j\right) &= O\left(p_{q+K} \cdot \log p_{q+K} \cdot \log\left(\frac{N}{1-\sigma}\right)\right) \quad (\text{see Equation 13}) \\ (24) \quad &= O\left(\frac{k \cdot \log_{(k/\epsilon)} N}{\epsilon} \cdot \log^2\left(\frac{k \cdot \log N}{\epsilon}\right) \cdot \log\left(\frac{N}{1-\sigma}\right)\right). \end{aligned}$$

Thus, Algorithm 2 executed with a random input matrix from Section 3.1 will have a total runtime complexity of $O\left(N \cdot \log\left(\frac{N}{1-\sigma}\right)\right)$ whenever (k/ϵ) is $O(N/\log^3 N)$. If we now set the desired success probability, σ , to be $1 - 1/N^{O(1)}$ we obtain an overall $O(N \cdot \log N)$ computational complexity for Algorithm 2. This matches the runtime behavior of a standard fast Fourier transform while requiring asymptotically fewer function evaluations.

In the next section we will discuss methods for further decreasing the runtime requirements of Algorithm 2 while maintaining its approximation guarantees (i.e., the error bound in Equation 21). As a result we will develop sublinear-time Fourier algorithms that have both universal recovery guarantees and uniformly bounded runtime requirements.

5. DECREASING THE RUNTIME COMPLEXITY

Let \mathcal{A}, \mathcal{B} be $m \times N$ and $\tilde{m} \times N$ complex valued matrixes, respectively. Then, their row tensor product, $\mathcal{A} \otimes \mathcal{B}$, is defined to be the $(m \cdot \tilde{m}) \times N$ complex valued matrix created by performing component-wise multiplication of all rows of \mathcal{A} with all rows of \mathcal{B} . More specifically,

$$(\mathcal{A} \otimes \mathcal{B})_{i,j} = \mathcal{A}_{i \bmod m, j} \cdot \mathcal{B}_{\frac{i-i \bmod m}{m}, j}.$$

In this section we will use the row tensor product of two types of specially constructed measurement matrices in order to improve the runtime complexity of Algorithm 2. One of these matrix types will be the $\mathcal{M}_{s_1, K}$ measurement matrices developed in Section 3. The other type of matrix is described in the next two paragraphs.

Suppose that an $m \times N$ measurement matrix, $\mathcal{M}_{s_1, K}$, is given. Furthermore, suppose that $s_1, \dots, s_K \in \mathbb{N}$ are such that there exist λ integers, $t_1 < \dots < t_\lambda < s_1$, with

$$\prod_{i=1}^{\lambda} t_i \geq \frac{N}{s_1}$$

that also have the property that the set

$$\{t_1, \dots, t_\lambda, s_1, \dots, s_K\}$$

is pairwise relatively prime. Note that such t_i values can indeed be found if all the given s_j values are prime numbers and $s_1 \geq \log_2 N \cdot (\ln \log_2 N + \ln \ln \log_2 N) \geq p_{\lfloor \log_2 N \rfloor}$ for $N \geq 64$ (see [10]). We will now demonstrate how to use such t_i values to create an $\tilde{m} \times N$ matrix, $\mathcal{N}_{\lambda, s_1}$, along the lines of Section 3.

Create a row, $\tilde{r}_{i,h}$, in $\mathcal{N}_{\lambda, s_1}$ for each possible residue of each t_i integer (i.e., $\tilde{r}_{i,h}$ has $i \in [1, \lambda] \cap \mathbb{N}$ and $h \in [0, t_i] \cap \mathbb{N}$). The n^{th} entry of each $\tilde{r}_{i,h}$ row, $n \in [0, N) \cap \mathbb{N}$, will be

$$(25) \quad \begin{aligned} (\tilde{r}_{i,h})_n &= \delta((n-h) \bmod t_i) \\ &= \begin{cases} 1 & \text{if } n \equiv h \bmod t_i \\ 0 & \text{otherwise} \end{cases}. \end{aligned}$$

We then define

$$(26) \quad \mathcal{N}_{\lambda, s_1} = \begin{pmatrix} \vec{\mathbb{1}}_N \\ \tilde{r}_{1,0} \\ \vdots \\ \tilde{r}_{1, t_1-1} \\ \vdots \\ \tilde{r}_{\lambda, t_\lambda-1} \end{pmatrix}.$$

The result is an $(\tilde{m} = 1 + \sum_{i=1}^{\lambda} t_i) \times N$ matrix with binary entries. The following Lemma, proven in Appendix D, upper bounds the smallest possible number of rows in any such $\mathcal{N}_{\lambda, s_1}$ matrix.

Lemma 4. *Suppose that $N, s_1, \dots, s_K \in \mathbb{N}$ with*

$$\frac{N}{3} \geq s_1 > \left\lceil 3 \cdot \frac{\ln(N/s_1)}{\ln \ln(N/s_1)} \right\rceil \cdot \left(\ln \left\lceil 3 \cdot \frac{\ln(N/s_1)}{\ln \ln(N/s_1)} \right\rceil + \ln \ln \left\lceil 3 \cdot \frac{\ln(N/s_1)}{\ln \ln(N/s_1)} \right\rceil \right),$$

and s_1, \dots, s_K containing no prime factors less than s_1 . Then, there exists a valid $\tilde{m} \times N$ measurement matrix, $\mathcal{N}_{\lambda, s_1}$, with a number of rows

$$\tilde{m} < \frac{3}{4} \left(\left\lceil 3 \cdot \frac{\ln(N/s_1)}{\ln \ln(N/s_1)} \right\rceil + 1 \right)^2 \cdot \ln \left(\left\lceil 3 \cdot \frac{\ln(N/s_1)}{\ln \ln(N/s_1)} \right\rceil + 1 \right) + 1.$$

The corresponding value of λ is $\lceil 3 \cdot \ln(N/s_1) / \ln \ln(N/s_1) \rceil$.

Proof: See Appendix D. \square

The $(m \cdot \tilde{m}) \times N$ row tensor product matrix, $\mathcal{R}_{\lambda, K} = \mathcal{M}_{s_1, K} \otimes \mathcal{N}_{\lambda, s_1}$, has several useful properties. First, the fact that the first row of $\mathcal{N}_{\lambda, s_1}$ is the all-ones vector means that $\mathcal{R}_{\lambda, K}$ will contain a copy of every row of $\mathcal{M}_{s_1, K}$. Second, all $\mathcal{R}_{\lambda, K}$ rows that are not copies of $\mathcal{M}_{s_1, K}$ rows will have the form $\tilde{r}_{i, j, h} = r_{j, h \bmod s_j} \otimes \tilde{r}_{i, h \bmod t_i}$ for some $i \in [1, \lambda] \cap \mathbb{N}$, $j \in [1, K] \cap \mathbb{N}$, and $h \in [0, t_i \cdot s_j) \cap \mathbb{N}$. That is, the Chinese Remainder Theorem tells us that each such $\mathcal{R}_{\lambda, K}$ row will have its n^{th} entry given by

$$(27) \quad (\tilde{r}_{i, j, h})_n = \delta((n - h) \bmod t_i \cdot s_j) = \begin{cases} 1 & \text{if } n \equiv h \bmod t_i \cdot s_j \\ 0 & \text{otherwise} \end{cases}.$$

The end result is that $\mathcal{R}_{\lambda, K}$ maintains a rigid number theoretic structure. The following Lemma summarizes the most important properties of $\mathcal{R}_{\lambda, K} = \mathcal{M}_{s_1, K} \otimes \mathcal{N}_{\lambda, s_1}$.

Lemma 5. Let $k, \epsilon^{-1}, s_1, \lambda, n \in [2, N) \cap \mathbb{N}$, $\vec{x} \in \mathbb{C}^N$, and $K = 4 \cdot (k/\epsilon) \lceil \log_{s_1} N \rceil + 1$. Then, more than $\frac{K}{2}$ of the K entries of $\mathcal{M}_{s_1, K, n} \cdot \vec{x}$ will estimate x_n to within $\bar{\delta} = \frac{\epsilon \cdot \|\vec{x} - \vec{x}^{\text{opt}}\|_1}{k}$ precision. Furthermore, if $r_{j', n \bmod s_{j'}} \in \{0, 1\}^N$ is a row of $\mathcal{M}_{s_1, K, n}$ associated with one of these more than $\frac{K}{2}$ entries then it will have all of the following properties:

- (1) $|r_{j', n \bmod s_{j'}} \cdot \vec{x} - x_n| \leq \bar{\delta}$,
- (2) $\left| \left(r_{j', n \bmod s_{j'}} \otimes \tilde{r}_{i, n \bmod t_i} \right) \cdot \vec{x} - x_n \right| = \left| \tilde{r}_{i, j', n \bmod t_i \cdot s_{j'}} \cdot \vec{x} - x_n \right| \leq \bar{\delta}$ for all $i \in [1, \lambda] \cap \mathbb{N}$, and
- (3) $\left| \left(r_{j', n \bmod s_{j'}} \otimes \tilde{r}_{i, h} \right) \cdot \vec{x} \right| = \left| \tilde{r}_{i, j', h \neq n \bmod t_i \cdot s_{j'}} \cdot \vec{x} \right| \leq \bar{\delta}$ for all $i \in [1, \lambda] \cap \mathbb{N}$ and $h \in [0, t_i) \cap (\mathbb{N} - \{n \bmod t_i\})$.

Proof: See Appendix E. \square

Suppose $f : [0, 2\pi] \rightarrow \mathbb{C}$ is a complex valued function with $\hat{f} \in l^1$. It is not difficult to see that $\mathcal{R}_{\lambda, K} \vec{f}$ can be approximated using Algorithm 1 from Section 3.2 since $\mathcal{R}_{\lambda, K}$ maintains the required number theoretic structure. We will simply perform FFTs on arrays of function samples with sizes given by all possible $t_i \cdot s_j$ value products. The total number of function samples taken will be at most $m \cdot \tilde{m} - (\lambda \cdot K + K - 1)$. For s_j and t_i values chosen as per Theorem 5 and Lemma 4, respectively, the runtime required by Algorithm 1 to approximate $\mathcal{R}_{\lambda, K} \vec{f}$ will be

$$(28) \quad \begin{aligned} O \left(\sum_{i=1}^{\lambda} \sum_{j=1}^K t_i \cdot s_j \log s_j \right) &= O \left(\sum_{i=1}^{\lambda} \sum_{j=0}^{K-1} p_i \cdot p_{q+j} \log p_{q+j} \right) = O \left(\frac{p_{q+K}^2 \cdot p_{\lambda}^2}{\ln p_{\lambda}} \right) \quad (\text{see [15]}) \\ &= O \left(k^2 \cdot \frac{\ln^2 N \cdot \ln^2 \left(\frac{k \ln N}{\epsilon} \right) \cdot \ln^2 \left(\frac{\epsilon N}{k} \right)}{\epsilon^2 \cdot \ln^2 \left(\frac{k}{\epsilon} \right) \cdot \ln \ln \left(\frac{\epsilon N}{k} \right)} \right). \end{aligned}$$

The last equality follows from Equation 13 and the Prime Number Theorem. Finally, it is not difficult to see that the precision guarantees of Lemma 3 will still hold for an Algorithm 1 approximation to $\mathcal{R}_{\lambda, K} \vec{f}$.

Perhaps most importantly, the number theoretic structure of $\mathcal{R}_{\lambda, K}$ also allows us to use methods analogous to those outlined in Sections 1.1 and 5 of [14] to quickly identify frequencies with large magnitude Fourier coefficients in \hat{f} . Suppose that $|\hat{f}_{\omega}|$ is large relative to $\|\hat{f}\|_1$ (e.g., more than one tenth as large). In this case Lemma 5 above tells us that \hat{f}_{ω} will also have a magnitude nearly as large as that of most entries of $\mathcal{M}_{s_1, K, \omega} \vec{f}$. Let $r_{j, \omega \bmod s_j}$ be the row of $\mathcal{M}_{s_1, K, \omega}$ associated with one of these $\mathcal{M}_{s_1, K, \omega} \vec{f}$ entries dominated by \hat{f}_{ω} . By its construction we know that $\mathcal{R}_{\lambda, K}$ will not only contain $r_{j, \omega \bmod s_j}$, but also the related rows $\tilde{r}_{1, j, \omega \bmod t_1 \cdot s_{j'}} \dots \tilde{r}_{\lambda, j, \omega \bmod t_{\lambda} \cdot s_j}$. Furthermore, all $\lambda + 1$ entries of $\mathcal{R}_{\lambda, K, \omega} \vec{f}$

Algorithm 3 FOURIER APPROXIMATE 2

- 1: **Input:** $k, N, \epsilon^{-1} \in \mathbb{N} - \{1\}$, Function f , An $(m \cdot \tilde{m}) \times N$ measurement matrix $\mathcal{R}_{\lambda, K}$ with $K = 4 \cdot (k/\epsilon) \lceil \log_{s_1} N \rceil + 1$
- 2: **Output:** \vec{x}_S , an approximation to \vec{f}_k^{opt}
- 3: Initialize $S \leftarrow \emptyset$, $\vec{x} \leftarrow \vec{0}_N$
- 4: $\mathcal{G}_{\lambda, K} \tilde{\Psi} \tilde{A} \leftarrow$ Algorithm 1(f, k, K, N, s_j and t_i values for $\mathcal{R}_{\lambda, K}$)
- 5: $\mathcal{E}_{s_1, K} \tilde{\Psi} \tilde{A} \leftarrow$ The m entries of $\mathcal{G}_{\lambda, K} \tilde{\Psi} \tilde{A}$ that approximate $\mathcal{M}_{s_1, K} \vec{f}$

IDENTIFICATION OF FREQUENCIES WITH LARGE FOURIER COEFFICIENTS

- 6: **for** j from 1 to K **do**
- 7: **for** h from 0 to $s_j - 1$ **do**
- 8: **for** i from 1 to λ **do**
- 9: $b_{\min} \leftarrow \arg \min_{b \in [0, t_i)} \left| \left(\mathcal{E}_{s_1, K} \tilde{\Psi} \tilde{A} \right)_{r_{jh}} - \left(\mathcal{G}_{\lambda, K} \tilde{\Psi} \tilde{A} \right)_{\tilde{r}_{i, jh + b s_j}} \right|$
- 10: $a_{jh, i} \leftarrow (h + b_{\min} \cdot s_j) \bmod t_i$
- 11: **end for**
- 12: Reconstruct ω_{jh} using that $\omega_{jh} \equiv h \bmod s_j$, $\omega_{jh} \equiv a_{j, h, 1} \bmod t_1, \dots, \omega_{jh} \equiv a_{j, h, \lambda} \bmod t_\lambda$
- 13: **end for**
- 14: **end for**

FOURIER COEFFICIENT ESTIMATION

- 15: **for each** ω_{jh} value reconstructed $> \frac{K}{2}$ times **do**
 - 16: $\Re \{x_{\omega_{jh}}\} \leftarrow$ median of multiset $\left\{ \Re \left\{ \left(\mathcal{G}_{\lambda, K, \omega_{jh}} \tilde{\Psi} \tilde{A} \right)_j \right\} \mid 1 \leq j \leq K \cdot (\lambda + 1) \right\}$
 - 17: $\Im \{x_{\omega_{jh}}\} \leftarrow$ median of multiset $\left\{ \Im \left\{ \left(\mathcal{G}_{\lambda, K, \omega_{jh}} \tilde{\Psi} \tilde{A} \right)_j \right\} \mid 1 \leq j \leq K \cdot (\lambda + 1) \right\}$
 - 18: **end for**
 - 19: Sort nonzero \vec{x} entries by magnitude so that $|x_{\omega_1}| \geq |x_{\omega_2}| \geq |x_{\omega_3}| \geq \dots$
 - 20: $S \leftarrow \{\omega_1, \omega_2, \dots, \omega_{2k}\}$
 - 21: Output \vec{x}_S
-

associated with these rows will also be dominated by \hat{f}_ω (see Lemma 5). On the other hand, for each $i \in [1, \lambda] \cap \mathbb{N}$ the $\left(\mathcal{R}_{\lambda, K, \omega} \vec{f} \right)_{\tilde{r}_{i, jh + b s_j}}$ entries will all be significantly smaller than \hat{f}_ω in magnitude. Hence, by comparing the relative magnitudes of the entries in $\left(r_{j, \omega \bmod s_j} \otimes \mathcal{N}_{\lambda, s_1} \right) \vec{f}$ we can discern $\omega \bmod s_j, \omega \bmod t_1 \cdot s_j, \dots, \omega \bmod t_\lambda \cdot s_j$. The end result is that ω can be recovered by inspecting $\mathcal{R}_{\lambda, K, \omega} \vec{f}$. See [14] for a detailed discussion of a similar recovery procedure. Utilizing these ideas we obtain Algorithm 3.

Note that Algorithms 2 and 3 are quite similar. The only significant difference between them is that Algorithm 2 estimates Fourier coefficients for all frequencies in the bandwidth specified by N whereas Algorithm 3 restricts itself to estimating the Fourier coefficients for only a small number of frequencies it identifies as significant. Given these similarities it should not be surprising that demonstrating the correctness of Algorithm 3 depends primarily on showing that it can correctly identify all frequencies with coefficients that are sufficiently large in magnitude. This is established in Lemma 6 below.

Lemma 6. *Suppose that $\omega \in \left(-\left\lceil \frac{N}{2} \right\rceil, \left\lfloor \frac{N}{2} \right\rfloor \right) \cap \mathbb{Z}$ is such that*

$$|\hat{f}_\omega| > 4 \cdot \left(\frac{\epsilon \cdot \left\| \vec{f} - \vec{f}_{(k/\epsilon)}^{\text{opt}} \right\|_1}{k} + \left\| \hat{f} - \tilde{f} \right\|_1 \right).$$

Then, lines 6 through 14 of Algorithm 3 will reconstruct ω more than $\frac{K}{2}$ times.

Proof:

Suppose that $\omega \in \left(-\left\lceil \frac{N}{2} \right\rceil, \left\lfloor \frac{N}{2} \right\rfloor\right) \cap \mathbb{Z}$ has $|\hat{f}_\omega| > 4\delta$ where

$$\delta = \frac{\epsilon \cdot \left\| \vec{f} - \vec{f}_{(k/\epsilon)}^{\text{opt}} \right\|_1}{k} + \left\| \hat{f} - \bar{f} \right\|_1.$$

Lemma 5 and Lemma 3 guarantee that $\left| \left(\mathcal{E}_{s_1, K, \omega} \vec{\Psi} \vec{A} \right)_j - \hat{f}_\omega \right| \leq \delta$ for more than $\frac{K}{2}$ entry indexes j . Furthermore, if $j' \in [1, K] \cap \mathbb{N}$ is one of these more than $\frac{K}{2}$ indexes, then property (2) of Lemma 5 together with the preceding discussion of Lemma 3 also ensures that $\left| \left(\mathcal{G}_{\lambda, K} \vec{\Psi} \vec{A} \right)_{\bar{r}_{i, j', \omega \bmod t_i s_j}} - \hat{f}_\omega \right| \leq \delta$ for all $i \in [1, \lambda] \cap \mathbb{N}$. Fix $i \in [1, \lambda] \cap \mathbb{N}$. Thus, if $b \in [0, t_i] \cap \mathbb{N}$ in line 9 of Algorithm 3 satisfies

$$(29) \quad \omega \equiv \left((\omega \bmod s_j) + b \cdot s_j \right) \bmod t_i \cdot s_j$$

we can see that

$$\left| \left(\mathcal{E}_{s_1, K} \vec{\Psi} \vec{A} \right)_{r_{j', \omega \bmod s_j}} - \left(\mathcal{G}_{\lambda, K} \vec{\Psi} \vec{A} \right)_{\bar{r}_{i, j', (\omega \bmod s_j) + b s_j}} \right| = \left| \left(\mathcal{E}_{s_1, K} \vec{\Psi} \vec{A} \right)_{r_{j', \omega \bmod s_j}} - \hat{f}_\omega + \hat{f}_\omega - \left(\mathcal{G}_{\lambda, K} \vec{\Psi} \vec{A} \right)_{\bar{r}_{i, j', \omega \bmod t_i s_j}} \right| \leq 2\delta.$$

Otherwise, if $b \in [0, t_i] \cap \mathbb{N}$ does not satisfy Equation 29, property (3) from Lemma 5 in combination with Lemma 3 ensures that

$$\begin{aligned} 2\delta &< \left| \hat{f}_\omega \right| - \left| \left(\mathcal{E}_{s_1, K} \vec{\Psi} \vec{A} \right)_{r_{j', \omega \bmod s_j}} - \hat{f}_\omega \right| - \left| \left(\mathcal{G}_{\lambda, K} \vec{\Psi} \vec{A} \right)_{\bar{r}_{i, j', (\omega \bmod s_j) + b s_j}} \right| \\ &\leq \left| \left(\mathcal{E}_{s_1, K} \vec{\Psi} \vec{A} \right)_{r_{j', \omega \bmod s_j}} - \hat{f}_\omega + \hat{f}_\omega - \left(\mathcal{G}_{\lambda, K} \vec{\Psi} \vec{A} \right)_{\bar{r}_{i, j', (\omega \bmod s_j) + b s_j}} \right| = \left| \left(\mathcal{E}_{s_1, K} \vec{\Psi} \vec{A} \right)_{r_{j', \omega \bmod s_j}} - \left(\mathcal{G}_{\lambda, K} \vec{\Psi} \vec{A} \right)_{\bar{r}_{i, j', (\omega \bmod s_j) + b s_j}} \right|. \end{aligned}$$

Therefore, the $b = b_{\min}$ identified in line 9 of Algorithm 3 will be guaranteed to satisfy Equation 29 for all $i \in [1, \lambda] \cap \mathbb{N}$.

Once we have identified $\omega \bmod t_i \cdot s_j$ in this fashion we can find $\omega \bmod t_i$ in line 10 of Algorithm 3 by computing $(\omega \bmod t_i \cdot s_j) \bmod t_i$. Finally, by construction, the set $\{t_1, \dots, t_\lambda, s_j\}$ both has a collective product larger than N , and is pairwise relatively prime. Therefore, the Chinese Remainder Theorem guarantees that line 12 of Algorithm 3 will indeed correctly reconstruct ω when $j = j'$ and $h = \omega \bmod s_j$. \square

With Lemma 6 in hand we are now prepared to prove that Algorithm 3 can indeed recover near-optimal sparse Fourier representations in sublinear-time. We begin by using Lemma 6 to show that all sufficiently energetic frequencies are guaranteed to be identified. Hence, the only way Algorithm 3 will not include an optimal Fourier representation frequency in its output is if the frequency is either (i) insufficiently energetic to be identified, or (ii) gets identified, but is then mistakenly estimated to have a smaller magnitude Fourier coefficient than many other somewhat energetic frequencies. In the case of (i) it is forgivable to exclude the frequency given that it must have a Fourier coefficient with a relatively small magnitude. In the case of (ii) we make up for the exclusion of a truly energetic frequency term by including many other less significant, but still fairly energetic, frequency terms in its place. Carefully combining these ideas leads us to the error, sampling, and runtime bounds for Algorithm 3 stated in Theorem 7 below.

Theorem 7. *Suppose $f : [0, 2\pi] \rightarrow \mathbb{C}$ has $\hat{f} \in l^1$. Let $N, k, \epsilon^{-1} \in \mathbb{N} - \{1\}$ with $N > (k/\epsilon) \geq 2$. Then, Algorithm 3 will output an $\vec{x}_S \in \mathbb{C}^N$ satisfying*

$$(30) \quad \left\| \vec{f} - \vec{x}_S \right\|_2 \leq \left\| \vec{f} - \vec{f}_k^{\text{opt}} \right\|_2 + \frac{22\epsilon \cdot \left\| \vec{f} - \vec{f}_{(k/\epsilon)}^{\text{opt}} \right\|_1}{\sqrt{k}} + 22\sqrt{k} \cdot \left\| \hat{f} - \bar{f} \right\|_1.$$

Under the conditions of Lemma 4, f will be evaluated at less than

$$19.52 \cdot \frac{k^2 \left\lceil \log_{(k/\epsilon)} N \right\rceil^2}{\epsilon^2} \cdot \ln \left(\frac{5.89 \cdot k \left\lceil \log_{(k/\epsilon)} N \right\rceil}{\epsilon} \right) \cdot \left\{ \left(\left\lceil 3 \cdot \frac{\ln(\epsilon N/k)}{\ln \ln(\epsilon N/k)} \right\rceil + 1 \right)^2 \cdot \ln \left(\left\lceil 3 \cdot \frac{\ln(\epsilon N/k)}{\ln \ln(\epsilon N/k)} \right\rceil + 1 \right) + \frac{4}{3} \right\}$$

points in $[0, 2\pi]$. The runtime of lines 6 through 21, as well as the number of f -evaluations, is $O\left(\frac{k^2 \cdot \log^2 N \cdot \log\left(\frac{k \cdot \ln N}{\epsilon}\right) \cdot \log^2\left(\frac{\epsilon N}{k}\right)}{\log^2\left(\frac{k}{\epsilon}\right) \cdot \epsilon^2 \cdot \log\log\left(\frac{\epsilon N}{k}\right)}\right)$.

Proof: See Appendix F. \square

The overall runtime behavior of Algorithm 3 is determined by both the runtime of Algorithm 1 as called in line 4 of Algorithm 3, and the runtime stated in Theorem 7. The overall runtime complexity of Algorithm 3 is therefore given in Equation 28. As in Section 4 above, both this runtime and the number of function evaluations required for approximate Fourier reconstruction can be decreased by reducing the number of measurement matrix rows (i.e., $\mathcal{R}_{\lambda, K}$ rows) used to estimate each Fourier coefficient. This effectively replaces K in Algorithm 3 with a significantly smaller value (e.g., the value l from Corollary 2). However, in exchange for the resulting runtime improvements we will once again have to sacrifice approximation guarantees for a small probability of outputting a highly inaccurate answer.

Following the strategy above, we will improve the performance of Algorithm 3 by modifying its utilized measurement matrix as follows: Instead of using a $\mathcal{M}_{s_1, K}$ matrix as constructed in Section 3 to build $\mathcal{R}_{\lambda, K} = \mathcal{M}_{s_1, K} \otimes \mathcal{N}_{\lambda, s_1}$, we will instead use a randomly constructed $\mathcal{M}_{\tilde{s}}$ matrix as described in Section 3.1 to build $\mathcal{R}_{\lambda, \tilde{s}} = \mathcal{M}_{\tilde{s}} \otimes \mathcal{N}_{\lambda, s_1}$. Corollary 2 combined with the proof of Lemma 5 ensures that such a randomly constructed measurement matrix, $\mathcal{R}_{\lambda, \tilde{s}}$, will be likely to have all the properties of $\mathcal{R}_{\lambda, K}$ matrices that Algorithm 3 needs to function correctly. Hence, with high probability we will receive output from Algorithm 3 with the same approximation error bounds as derived for Theorem 7. Formalizing these ideas we obtain the following Corollary proven in Appendix G.

Corollary 4. *Suppose $f : [0, 2\pi] \rightarrow \mathbb{C}$ has $\hat{f} \in l^1$. Let $\sigma \in [2/3, 1)$ and $N, k, \epsilon^{-1} \in \mathbb{N} - \{1\}$ with $N > (k/\epsilon) \geq 2$. Algorithm 3 may be executed using a random matrix, $\mathcal{R}_{\lambda, \tilde{s}} = \mathcal{M}_{\tilde{s}} \otimes \mathcal{N}_{\lambda, s_1}$, in place of the deterministic matrix, $\mathcal{R}_{\lambda, K} = \mathcal{M}_{s_1, K} \otimes \mathcal{N}_{\lambda, s_1}$, considered above. In this case Algorithm 3 will produce an output vector, $\vec{x}_{\tilde{s}} \in \mathbb{C}^N$, that satisfies Equation 30 with probability at least σ . Both the runtime of lines 6 through 21 and the number of points in $[0, 2\pi]$ at which f will be evaluated are*

$$O\left(\frac{k}{\epsilon} \cdot \log^3 N \cdot \log\left(\frac{N}{1-\sigma}\right)\right).$$

Explicit upper bounds on the number of point evaluations are easily obtained from the proof below.

Proof: See Appendix G. \square

When executed with a random matrix, $\mathcal{R}_{\lambda, \tilde{s}}$, as input the overall runtime complexity of Algorithm 3 will be determined by both the runtime stated in Corollary 4 and the runtime of Algorithm 1. Suppose \tilde{S} is a subset of $O\left(\log\left(\frac{N}{1-\sigma}\right)\right)$ s_j values defined as per Equations 9 – 11. Then, Algorithm 1 will have a runtime complexity of

$$\begin{aligned} O\left(\sum_{i=1}^{\lambda} \sum_{s_j \in \tilde{S}} t_i \cdot s_j \log s_j\right) &= O\left(\frac{p_{q+K} \cdot \log p_{q+K} \cdot p_{\lambda}^2}{\log p_{\lambda}} \cdot \log\left(\frac{N}{1-\sigma}\right)\right) \text{ (see [15], Corollary 2)} \\ (31) \quad &= O\left(\frac{k \cdot \log_{\mathfrak{S}(k/\epsilon)} N \cdot \log^2\left(\frac{k \cdot \log N}{\epsilon}\right) \cdot \ln^2\left(\frac{\epsilon N}{k}\right)}{\epsilon \cdot \ln \ln\left(\frac{\epsilon N}{k}\right)} \cdot \log\left(\frac{N}{1-\sigma}\right)\right) \text{ (see Equation 13, Lemma 4).} \end{aligned}$$

Thus, if we are willing to fail with probability at most $1 - \sigma = 1/N^{O(1)}$, then Algorithm 3 executed with a random input matrix will have a total runtime complexity of $O\left((k/\epsilon) \cdot \log^4 N \cdot \log\left(\frac{k \cdot \log N}{\epsilon}\right)\right)$.

6. HIGHER DIMENSIONAL FOURIER TRANSFORMS

In this section we will consider methods for approximating the Fourier transform of a periodic function of D variables, $f : [0, 2\pi]^D \rightarrow \mathbb{C}$. To begin, we will demonstrate how to approximate the Fourier transform of f by calculating the discrete Fourier transform of a related one-dimensional function, $f_{\text{new}} : [0, 2\pi] \rightarrow \mathbb{C}$. This dimensionality reduction technique for multidimensional Fourier transforms will ultimately enable us to quickly approximate \hat{f} by applying the methods of Section 5 to f 's related one dimensional function f_{new} . The end result will be a set of algorithms for approximating \hat{f} whose runtimes scale polynomially in the input dimension D .

Suppose that the Fourier transform of f above, $\hat{f} : \mathbb{Z}^D \rightarrow \mathbb{C}$, is near zero for all integer points outside of the D -dimensional cubic lattice $([-M/2, M/2] \cap \mathbb{Z})^D$. In order to help us approximately recover \hat{f} we will choose D pairwise relatively prime integers, $P_1, \dots, P_D \in \mathbb{N}$, with the property that $P_d > M \cdot D$ for all $d \in [1, D] \cap \mathbb{N}$. Set $\tilde{N} = \prod_{d=1}^D P_d$. Furthermore, let $y^{-1 \bmod p} \in [0, p) \cap \mathbb{N}$ denote the multiplicative inverse of $(y \bmod p) \in \mathbb{Z}_p$ when it exists. Note that $y^{-1 \bmod p}$ will exist whenever y is relatively prime to p .

We may now define the function $f_{\text{new}} : [0, 2\pi] \rightarrow \mathbb{C}$ to be

$$(32) \quad f_{\text{new}}(x) = f\left(\frac{\tilde{N}}{P_1}x, \frac{\tilde{N}}{P_2}x, \dots, \frac{\tilde{N}}{P_D}x\right).$$

Considering the Fourier transform of f_{new} we can see that

$$(33) \quad \begin{aligned} \hat{f}_{\text{new}}(\omega) &= \frac{1}{2\pi} \int_0^{2\pi} e^{-i\omega x} f_{\text{new}}(x) dx = \frac{1}{2\pi} \sum_{(\omega_1, \dots, \omega_D) \in \mathbb{Z}^D} \hat{f}(\omega_1, \dots, \omega_D) \int_0^{2\pi} e^{-ix(\omega - \sum_{d=1}^D \frac{\tilde{N}}{P_d} \omega_d)} dx \\ &= \sum_{(\omega_1, \dots, \omega_D) \in \mathbb{Z}^D \text{ s.t. } \omega = \sum_{d=1}^D \frac{\tilde{N}}{P_d} \omega_d} \hat{f}(\omega_1, \dots, \omega_D). \end{aligned}$$

Recall that we are primarily interested in capturing the information about \hat{f} inside $([-M/2, M/2] \cap \mathbb{Z})^D$. Looking at the $\omega \in \mathbb{Z}$ for which \hat{f}_{new} can be impacted by $(\omega_1, \dots, \omega_D) \in ([-M/2, M/2] \cap \mathbb{Z})^D$ we can see that

$$|\omega| \leq \sum_{d=1}^D \left| \frac{\omega_d \tilde{N}}{P_d} \right| \leq \sum_{d=1}^D \frac{M \tilde{N}}{2P_d} < \sum_{d=1}^D \frac{\tilde{N}}{2D} = \frac{\tilde{N}}{2}.$$

Hence, we may consider f_{new} to have an effective bandwidth of \tilde{N} .

More importantly, there is a bijective correspondence between the integer lattice points, $(\omega_1, \dots, \omega_D) \in ([-M/2, M/2] \cap \mathbb{Z})^D$, and their representative frequency, $\omega \in [-\tilde{N}/2, \tilde{N}/2] \cap \mathbb{Z}$, in \hat{f}_{new} . Define the function

$$g : \left(-\frac{P_1}{2}, \frac{P_1}{2}\right] \cap \mathbb{N} \times \dots \times \left(-\frac{P_D}{2}, \frac{P_D}{2}\right] \cap \mathbb{N} \rightarrow \left(-\frac{\tilde{N}}{2}, \frac{\tilde{N}}{2}\right] \cap \mathbb{N}$$

to be

$$g(x_1, \dots, x_D) = \left(\sum_{d=1}^D \left(\frac{\tilde{N}}{P_d} \right) \cdot x_d \right) \bmod \tilde{N}.$$

The Chinese Remainder Theorem tells us that g is a well-defined bijection. Furthermore, it is not difficult to see that

$$g^{-1}(x) = \left(x \cdot (\tilde{N}/P_1)^{-1 \bmod P_1} \bmod P_1, \dots, x \cdot (\tilde{N}/P_D)^{-1 \bmod P_D} \bmod P_D \right).$$

Thus, we have $\hat{f}_{\text{new}}(\omega) \approx \hat{f}(g^{-1}(\omega))$.

We now have a three-step algorithm for finding a sparse Fourier approximation for any function $f : [0, 2\pi]^D \rightarrow \mathbb{C}$. All we must do is: (i) Implicitly create f_{new} as per Equation 32, (ii) Use the techniques from Section 5 to approximate \hat{f}_{new} , and then (iii) Use the approximation for \hat{f}_{new} to approximate \hat{f} via Equation 33. The following theorem summarizes some of the results one can achieve by utilizing this approach.

Theorem 8. *Suppose $f : [0, 2\pi]^D \rightarrow \mathbb{C}$ is bandlimited so that $\hat{f}(\omega_1, \dots, \omega_D) = 0$ if $(\omega_1, \dots, \omega_D) \notin ([-\frac{M}{2}, \frac{M}{2}] \cap \mathbb{Z})^D$. Define \tilde{N} as above and suppose that $\tilde{N}, k, \epsilon^{-1} \in \mathbb{N} - \{1\}$ with $\tilde{N} > (k/\epsilon)^2 \geq 4$. Then, Algorithm 3 combined with the bijective mapping, g , above will output an $\vec{x}_S \in \mathbb{C}^{\tilde{N}}$ satisfying*

$$(34) \quad \|\hat{f} - (\vec{x}_S \circ g)\|_2 \leq \|\hat{f} - \hat{f}_k^{\text{opt}}\|_2 + \frac{22\epsilon \cdot \|\hat{f} - \hat{f}_{(k/\epsilon)}^{\text{opt}}\|_1}{\sqrt{k}}.$$

Both the runtime of lines 6 – 21, and the number of points in $[0, 2\pi]^D$ at which f will be evaluated, will be

$$O\left(\frac{k^2 \cdot D^4 \cdot \log^4(MD)}{\log\left(\frac{k}{\epsilon}\right) \cdot \epsilon^2}\right).$$

If succeeding with probability $\sigma \in [2/3, 1)$ is sufficient, and $\tilde{N} > (k/\epsilon) \geq 2$, Algorithm 3 may instead be executed using a random matrix $\mathcal{R}_{\lambda, \delta}$. In this case Algorithm 3 will produce an output vector, $\vec{x}_S \in \mathbb{C}^{\tilde{N}}$, that satisfies Equation 34 with probability at least σ . Both the runtime of lines 6 – 21, and the number of points in $[0, 2\pi]^D$ at which f will be evaluated, will be

$$O\left(\frac{k \cdot D^4}{\epsilon} \cdot \log^3(MD) \cdot \log\left(\frac{MD}{1-\sigma}\right)\right).$$

Finally, if an exponential runtime of $\Omega((DM)^D)$ is acceptable, we note that both Theorem 6 and Corollary 3 can also be adapted to recovering $f : [0, 2\pi]^D \rightarrow \mathbb{C}$ by substituting N with $\Theta((MD)^D)$ everywhere in their statements.

Proof: See Appendix H. \square

Note that traditional FFT algorithms (e.g., [8, 23, 1]) require $\Omega(M^D)$ -time to calculate the Fourier transform of a bandlimited function $f : [0, 2\pi]^D \rightarrow \mathbb{C}$. In contrast, Theorem 8 allows us to approximate \hat{f} using exponentially fewer (in D) operations. Hence, if f has a relatively sparse Fourier representation (e.g., if \hat{f} is dominated by $k = M^{o(D)}$ energetic frequencies), Theorem 8 allows \hat{f} to be accurately approximated much more quickly than possible using standard techniques.

7. CONCLUSION

In conclusion, it is worth pointing out that the methods developed in this paper for approximating the Fourier transforms of periodic functions are also applicable to the approximation of functions which have accurate sparse representations in related bases. For example, all the theorems proven herein will also apply to functions with sparsely representable Cosine or Chebyshev expansions (see [3] for an in depth discussion of the relationships between these series expansions). Hence, we have also implicitly constructed sublinear-time algorithms for approximating these related transforms.

ACKNOWLEDGMENTS

I would like to thank Martin Strauss and Holger Rauhut for answering questions about their work. I would also like to thank Martin Strauss, Holger Rauhut, and Tsvetanka Sendova for helpful discussions and advice. The vast majority of this paper was written while the author was supported as a postdoctoral fellow at the Institute for Mathematics and its Applications (IMA).

REFERENCES

- [1] L. I. Bluestein. A Linear Filtering Approach to the Computation of Discrete Fourier Transform. *IEEE Transactions on Audio and Electroacoustics*, 18:451–455, 1970.
- [2] T. Blumensath and M. E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265 – 274, 2009.
- [3] J. P. Boyd. *Chebyshev and Fourier Spectral Methods*. Dover Publications, Inc., 2001.
- [4] E. Candes, J. Romberg, and T. Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inform. Theory*, 52:489–509, 2006.
- [5] E. Candes, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.
- [6] E. Candes and T. Tao. Near optimal signal recovery from random projections: Universal encoding strategies? *IEEE Trans. on Information Theory*, 2006.
- [7] A. Cohen, W. Dahmen, and R. DeVore. Compressed Sensing and Best k -term Approximation. *Journal of the American Mathematical Society*, 22(1):211–231, January 2008.
- [8] J. Cooley and J. Tukey. An algorithm for the machine calculation of complex Fourier series. *Math. Comput.*, 19:297–301, 1965.
- [9] D. Dor and U. Zwick. Median selection requires $(2+\epsilon)n$ comparisons. *SIAM Journal of Discrete Mathematics*, 14:125–134, 2001.
- [10] P. Dusart. The k^{th} prime is greater than $k(\ln k + \ln \ln k - 1)$ for $k \geq 2$. *Mathematics of Computation*, 68(225), 1999.
- [11] M. Fornasier and H. Rauhut. Compressive sensing. *Handbook of Mathematical Methods in Imaging*, Springer, to appear.
- [12] A. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, and M. Strauss. Near-optimal sparse Fourier estimation via sampling. *ACM STOC*, pages 152–161, 2002.
- [13] A. Gilbert, S. Muthukrishnan, and M. Strauss. Improved time bounds for near-optimal sparse Fourier representations. *Proceedings of SPIE Wavelets XI*, 2005.
- [14] M. A. Iwen. Combinatorial sublinear-time fourier algorithms. *Foundations of Computational Mathematics*, 10(3):303 – 338, 2010.

- [15] M. A. Iwen and C. V. Spencer. Improved bounds for a deterministic sublinear-time sparse fourier algorithm. In *Conference on Information Sciences and Systems (CISS)*, 2008.
- [16] D. Knuth. *The Art of Computer Programming. Volume 3: Sorting and Searching*. Addison-Wesley, 1998.
- [17] S. Kunis and H. Rauhut. Random Sampling of Sparse Trigonometric Polynomials II - Orthogonal Matching Pursuit versus Basis Pursuit. *Foundations of Computational Mathematics*, 8(6):737–763, 2008.
- [18] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
- [19] D. Needell and J. A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3), 2008.
- [20] D. Needell and R. Vershynin. Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit. *Foundations of Computational Mathematics*, 9:317–334, 2009.
- [21] D. Needell and R. Vershynin. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. *IEEE Journal of Selected Topics in Signal Processing*, to appear.
- [22] I. Niven, H. S. Zuckerman, and H. L. Montgomery. *An Introduction to The Theory of Numbers*. John Wiley & Sons, Inc., 1991.
- [23] L. Rabiner, R. Schafer, and C. Rader. The Chirp z-Transform Algorithm. *IEEE Transactions on Audio and Electroacoustics*, AU-17(2):86–92, June 1969.
- [24] H. Rauhut. Compressive sensing and structured random matrices. *Theoretical Foundations and Numerical Methods for Sparse Recovery*, vol. 9 in Radon Series Comp. Appl. Math.:1–92, deGruyter, 2010.
- [25] G. Robin. Estimation de la fonction de tchebychef θ sur le k-ieme nombre premier et grandes valeurs de la fonctions $\omega(n)$, nombre de diviseurs premiers de n. *Acta Arithmetica*, 42(4):367 – 389, 1983.
- [26] M. Rudelson and R. Vershynin. Sparse reconstruction by convex relaxation: Fourier and gaussian measurements. In *40th Annual Conference on Information Sciences and Systems (CISS)*, 2006.
- [27] J. Tropp and A. Gilbert. Signal recovery from partial information via orthogonal matching pursuit. *IEEE Trans. Info. Theory*, 53(12):4655–4666, Dec. 2007.

APPENDIX A. PROOF OF THEOREM 5

Let $\pi(n)$ be the number of primes no greater than n . In [10] it is shown that

$$\frac{n}{\ln n} \left(1 + \frac{0.992}{\ln n}\right) \leq \pi(n) \leq \frac{n}{\ln n} \left(1 + \frac{1.2762}{\ln n}\right)$$

for all $n \geq 599$. Using this result (in combination with numerical tests for $n < 600$) we obtain the following bounds for $q + K$ and q (see Equation 11).

$$(35) \quad q + K \leq \pi(k/\epsilon) + K + 1 \leq \frac{k \lfloor \log_{(k/\epsilon)} N \rfloor}{\epsilon} \left(c + \frac{1}{\ln(k/\epsilon) \cdot \lfloor \log_{(k/\epsilon)} N \rfloor} + \frac{1.2762}{\ln^2(k/\epsilon) \cdot \lfloor \log_{(k/\epsilon)} N \rfloor} + \frac{2 \cdot \epsilon}{k \cdot \lfloor \log_{(k/\epsilon)} N \rfloor} \right).$$

and

$$(36) \quad q \geq \pi(k/\epsilon) \geq \max \left\{ \frac{k}{\epsilon \cdot \ln(k/\epsilon)} \left(1 + \frac{0.992}{\ln(k/\epsilon)} - \frac{8.85 \cdot \epsilon}{k}\right), 1 \right\}.$$

Continuing, we can bound m if our s_j values are chosen to be primes as above by noting that

$$(37) \quad \begin{aligned} \sum_{j=1}^{q-1} p_j &\geq \sum_{j=1}^{q-1} j \cdot \ln(j) \quad (\text{see [10]}) \\ &\geq \int_1^{q-1} x \cdot \ln x \, dx \geq \frac{(q-1)^2}{2} \left(\ln(q-1) - \frac{1}{2} \right) \end{aligned}$$

and

$$(38) \quad \begin{aligned} \sum_{j=1}^{q+K-1} p_j &\leq 10 + \sum_{j=4}^{q+K-1} j \cdot \ln(p_j) \quad (\text{see [10]}) \\ &\leq 10 + \ln(p_{q+K}) \cdot \left(\sum_{j=4}^{q+K-1} j \right) \leq \frac{(q+K-1)(q+K)}{2} \cdot \ln \left((q+K) \cdot (\ln(q+K) + \ln \ln(q+K)) \right) \quad (\text{see [10]}) \end{aligned}$$

$$(39) \quad \leq \frac{3}{4} (q+K)^2 \cdot \ln(q+K).$$

Using Equation 35 together with Equation 39 finishes the proof. More specifically, we have that

$$\left(c + \frac{1}{\ln(k/\epsilon) \cdot \lceil \log_{(k/\epsilon)} N \rceil} + \frac{1.2762}{\ln^2(k/\epsilon) \cdot \lceil \log_{(k/\epsilon)} N \rceil} + \frac{2 \cdot \epsilon}{k \cdot \lceil \log_{(k/\epsilon)} N \rceil} \right) \leq \left(c + \frac{1}{\ln 4} + \frac{1.2762}{\ln^2 4} + \frac{1}{2} \right) \leq (c + 1.89).$$

Therefore, we can see that

$$m \leq \sum_{j=1}^{q+K-1} p_j \leq \frac{3(c + 1.89)^2 \cdot k^2 \lceil \log_{(k/\epsilon)} N \rceil^2}{4 \cdot \epsilon^2} \cdot \ln \left(\frac{(c + 1.89) \cdot k \lceil \log_{(k/\epsilon)} N \rceil}{\epsilon} \right)$$

as we wished to prove.

APPENDIX B. PROOF OF COROLLARY 2

We prove the result via an argument similar to the one used to prove Lemma 2 in [14]. Fix $n \in S$. We will select our multiset of s_j values, \tilde{S} , by independently choosing l elements of $\{s_1, s_2, \dots, s_K\}$ uniformly at random with replacement. The first element chosen for \tilde{S} will be denoted s_{j_1} , the second s_{j_2} , and so forth. Let Q_h^n be the random variable indicating whether the s_{j_h} value selected for \tilde{S} satisfies

$$(40) \quad \left| (\mathcal{M}_{s_1, K, n, \vec{x}})_{j_h} - x_n \right| \leq \frac{\epsilon \cdot \left\| \vec{x} - \vec{x}_{(k/\epsilon)}^{\text{opt}} \right\|_1}{k}.$$

Therefore,

$$Q_h^n = \begin{cases} 1 & \text{if } s_{j_h} \text{ satisfies Property 40} \\ 0 & \text{otherwise} \end{cases}.$$

Theorem 4 tells us that $\mathbb{P}[Q_h^n = 1] > \frac{6}{7}$. Furthermore, $\mu = \mathbb{E}[\sum_{h=1}^l Q_h^n] \geq \frac{6l}{7}$.

Using the Chernoff bound (see [18]) we get that the probability of

$$\sum_{h=1}^l Q_h^n < \frac{4 \cdot l}{7}$$

is less than $e^{-\frac{l}{18}} \leq e^{-\frac{l}{21}} \leq \frac{1-\sigma}{|S|}$. Since $l > 21$ we can see that $\sum_{h=1}^l Q_h^n$ will be less than $\frac{l+1}{2}$ with probability less than $\frac{1-\sigma}{|S|}$. Hence, Property 40 will be satisfied by more than $l/2$ of the $s_{j_h} \in \tilde{S}$ with high probability. Applying the union bound shows that the majority of the entries in \tilde{S} will indeed satisfy Property 40 for all $n \in S$ with probability at least σ . The result follows.

APPENDIX C. PROOF OF COROLLARY 3

Apply Corollary 2 with $c = 14$, $\vec{x} = \vec{f}$, and $S = \left(-\lceil \frac{N}{2} \rceil, \lfloor \frac{N}{2} \rfloor\right) \cap \mathbb{Z}$ to obtain \tilde{S} , a multiset of $\lceil 21 \cdot \ln\left(\frac{N}{1-\sigma}\right) \rceil$ s_j values. With probability at least σ more than half (with multiplicity) of the entries of $\mathcal{M}_{\tilde{S}, \omega \vec{f}}$ will estimate \hat{f}_ω to within $(\epsilon/k) \cdot \left\| \vec{f} - \vec{f}_{(k/\epsilon)}^{\text{opt}} \right\|_1$ precision for all $\omega \in \left(-\lceil \frac{N}{2} \rceil, \lfloor \frac{N}{2} \rfloor\right) \cap \mathbb{Z}$. Furthermore, $\mathcal{M}_{\tilde{S}, \vec{f}}$ can still be approximately computed using Algorithm 1 if only the unique s_j values in \tilde{S} are given as the relatively prime inputs. In this case Lemma 3 will also still hold. Taken all together we can see that with probability at least σ all N x_ω values produced by lines 6 and 7 of Algorithm 2 will have

$$\left| x_\omega - \hat{f}_\omega \right| \leq \sqrt{2} \cdot \left(\frac{\epsilon \cdot \left\| \vec{f} - \vec{f}_{(k/\epsilon)}^{\text{opt}} \right\|_1}{k} + \left\| \vec{f} - \tilde{f} \right\|_1 \right).$$

The Equation 21 error bound now follows from the proof of Theorem 6.

To upper bound the number of required function evaluations we will bound the number of rows for a particular $\mathcal{M}_{\tilde{S}}$ matrix constructed with primes as per Section 3. In particular, we will assume that \tilde{S} contains at most $\lceil 21 \cdot \ln\left(\frac{N}{1-\sigma}\right) \rceil$

individual s_j values defined as in Equations 9 – 11 with $K = 14 \cdot (k/\epsilon) \lfloor \log_{s_1} N \rfloor + 1$. In this case Equation 35 together with results from [10] tell us that s_K is at most

$$(41) \quad 15.89 \cdot \frac{k \lfloor \log_{(k/\epsilon)} N \rfloor}{\epsilon} \cdot \left(\ln \left(\frac{15.89 \cdot k \lfloor \log_{(k/\epsilon)} N \rfloor}{\epsilon} \right) + \ln \ln \left(\frac{15.89 \cdot k \lfloor \log_{(k/\epsilon)} N \rfloor}{\epsilon} \right) \right).$$

The stated upper bound on the number of required function evaluations follows. The stated runtime follows from the fact that each line 6 and 7 median now only involves $O\left(\log\left(\frac{N}{1-\sigma}\right)\right)$ values.

APPENDIX D. PROOF OF LEMMA 4

We can always set $t_1 = p_1 < \dots < t_\lambda = p_\lambda$. In this case we require that $p_\lambda < s_1 \leq$ the smallest prime factor of s_1, \dots, s_K . Secondly, we require that $\sum_{i=1}^\lambda \ln p_i \geq \ln\left(\frac{N}{s_1}\right)$. Using results from [25] it is easily verified that

$$\sum_{i=1}^\lambda \ln p_i \geq \lambda \cdot (\ln \lambda - 1)$$

for all $\lambda \in \mathbb{N}^+$. Setting $\lambda = \left\lceil 3 \ln\left(\frac{N}{s_1}\right) / \ln \ln\left(\frac{N}{s_1}\right) \right\rceil$ in the equation above we can see that

$$\sum_{i=1}^\lambda \ln p_i \geq \ln\left(\frac{N}{s_1}\right) \cdot 3 \left(1 - \frac{\ln \ln \ln\left(\frac{N}{s_1}\right)}{\ln \ln\left(\frac{N}{s_1}\right)} \right) \geq \ln\left(\frac{N}{s_1}\right)$$

as long as $N/s_1 \geq 3$. Hence, if we choose our t_i values to be the first λ primes the second requirement will be satisfied.

Results from [10] then tell us that

$$t_\lambda = p_\lambda \leq p_{\lceil 3 \ln(N/s_1) / \ln \ln(N/s_1) \rceil} \leq \left\lceil 3 \cdot \frac{\ln(N/s_1)}{\ln \ln(N/s_1)} \right\rceil \cdot \left(\ln \left\lceil 3 \cdot \frac{\ln(N/s_1)}{\ln \ln(N/s_1)} \right\rceil + \ln \ln \left\lceil 3 \cdot \frac{\ln(N/s_1)}{\ln \ln(N/s_1)} \right\rceil \right) < s_1.$$

Therefore, the prime t_i values we have selected will also satisfy the first requirement above. To bound the smallest possible number of rows we note that

$$\hat{m} \leq 1 + \sum_{i=1}^{\lceil 3 \ln(N/s_1) / \ln \ln(N/s_1) \rceil} p_i \leq \frac{3}{4} \left(\left\lceil 3 \cdot \frac{\ln(N/s_1)}{\ln \ln(N/s_1)} \right\rceil + 1 \right)^2 \cdot \ln \left(\left\lceil 3 \cdot \frac{\ln(N/s_1)}{\ln \ln(N/s_1)} \right\rceil + 1 \right) + 1 \text{ (see Equation 39).}$$

The stated result follows.

APPENDIX E. PROOF OF LEMMA 5

In addition to \vec{x} we will also consider $\vec{y} \in \mathbb{C}^N$ defined by

$$y_{n'} = |x_{n'}| \text{ for all } n' \in [0, N) \cap \mathbb{N}.$$

Note that \vec{y} and \vec{x} will not only share the same optimal (k/ϵ) -term support subset, $S_{(k/\epsilon)}^{\text{opt}} \subset [0, N) \cap \mathbb{N}$, but will also have $\left\| \vec{x} - \vec{x}_{(k/\epsilon)}^{\text{opt}} \right\|_1 = \left\| \vec{y} - \vec{y}_{(k/\epsilon)}^{\text{opt}} \right\|_1$. Theorem 4 tells us that more than $\frac{K}{2}$ entries of $\mathcal{M}_{s_1, K, n} \cdot \vec{y}$ will estimate y_n to within $\frac{\epsilon \cdot \left\| \vec{y} - \vec{y}_{(k/\epsilon)}^{\text{opt}} \right\|_1}{k} = \bar{\delta} = \frac{\epsilon \cdot \left\| \vec{x} - \vec{x}_{(k/\epsilon)}^{\text{opt}} \right\|_1}{k}$ precision. Let $(\mathcal{M}_{s_1, K, n} \cdot \vec{y})_{j'}$ for $j' \in [1, K) \cap \mathbb{N}$ be one of these $\frac{K}{2}$ entries. The proof of Lemma 2 tells us that the row associated with this entry also has the property that

$$\left| (\mathcal{M}_{s_1, K, n} \cdot \vec{x})_{j'} - x_n \right| \leq \sum_{n' \equiv n \pmod{s_j}, n' \notin S_{(k/\epsilon)}^{\text{opt}}, n' \neq n} y_{n'} = \left| (\mathcal{M}_{s_1, K, n} \cdot \vec{y})_{j'} - y_n \right| \leq \bar{\delta}.$$

Therefore, we have established property (1).

Considering property (2) for this j' we can see that for all $i \in [1, \lambda] \cap \mathbb{N}$ we will have

$$\begin{aligned} \left| \tilde{r}_{i,j',n \bmod t_i s_{j'}} \cdot \vec{x} - x_n \right| &= \left| \sum_{n' \equiv n \bmod t_i s_{j'}, n' \notin S_{(k/\epsilon)}^{\text{opt}}, n' \neq n} x_{n'} \right| \leq \sum_{n' \equiv n \bmod t_i s_{j'}, n' \notin S_{(k/\epsilon)}^{\text{opt}}, n' \neq n} y_{n'} \\ &\leq \sum_{n' \equiv n \bmod s_{j'}, n' \notin S_{(k/\epsilon)}^{\text{opt}}, n' \neq n} y_{n'} = \left| (\mathcal{M}_{s_1, K, n} \cdot \vec{y})_{j'} - y_n \right| \leq \bar{\delta}. \end{aligned}$$

Finally, to verify property (3) we can bound $\left| (r_{j', n \bmod s_{j'}} \oplus \tilde{r}_{i,h}) \cdot \vec{x} \right|$ from above for all $i \in [1, \lambda] \cap \mathbb{N}$ and $h \in [0, t_i] \cap (\mathbb{N} - \{n \bmod t_i\})$ by

$$\left| \sum_{n' \equiv n \bmod s_{j'}, n' \equiv h \bmod t_i, n' \notin S_{(k/\epsilon)}^{\text{opt}}} x_{n'} \right| \leq \sum_{n' \equiv n \bmod s_{j'}, n' \notin S_{(k/\epsilon)}^{\text{opt}}, n' \neq n} y_{n'} = \left| (\mathcal{M}_{s_1, K, n} \cdot \vec{y})_{j'} - y_n \right| \leq \bar{\delta}.$$

Hence, we can see that all three properties will indeed hold for at least $\frac{K}{2}$ rows of $\mathcal{M}_{s_1, K, n}$.

APPENDIX F. PROOF OF THEOREM 7

Let δ be defined as

$$\delta = \frac{\epsilon \cdot \left\| \vec{f} - \vec{f}_{(k/\epsilon)}^{\text{opt}} \right\|_1}{k} + \left\| \hat{f} - \tilde{f} \right\|_1.$$

Furthermore, suppose $j \in [1, K] \cap \mathbb{N}$ and $h \in [0, s_j)$ correspond to an $\omega_{jh} \in \left(-\left\lfloor \frac{N}{2} \right\rfloor, \left\lfloor \frac{N}{2} \right\rfloor \right) \cap \mathbb{Z}$ which is reconstructed more than $\frac{K}{2}$ times by line 12 of Algorithm 3. As a consequence of Lemmas 3 and 5 we can see that more than half of the entries of $\mathcal{G}_{\lambda, K, \omega_{jh}} \tilde{\Psi} \vec{A}$ produced in line 4 will satisfy $\left| (\mathcal{G}_{\lambda, K, \omega_{jh}} \tilde{\Psi} \vec{A})_j - \hat{f}_{\omega_{jh}} \right| \leq \delta$. Therefore, the $x_{\omega_{jh}}$ value produced by lines 16 and 17 will have

$$(42) \quad \left| x_{\omega_{jh}} - \hat{f}_{\omega_{jh}} \right| \leq \sqrt{2} \cdot \delta.$$

Since Equation 42 will hold for all $\omega \in \left(-\left\lfloor \frac{N}{2} \right\rfloor, \left\lfloor \frac{N}{2} \right\rfloor \right) \cap \mathbb{Z}$ reconstructed more than $\frac{K}{2}$ times, we can begin to bound the approximation error by

$$\begin{aligned} \left\| \vec{f} - \vec{x}_S \right\|_2 &\leq \left\| \vec{f} - \vec{f}_S \right\|_2 + \left\| \vec{f}_S - \vec{x}_S \right\|_2 \leq \left\| \vec{f} - \vec{f}_S \right\|_2 + 2\sqrt{k} \cdot \delta \\ (43) \quad &= \sqrt{\left\| \vec{f} - \vec{f}_k^{\text{opt}} \right\|_2^2 + \sum_{\omega \in S_k^{\text{opt}} - S} |\hat{f}_\omega|^2 - \sum_{\tilde{\omega} \in S - S_k^{\text{opt}}} |\hat{f}_{\tilde{\omega}}|^2} + 2\sqrt{k} \cdot \delta. \end{aligned}$$

In order to make additional progress on Equation 43 we must now consider the possible magnitudes of \hat{f} entries at indices in $S - S_k^{\text{opt}}$ and $S_k^{\text{opt}} - S$.

Suppose $\omega \in S_k^{\text{opt}} - S \neq \emptyset$. In this case either (i) $|\hat{f}_\omega| \leq 4\delta$, or (ii) $|\hat{f}_\omega| > 4\delta$ in which case Lemma 6 guarantees that ω will be identified by lines 6 through 14 of Algorithm 3. Once identified, an $\tilde{\omega} \in S_k^{\text{opt}}$ will always be placed in S unless at least $k+1$ other distinct identified elements, $\tilde{\omega} \notin S_k^{\text{opt}}$, have the property that $|x_{\tilde{\omega}}| \geq |x_\omega|$. Thus, if (ii) occurs then

$$(44) \quad \left| \hat{f}_{\omega_k} \right| + \sqrt{2} \cdot \delta \geq \left| \hat{f}_{\tilde{\omega}} \right| + \sqrt{2} \cdot \delta \geq \left| \hat{f}_\omega \right| - \sqrt{2} \cdot \delta \geq \left| \hat{f}_{\omega_k} \right| - \sqrt{2} \cdot \delta$$

will hold for all $\tilde{\omega} \in S - S_k^{\text{opt}}$. The end result is that if $\omega \in S_k^{\text{opt}} - S$ then either $|\hat{f}_\omega| \leq 4\delta$, or else \hat{f}_ω is roughly the same magnitude as \hat{f}_{ω_k} (up to a $O(\delta)$ tolerance). Furthermore, because line 20 chooses $2k$ elements for S whenever possible, we can see that $S - S_k^{\text{opt}}$ must contain at least

$$2 \cdot \left| \left(S_k^{\text{opt}} \cap \left\{ \omega \mid |\hat{f}_\omega| > 4\delta \right\} \right) - S \right|$$

elements, $\tilde{\omega}$, all of which satisfy Equation 44 for every $\omega \in \left(S_k^{\text{opt}} \cap \left\{\omega \mid |\hat{f}_\omega| > 4\delta\right\}\right) - S$. We are now ready to give Equation 43 further consideration.

If $S_k^{\text{opt}} - S = \emptyset$ we are finished. Otherwise, if $S_k^{\text{opt}} - S \neq \emptyset$, we can bound the squared l^2 -norm of $\vec{f}_{S - S_k^{\text{opt}}}$ from below by

$$\sum_{\tilde{\omega} \in S - S_k^{\text{opt}}} |\hat{f}_{\tilde{\omega}}|^2 \geq 2 \cdot \left| \left(S_k^{\text{opt}} \cap \left\{\omega \mid |\hat{f}_\omega| > 4\delta\right\} \right) - S \right| \cdot \left(|\hat{f}_{\omega_k}| - 2\sqrt{2} \cdot \delta \right)^2 = \mathbf{A}.$$

Furthermore, we can upper bound the squared l^2 -norm of $\vec{f}_{S_k^{\text{opt}} - S}$ by

$$\left| \left(S_k^{\text{opt}} \cap \left\{\omega \mid |\hat{f}_\omega| > 4\delta\right\} \right) - S \right| \cdot \left(|\hat{f}_{\omega_k}| + 2\sqrt{2} \cdot \delta \right)^2 + \left| \left(S_k^{\text{opt}} \cap \left\{\omega \mid |\hat{f}_\omega| \leq 4\delta\right\} \right) - S \right| \cdot 16\delta^2 \geq \sum_{\omega \in S_k^{\text{opt}} - S} |\hat{f}_\omega|^2.$$

Let $\mathbf{B} = \left| \left(S_k^{\text{opt}} \cap \left\{\omega \mid |\hat{f}_\omega| > 4\delta\right\} \right) - S \right| \cdot \left(|\hat{f}_{\omega_k}| + 2\sqrt{2} \cdot \delta \right)^2$. We will now concentrate on bounding

$$\mathbf{C} = \sum_{\omega \in S_k^{\text{opt}} - S} |\hat{f}_\omega|^2 - \sum_{\tilde{\omega} \in S - S_k^{\text{opt}}} |\hat{f}_{\tilde{\omega}}|^2.$$

If $\mathbf{A} \geq \mathbf{B}$ then $\mathbf{C} \leq 16k \cdot \delta^2$. Otherwise, if $\mathbf{A} < \mathbf{B}$ then

$$|\hat{f}_{\omega_k}|^2 - 12\sqrt{2}\delta \cdot |\hat{f}_{\omega_k}| + 8\delta^2 < 0$$

which can only happen if $|\hat{f}_{\omega_k}| \in \left((6\sqrt{2} - 8) \cdot \delta, (6\sqrt{2} + 8) \cdot \delta \right)$. Hence, $\mathbf{A} < \mathbf{B}$ implies that $\mathbf{C} \leq k \cdot (8\sqrt{2} + 8)^2 \cdot \delta^2$.

Finishing our error analysis, we can see that in the worst possible case Equation 43 will remain bounded by

$$\left\| \vec{f} - \vec{x}_S \right\|_2 \leq \sqrt{\left\| \vec{f} - \vec{f}_k^{\text{opt}} \right\|_2^2 + k \cdot (8\sqrt{2} + 8)^2 \cdot \delta^2 + 2\sqrt{k} \cdot \delta} \leq \left\| \vec{f} - \vec{f}_k^{\text{opt}} \right\|_2 + 22\sqrt{k} \cdot \delta.$$

The error bound stated in Equation 30 follows. The upper bound on the number of point evaluations of f follows from an application of Lemma 4 and Theorem 5 with $c = 4$.

We will begin bounding the runtime of Algorithm 3 by bounding the runtime of lines 15 through 21. Line 12 of Algorithm 3 will be executed a total of $O\left(\frac{k^2 \cdot \log_{(k/\epsilon)}^2 N \cdot \log\left(\frac{k \ln N}{\epsilon}\right)}{\epsilon^2}\right)$ times (see Equation 14). Therefore, lines 16 and 17 will be executed $O\left(\frac{k \cdot \log_{(k/\epsilon)} N \cdot \log\left(\frac{k \ln N}{\epsilon}\right)}{\epsilon}\right)$ times apiece. Each such median operation can be accomplished in $O(K \cdot \lambda)$ time using a median-of-medians algorithm (e.g., see [9]). Therefore, the total runtime of lines 15 through 21 will be $O\left(\frac{k^2 \cdot \log_{(k/\epsilon)}^2 N \cdot \log\left(\frac{k \ln N}{\epsilon}\right) \cdot \log(\epsilon N/k)}{\epsilon^2 \cdot \log \log(\epsilon N/k)}\right)$. Turning our attention to lines 6 through 14, we note that their runtime will be dominated by the $O\left(\frac{k^2 \cdot \log_{(k/\epsilon)}^2 N \cdot \log\left(\frac{k \ln N}{\epsilon}\right) \cdot \log(\epsilon N/k)}{\epsilon^2 \cdot \log \log(\epsilon N/k)}\right)$ executions of line 9. Therefore, the total runtime of lines 6 through 14 will be $O\left(\frac{k^2 \cdot \log_{(k/\epsilon)}^2 N \cdot \log\left(\frac{k \ln N}{\epsilon}\right) \cdot \log^2(\epsilon N/k)}{\epsilon^2 \cdot \log \log(\epsilon N/k)}\right)$ (see [15] and Lemma 4). The stated overall runtime of lines 6 through 21 follows.

APPENDIX G. PROOF OF COROLLARY 4

Define $\vec{|f|} \in \mathbb{R}^N$ by

$$\left(\vec{|f|} \right)_\omega = |\hat{f}_\omega| \text{ for all } \omega \in \left(-\left\lfloor \frac{N}{2} \right\rfloor, \left\lfloor \frac{N}{2} \right\rfloor \right) \cap \mathbb{Z}.$$

Clearly \vec{f} and $\vec{|f|}$ will both have the same optimal (k/ϵ) -term support subset, $S_{(k/\epsilon)}^{\text{opt}} \subset [0, N] \cap \mathbb{N}$. Similarly, it is easy to

see that $\left\| \vec{f} - \vec{f}_{(k/\epsilon)}^{\text{opt}} \right\|_1 = \left\| \vec{|f|} - \left(\vec{|f|} \right)_{(k/\epsilon)}^{\text{opt}} \right\|_1$. Apply Corollary 2 with $c = 14$, $\vec{x} = \vec{|f|}$, and $S = \left(-\left\lfloor \frac{N}{2} \right\rfloor, \left\lfloor \frac{N}{2} \right\rfloor \right) \cap \mathbb{Z}$ to obtain

\tilde{S} , a multiset of $\lceil 21 \cdot \ln\left(\frac{N}{1-\sigma}\right) \rceil s_j$ values. With probability at least σ more than half (with multiplicity) of the entries of $\mathcal{M}_{\tilde{S}, \omega} \cdot \vec{|f|}$ will estimate the ω^{th} entry of $\vec{|f|}$ to within $(\epsilon/k) \cdot \left\| \vec{f} - \vec{f}_{(k/\epsilon)}^{\text{opt}} \right\|_1$ precision for all $\omega \in S$.

Given the last paragraph, it is not difficult to see that with probability at least σ a result analogous to that of Lemma 5 will hold for $\mathcal{R}_{\lambda, \tilde{S}} \cdot \vec{f}$. That is, with probability at least σ the following will hold for all $\omega \in S$: The majority (when counted with multiplicity) of $\mathcal{M}_{\tilde{S}, \omega}$ rows, $\vec{r} \in \{0, 1\}^N$, will have $(\vec{r} \otimes \vec{s}) \cdot \vec{f} \approx \hat{f}_\omega$ for a given row, \vec{s} , of $\mathcal{N}_{\lambda, s_1}$ if and only if \vec{s} is also a row of $\mathcal{N}_{\lambda, s_1, \omega}$. Furthermore, $\mathcal{R}_{\lambda, \tilde{S}} \cdot \vec{f}$ can still be approximately computed using Algorithm 1 if only the unique s_j values in \tilde{S} are given as relatively prime s_j -inputs. In this case a result analogous to Lemma 3 will also still hold since we will merely be computing a subset of the previously calculated vector entries. Finally, by inspecting the proof of Lemma 6 we can see that an almost identical result (with K replaced by the l value from Corollary 2) will hold any time $\mathcal{R}_{\lambda, \tilde{S}} \cdot \vec{f}$ satisfies the aforementioned variants of both Lemmas 5 and 3.

Taken all together, we can see that with probability at least σ both of the following statements will be true: First, all at most $N x_{\omega_{jh}}$ values ever produced by lines 16 and 17 of Algorithm 3 will have

$$|x_{\omega_{jh}} - \hat{f}_{\omega_{jh}}| \leq \sqrt{2} \cdot \left(\frac{\epsilon \cdot \left\| \vec{f} - \vec{f}_{(k/\epsilon)}^{\text{opt}} \right\|_1}{k} + \left\| \hat{f} - \vec{f} \right\|_1 \right).$$

Second, a variant of Lemma 6 will ensure that all $\omega \in S$ with

$$|\hat{f}_\omega| > 4 \cdot \left(\frac{\epsilon \cdot \left\| \vec{f} - \vec{f}_{(k/\epsilon)}^{\text{opt}} \right\|_1}{k} + \left\| \hat{f} - \vec{f} \right\|_1 \right)$$

are reconstructed by lines 6 through 14 of Algorithm 3 more than $\frac{\lceil 21 \cdot \ln\left(\frac{N}{1-\sigma}\right) \rceil}{2}$ times. The Equation 30 error bound now follows from the proof of Theorem 7.

We upper bound the number of required function evaluations by bounding the number of rows for a particular $\mathcal{R}_{\lambda, \tilde{S}}$ matrix constructed with $\lceil 21 \cdot \ln\left(\frac{N}{1-\sigma}\right) \rceil$ randomly chosen s_j values defined as in Equations 9 – 11 (with $K = 14 \cdot (k/\epsilon) \lceil \log_{s_1} N \rceil + 1$). In this case Equation 35 together with results from [10] tell us that s_K is itself bounded above by Equation 41. The final upper bound on the number of point evaluations of f then follows from an application of Lemma 4. Note that the product of the Lemma 4 row bound with $\lceil 21 \cdot \ln\left(\frac{N}{1-\sigma}\right) \rceil$ and Equation 41 provides a concrete upper bound for the number of point evaluations of f .

We will begin bounding the runtime of Algorithm 3 by bounding the runtime of lines 15 through 21. Line 12 of Algorithm 3 will be executed a total of $O\left(\frac{k \cdot \log\left(\frac{N}{1-\sigma}\right) \cdot \log_{(k/\epsilon)} N \cdot \log\left(\frac{k \log N}{\epsilon}\right)}{\epsilon}\right)$ times (see Equation 41). Therefore, lines 16 and 17 will be executed $O\left(\frac{k \cdot \log_{(k/\epsilon)} N \cdot \log\left(\frac{k \ln N}{\epsilon}\right)}{\epsilon}\right)$ times apiece. Each such median operation can be accomplished in $O\left(\log\left(\frac{N}{1-\sigma}\right) \cdot \lambda\right)$ time using a median-of-medians algorithm (e.g., see [9]). Therefore, the total runtime of lines 15 through 21 will be $O\left(\frac{k \cdot \log\left(\frac{N}{1-\sigma}\right) \cdot \log_{(k/\epsilon)} N \cdot \log\left(\frac{k \log N}{\epsilon}\right) \cdot \log(\epsilon N/k)}{\epsilon \cdot \log \log(\epsilon N/k)}\right)$. Turning our attention to lines 6 through 14, we note that their runtime will be dominated by the $O\left(\frac{k \cdot \log\left(\frac{N}{1-\sigma}\right) \cdot \log_{(k/\epsilon)} N \cdot \log\left(\frac{k \log N}{\epsilon}\right) \cdot \log(\epsilon N/k)}{\epsilon \cdot \log \log(\epsilon N/k)}\right)$ executions of line 9. Therefore, the total runtime of lines 6 through 14 will be $O\left(\frac{k \cdot \log\left(\frac{N}{1-\sigma}\right) \cdot \log_{(k/\epsilon)} N \cdot \log\left(\frac{k \log N}{\epsilon}\right) \cdot \log^2(\epsilon N/k)}{\epsilon \cdot \log \log(\epsilon N/k)}\right)$ (see Lemma 4). The stated overall runtime of lines 6 through 21 follows.

APPENDIX H. PROOF OF THEOREM 8

Suppose we want to resolve at least M frequencies in each of D dimensions (i.e., we want to approximate the D -dimensional array $\vec{f} \in \mathbb{C}^{M^D}$). We begin by choosing the smallest $\tilde{D} \in \mathbb{N}$ such that

$$\prod_{j=1}^{\tilde{D}-D+1} p_j > (MD)^D.$$

The first paragraph of Appendix D reveals that

$$(45) \quad \tilde{D} < 3 \cdot \frac{D \cdot \ln(MD)}{\ln(D \cdot \ln(MD))} + D = \frac{D \cdot \ln(MD)}{\ln(D \cdot \ln(MD))} \cdot 3 \left(1 + \frac{\ln D + \ln \ln(MD)}{\ln(MD)} \right) = O\left(\frac{D \cdot \log(MD)}{\log(D \cdot \log(MD))} \right).$$

Furthermore, we can see from [10] that

$$(46) \quad \ln \left(\prod_{j=1}^{\tilde{D}} p_j \right) \leq D \cdot \ln(MD) + \sum_{j=\tilde{D}-D+1}^{\tilde{D}} \ln p_j \leq D \cdot (\ln(MD) + \ln p_{\tilde{D}}) \leq D \cdot (\ln(MD) + \ln(\tilde{D} \cdot (\ln \tilde{D} + \ln \ln \tilde{D})))$$

for $D \geq 2$ and $(MD)^D \geq 2, 310$. Thus, $\log(\prod_{j=1}^{\tilde{D}} p_j)$ is generally $O(D \cdot \log(M \cdot D))$. We will use these first \tilde{D} primes to help define our new one-dimensional function f_{new} (see Equation 32 above).

Set $\tilde{D}_0 = 0$ and recursively define \tilde{D}_d to be such that

$$\prod_{j=\tilde{D}_{d-1}+1}^{\tilde{D}_d-1} p_j \leq MD < \prod_{j=\tilde{D}_{d-1}+1}^{\tilde{D}_d} p_j$$

for all $1 \leq d \leq D$. We then define the D pairwise relatively prime values required for approximating each Fourier coefficient, $\vec{f}(g^{-1}(\omega)) \in \mathbb{C}$, via Equation 33 to be

$$P_d = \prod_{j=\tilde{D}_{d-1}+1}^{\tilde{D}_d} p_j$$

for $1 \leq d \leq D$. Set $\tilde{N} = \prod_{d=1}^D P_d \leq \prod_{j=1}^{\tilde{D}} p_j$. The stated runtime and sampling bounds follow.

We are now in the position to apply any of Theorem 6, Corollary 3, Theorem 7, or Corollary 4 to approximate $\vec{f}_{\text{new}} \in \mathbb{C}^{\tilde{N}}$. Upon applying any of these four results to f_{new} we will obtain (either deterministically, or randomly with high probability) a $2k$ -sparse $\vec{x}_S \in \mathbb{C}^{\tilde{N}}$ satisfying

$$\left\| \vec{f}_{\text{new}} - \vec{x}_S \right\|_2 \leq \left\| \vec{f}_{\text{new}} - \vec{f}_{\text{new } k}^{\text{opt}} \right\|_2 + \frac{22\epsilon \cdot \left\| \vec{f}_{\text{new}} - \vec{f}_{\text{new } (k/\epsilon)}^{\text{opt}} \right\|_1}{\sqrt{k}} + 22\sqrt{k} \cdot \left\| \hat{f}_{\text{new}} - \tilde{f}_{\text{new}} \right\|_1.$$

Recall that we have only guaranteed that \hat{f} 's Fourier coefficients for $([-M/2, M/2] \cap \mathbb{Z})^D$ map into \hat{f}_{new} 's Fourier coefficients for $[-\tilde{N}/2, \tilde{N}/2] \cap \mathbb{Z}$ (although many others will as well). Thus, for simplicity, we assumed that f is bandlimited when translating these error bounds back into terms of \hat{f} . Given this assumption we have $\left\| \hat{f}_{\text{new}} - \tilde{f}_{\text{new}} \right\|_1 = 0$. The stated error bound now follows from the fact that g is a bijection.