

A NEW CLASS OF FULLY DISCRETE SPARSE FOURIER TRANSFORMS: FASTER STABLE IMPLEMENTATIONS WITH GUARANTEES

SAMI MERHI, RUOCHUAN ZHANG, MARK A. IWEN, ANDREW CHRISTLIEB

ABSTRACT. In this paper we consider Sparse Fourier Transform (SFT) algorithms for approximately computing the best s -term approximation of the Discrete Fourier Transform (DFT) $\hat{\mathbf{f}} \in \mathbb{C}^N$ of any given input vector $\mathbf{f} \in \mathbb{C}^N$ in just $(s \log N)^{\mathcal{O}(1)}$ -time using only a similarly small number of entries of \mathbf{f} . In particular, we present a deterministic SFT algorithm which is guaranteed to always recover a near best s -term approximation of the DFT of any given input vector $\mathbf{f} \in \mathbb{C}^N$ in $\mathcal{O}\left(s^2 \log^{\frac{11}{2}}(N)\right)$ -time. Unlike previous deterministic results of this kind, our deterministic result holds for both arbitrary vectors $\mathbf{f} \in \mathbb{C}^N$ and vector lengths N . In addition to these deterministic SFT results, we also develop several new publicly available randomized SFT implementations for approximately computing $\hat{\mathbf{f}}$ from \mathbf{f} using the same general techniques. The best of these new implementations is shown to outperform existing discrete sparse Fourier transform methods with respect to both runtime and noise robustness for large vector lengths N .

1. INTRODUCTION

Herein we are concerned with the rapid approximation of the discrete Fourier transform $\hat{\mathbf{f}} \in \mathbb{C}^N$ of a given vector $\mathbf{f} \in \mathbb{C}^N$ for large values of N . Though standard Fast Fourier Transform (FFT) algorithms [7, 4, 24] can accomplish this task in $\mathcal{O}(N \log N)$ -time for arbitrary $N \in \mathbb{N}$, this runtime complexity may still be unnecessarily computationally taxing when N is extremely large. This is particularly true when the vector $\hat{\mathbf{f}}$ is approximately s -sparse (i.e., contains only $s \ll N$ nonzero entries) as in compressive sensing [8] and certain wideband signal processing applications (see, e.g., [20]). Such applications have therefore motivated the development of Discrete Sparse Fourier Transform (DSFT) techniques [11, 9] which are capable of accurately approximating s -sparse DFT vectors $\hat{\mathbf{f}} \in \mathbb{C}^N$ in just $s \cdot \log^{\mathcal{O}(1)} N$ -time. When $s \ll N$ these methods are significantly faster than standard $\mathcal{O}(N \log N)$ -time FFT methods, effectively achieving sublinear $o(N)$ runtime complexities in such cases.

Currently, the most widely used $s \cdot \log^{\mathcal{O}(1)} N$ -time DSFT methods [10, 14, 12] are randomized algorithms which accurately compute $\hat{\mathbf{f}}$ with high probability when given sampling access to \mathbf{f} . Many existing sparse Fourier transforms which are entirely deterministic [17, 19, 25, 21, 5], on the other hand, are perhaps best described as Unequally Spaced Sparse Fourier Transform (USSFT) methods in that they approximately compute $\hat{\mathbf{f}}$, with its entries \hat{f}_ω indexed by the set $B := \left(-\left\lceil \frac{N}{2} \right\rceil, \left\lfloor \frac{N}{2} \right\rfloor\right) \cap \mathbb{Z}$, by sampling its associated trigonometric polynomial

$$f(x) = \sum_{\omega \in B} \hat{f}_\omega e^{i\omega x}$$

Sami Merhi: Department of Mathematics, Michigan State University, East Lansing, MI, 48824, USA (merhisam@math.msu.edu).

Ruochuan Zhang: Department of Mathematics, Michigan State University, East Lansing, MI, 48824, USA (zhangr12@msu.edu).

Mark A. Iwen: Department of Mathematics, and Department of Computational Mathematics, Science, and Engineering (CMSE), Michigan State University, East Lansing, MI, 48824, USA (markiwen@math.msu.edu).

Andrew Christlieb: Department of Computational Mathematics, Science, and Engineering (CMSE), Michigan State University, East Lansing, MI, 48824, USA (andrewc@msu.edu).

at a collection of $m \ll N$ specially constructed unequally spaced points $x_1, \dots, x_m \in [-\pi, \pi]$. These methods have no probability of failing to recover s -sparse $\hat{\mathbf{f}}$, but can not accurately compute the DFT $\hat{\mathbf{f}}$ of an arbitrary given vector $\mathbf{f} \in \mathbb{C}^N$ due to their need for *unequally spaced* function evaluations from f of the form $\{f(x_k)\}_{k=1}^m$.¹

This state of affairs has left a gap in the theory of DSFT methods. Existing deterministic sparse Fourier transform algorithms currently can efficiently compute the s -sparse DFT $\hat{\mathbf{f}}$ of a given vector $\mathbf{f} \in \mathbb{C}^N$ only if either (i) N is a power of a small prime [22], or else (ii) $\hat{f}_\omega = 0$ for all $\omega \in B$ with $|\omega| > N/4$ [17, 18]. In this paper this gap is filled by the development of a new entirely deterministic DSFT algorithm which is always guaranteed to accurately approximate any (nearly) s -sparse $\hat{\mathbf{f}} \in \mathbb{C}^N$ of any size N when given access only to $\mathbf{f} \in \mathbb{C}^N$. In addition, the method used to develop this new deterministic DSFT algorithm is general enough that it can be applied to *any* fast and noise robust USSFT method of the type mentioned above (be it deterministic, or randomized) in order to yield a new fast and robust DSFT algorithm. As a result, we are also able to use the fastest of the currently existing USSFT methods [15, 17, 19, 25, 21, 5, 3] in order to create new publicly available DSFT implementations herein which are both faster and more robust to noise than currently existing noise robust DSFT methods for large N .

More generally, we emphasize that the techniques utilized below free developers of SFT methods to develop more general USSFT methods which utilize samples from the trigonometric polynomial f above at any points $\{x_k\}_{k=1}^m \subset [-\pi, \pi]$ they like when attempting to create better DSFT algorithms in the future. Indeed, the techniques herein provide a relatively simple means of translating any future fast and robust USSFT algorithms into (still fast) DSFT algorithms.

1.1. Theoretical Results. Herein we focus on rapidly producing near best s -term approximations of $\hat{\mathbf{f}}$ of the type usually considered in compressive sensing [6]. Let $\hat{\mathbf{f}}_s^{\text{opt}} \in \mathbb{C}^N$ denote an optimal s -term approximation to $\hat{\mathbf{f}} \in \mathbb{C}^N$. That is, let $\hat{\mathbf{f}}_s^{\text{opt}}$ preserve s of the largest magnitude entries of $\hat{\mathbf{f}}$ while setting the rest of its $N - s$ smallest magnitude entries to 0.² The following DSFT theorem is proven below.³

Theorem 1. *Let $N \in \mathbb{N}$, $s \in [2, N] \cap \mathbb{N}$, $1 \leq r \leq \frac{N}{36}$, and $\mathbf{f} \in \mathbb{C}^N$. There exists an algorithm that will always deterministically return an s -sparse vector $\mathbf{v} \in \mathbb{C}^N$ satisfying*

$$(1) \quad \left\| \hat{\mathbf{f}} - \mathbf{v} \right\|_2 \leq \left\| \hat{\mathbf{f}} - \hat{\mathbf{f}}_s^{\text{opt}} \right\|_2 + \frac{33}{\sqrt{s}} \cdot \left\| \hat{\mathbf{f}} - \hat{\mathbf{f}}_s^{\text{opt}} \right\|_1 + 198\sqrt{s} \|\mathbf{f}\|_\infty N^{-r}$$

in just $\mathcal{O}\left(\frac{s^2 \cdot r^{\frac{3}{2}} \cdot \log^{\frac{11}{2}}(N)}{\log(s)}\right)$ -time when given access to \mathbf{f} . If returning an s -sparse vector $\mathbf{v} \in \mathbb{C}^N$ that satisfies (1) for each \mathbf{f} with probability at least $(1 - p) \in [2/3, 1)$ is sufficient, a Monte Carlo algorithm also exists which will do so in just $\mathcal{O}\left(s \cdot r^{\frac{3}{2}} \cdot \log^{\frac{9}{2}}(N) \cdot \log\left(\frac{N}{p}\right)\right)$ -time.

Note the quadratic-in- s runtime dependence of the deterministic algorithm mentioned by Theorem 1. It turns out that there is a close relationship between the sampling points $\{x_k\}_{k=1}^m$ used by the deterministic USSFT methods [19] employed as part of the proof of Theorem 1 and the construction of explicit (deterministic) RIP matrices (see [16, 1] for details). As a result, reducing the quadratic dependence on s of the $s^2 \log^{\mathcal{O}(1)} N$ -runtime complexity of the deterministic DSFT algorithms referred to by Theorem 1 while still satisfying the error guarantee (1) is likely at least as difficult as constructing explicit deterministic RIP matrices with fewer than $s^2 \log^{\mathcal{O}(1)} N$ rows

¹Note that methods which compute the DFT $\hat{\mathbf{f}}$ of a given vector \mathbf{f} implicitly assume that \mathbf{f} contains *equally spaced* samples from the trigonometric polynomial f above.

²Note that $\hat{\mathbf{f}}_s^{\text{opt}}$ may not be unique as there can be ties for the s^{th} largest entry in magnitude of \mathbf{f} . This trivial ambiguity turns out not to matter.

³Theorem 1 is a slightly simplified version of Theorem 5 proven in §4.

by subsampling rows from an $N \times N$ DFT matrix. Unfortunately, explicitly constructing RIP matrices of this type is known to be a very difficult problem [8]. This means that constructing an entirely deterministic DSFT algorithm which is both guaranteed to always satisfy (1), and which also always runs in $s \log^{\mathcal{O}(1)} N$ -time, is also likely to be extremely difficult to achieve at present.⁴

The remainder of this paper is organized as follows: In section 2 we set up notation and establish necessary background results. Then, in section 3, we describe our method for converting noise robust USSFT methods into DSFT methods. The resulting approach is summarized in Algorithm 1 therein. Next, Theorem 1 is proven in section 4 using the intermediary results of sections 2 and 3. An empirical evaluation of several new DSFT algorithms resulting from our proposed approach is then performed in section 5. The paper is finally concluded with a few additional comments in §6.

2. NOTATION AND SETUP

The Fourier series representation of a 2π -periodic function $f : [-\pi, \pi] \rightarrow \mathbb{C}$ will be denoted by

$$f(x) = \sum_{\omega \in \mathbb{Z}} \widehat{f}_\omega e^{i\omega x}$$

with its Fourier coefficients given by

$$\widehat{f}_\omega = \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) e^{-i\omega x} dx.$$

We let $\widehat{f} := \{\widehat{f}_\omega\}_{\omega \in \mathbb{Z}}$ represent the infinite sequence of all Fourier coefficients of f below. Given two 2π -periodic functions f and g we define the convolution of f and g at $x \in \mathbb{R}$ to be

$$(f * g)(x) = (g * f)(x) := \frac{1}{2\pi} \int_{-\pi}^{\pi} g(x-y) f(y) dy.$$

This definition, coupled with the definition of the Fourier transform, yields the well-known equality

$$\widehat{f * g}_\omega = \widehat{f}_\omega \widehat{g}_\omega \quad \forall \omega \in \mathbb{Z}.$$

We may also write $\widehat{f * g} = \widehat{f} \circ \widehat{g}$ where \circ denotes the Hadamard product.

For any $N \in \mathbb{N}$, define the Discrete Fourier Transform (DFT) matrix $F \in \mathbb{C}^{N \times N}$ by

$$F_{\omega, j} := \frac{(-1)^\omega}{N} e^{-\frac{2\pi i \cdot \omega \cdot j}{N}},$$

and let $B := (-\lceil \frac{N}{2} \rceil, \lfloor \frac{N}{2} \rfloor] \cap \mathbb{Z}$ be a set of N integer frequencies centered at 0. Furthermore, let $\mathbf{f} \in \mathbb{C}^N$ denote the vector of equally spaced samples from f whose entries are given by

$$f_j := f\left(-\pi + \frac{2\pi j}{N}\right)$$

for $j = 0, \dots, N-1$. One can now see that if

$$f(x) = \sum_{\omega \in B} \widehat{f}_\omega e^{i\omega x},$$

then

$$(2) \quad F\mathbf{f} =: \widehat{\mathbf{f}}$$

⁴Of course deterministic algorithms with error guarantees of the type of (1) do exist for more restricted classes of periodic functions f . See, e.g., [2, 3, 23] for some examples. These include USSFT methods developed for periodic functions with structured Fourier support [2] which are of use for, among other things, the fast approximation of functions which exhibit sparsity with respect to other bounded orthonormal basis functions [13].

where $\hat{\mathbf{f}} \in \mathbb{C}^N$ denotes the subset of \hat{f} with indices in B , and in vector form. More generally, bolded lower case letters will always represent vectors in \mathbb{C}^N below.

As mentioned above, $\hat{f} := \{\hat{f}_\omega\}_{\omega \in \mathbb{Z}}$ is the infinite sequence of all Fourier coefficients of f . For any subset $S \subseteq \mathbb{Z}$ we let $\hat{f}|_S \in \mathbb{C}^{\mathbb{Z}}$ be the sequence \hat{f} restricted to the subset S , so that $\hat{f}|_S$ has terms $(\hat{f}|_S)_\omega = \hat{f}_\omega$ for all $\omega \in S$, and $(\hat{f}|_S)_\omega = 0$ for all $\omega \in S^c := \mathbb{Z} \setminus S$. Note that \hat{f} above is exactly $\hat{f}|_B$ excluding its zero terms for all $\omega \notin B$. Thus, given any subset $S \subseteq B$, we let $\hat{\mathbf{f}}|_S \in \mathbb{C}^N$ be the vector $\hat{\mathbf{f}}$ restricted to the set S in an analogous fashion. That is, for $S \subseteq B$ we will have $(\hat{\mathbf{f}}|_S)_\omega = \hat{\mathbf{f}}_\omega$ for all $\omega \in S$, and $(\hat{\mathbf{f}}|_S)_\omega = 0$ for all $\omega \in B \setminus S$.

Given the sequence $\hat{f} \in \mathbb{C}^{\mathbb{Z}}$ and $s \leq N$, we denote by $R_s^{\text{opt}}(\hat{f})$ a subset of B containing s of the most energetic frequencies of f ; that is

$$R_s^{\text{opt}}(\hat{f}) := \{\omega_1, \dots, \omega_s\} \subseteq B \subset \mathbb{Z}$$

where the frequencies $\omega_j \in B$ are ordered such that

$$|\hat{f}_{\omega_1}| \geq |\hat{f}_{\omega_2}| \geq \dots \geq |\hat{f}_{\omega_s}| \geq \dots \geq |\hat{f}_{\omega_N}|.$$

Here, if desired, one may break ties by also requiring, e.g., that $\omega_j < \omega_k$ for all $j < k$ with $|\hat{f}_{\omega_j}| = |\hat{f}_{\omega_k}|$. We will then define $f_s^{\text{opt}} : [-\pi, \pi] \rightarrow \mathbb{C}$ based on $R_s^{\text{opt}}(\hat{f})$ by

$$f_s^{\text{opt}}(x) := \sum_{\omega \in R_s^{\text{opt}}(\hat{f})} \hat{f}_\omega e^{i\omega x}.$$

Any such 2π -periodic function f_s^{opt} will be referred to as an optimal s -term approximation to f . Similarly, we also define both $\hat{f}_s^{\text{opt}} \in \mathbb{C}^{\mathbb{Z}}$ and $\hat{\mathbf{f}}_s^{\text{opt}} \in \mathbb{C}^N$ to be $\hat{f}|_{R_s^{\text{opt}}(\hat{f})}$ and $\hat{\mathbf{f}}|_{R_s^{\text{opt}}(\hat{f})}$, respectively.

2.1. Periodized Gaussians. In the sections that follow the 2π -periodic Gaussian $g : [-\pi, \pi] \rightarrow \mathbb{R}^+$ defined by

$$(3) \quad g(x) = \frac{1}{c_1} \sum_{n=-\infty}^{\infty} e^{-\frac{(x-2n\pi)^2}{2c_1^2}}$$

with $c_1 \in \mathbb{R}^+$ will play a special role. The following lemmas recall several useful facts concerning both its decay, and its Fourier series coefficients.

Lemma 1. *The 2π -periodic Gaussian $g : [-\pi, \pi] \rightarrow \mathbb{R}^+$ has*

$$g(x) \leq \left(\frac{3}{c_1} + \frac{1}{\sqrt{2\pi}} \right) e^{-\frac{x^2}{2c_1^2}}$$

for all $x \in [-\pi, \pi]$.

Lemma 2. *The 2π -periodic Gaussian $g : [-\pi, \pi] \rightarrow \mathbb{R}^+$ has*

$$\hat{g}_\omega = \frac{1}{\sqrt{2\pi}} e^{-\frac{c_1^2 \omega^2}{2}}$$

for all $\omega \in \mathbb{Z}$. Thus, $\hat{g} = \{\hat{g}_\omega\}_{\omega \in \mathbb{Z}} \in \ell^2$ decreases monotonically as $|\omega|$ increases, and also has $\|\hat{g}\|_\infty = \frac{1}{\sqrt{2\pi}}$.

Lemma 3. Choose any $\tau \in \left(0, \frac{1}{\sqrt{2\pi}}\right)$, $\alpha \in \left[1, \frac{N}{\sqrt{\ln N}}\right]$, and $\beta \in \left(0, \alpha \sqrt{\frac{\ln(1/\tau\sqrt{2\pi})}{2}}\right]$. Let $c_1 = \frac{\beta\sqrt{\ln N}}{N}$ in the definition of the periodic Gaussian g from (3). Then $\hat{g}_\omega \in \left[\tau, \frac{1}{\sqrt{2\pi}}\right]$ for all $\omega \in \mathbb{Z}$ with $|\omega| \leq \left\lceil \frac{N}{\alpha\sqrt{\ln N}} \right\rceil$.

The proofs of Lemmas 1, 2, and 3 are included in Appendix A for the sake of completeness. Intuitively, we will utilize the periodic function g from (3) as a bandpass filter below. Looking at Lemma 3 in this context we can see that its parameter τ will control the effect of \hat{g} on the frequency passband defined by its parameter α . Deciding on the two parameters τ, α then constrains β which, in turn, fixes the periodic Gaussian g by determining its constant coefficient c_1 . As we shall see, the parameter β will also determine the speed and accuracy with which we can approximately sample (i.e., evaluate) the function $f * g$. For this reason it will become important to properly balance these parameters against one another in subsequent sections.

2.2. On the Robustness of the SFTs proposed in [19]. The sparse Fourier transforms presented in [19] include both deterministic and randomized methods for approximately computing the Fourier series coefficients of a given 2π -periodic function f from its evaluations at m -points $\{x_k\}_{k=1}^m \subset [-\pi, \pi]$. The following results describe how accurate these algorithms will be when they are only given approximate evaluations of f at these points instead. These results are necessary because we will want to execute the SFTs developed in [19] on convolutions of the form $f * g$ below, but will only be able to approximately compute their values at each of the required points $x_1, \dots, x_m \in [-\pi, \pi]$.

Lemma 4. Let $s, \epsilon^{-1} \in \mathbb{N} \setminus \{1\}$ with $(s/\epsilon) \geq 2$, and $\mathbf{n} \in \mathbb{C}^m$ be an arbitrary noise vector. There exists a set of m points $\{x_k\}_{k=1}^m \subset [-\pi, \pi]$ such that Algorithm 3 on page 72 of [19], when given access to the corrupted samples $\{f(x_k) + n_k\}_{k=1}^m$, will identify a subset $S \subseteq B$ which is guaranteed to contain all $\omega \in B$ with

$$(4) \quad \left| \hat{f}_\omega \right| > 4 \left(\frac{\epsilon \cdot \left\| \hat{\mathbf{f}} - \hat{\mathbf{f}}_{(s/\epsilon)}^{\text{opt}} \right\|_1}{s} + \left\| \hat{f} - \hat{f}|_B \right\|_1 + \|\mathbf{n}\|_\infty \right).$$

Furthermore, every $\omega \in S$ returned by Algorithm 3 will also have an associated Fourier series coefficient estimate $z_\omega \in \mathbb{C}$ which is guaranteed to have

$$(5) \quad \left| \hat{f}_\omega - z_\omega \right| \leq \sqrt{2} \left(\frac{\epsilon \cdot \left\| \hat{\mathbf{f}} - \hat{\mathbf{f}}_{(s/\epsilon)}^{\text{opt}} \right\|_1}{s} + \left\| \hat{f} - \hat{f}|_B \right\|_1 + \|\mathbf{n}\|_\infty \right).$$

Both the number of required samples, m , and Algorithm 3's operation count are

$$(6) \quad \mathcal{O} \left(\frac{s^2 \cdot \log^4(N)}{\log\left(\frac{s}{\epsilon}\right) \cdot \epsilon^2} \right).$$

If succeeding with probability $(1 - \delta) \in [2/3, 1)$ is sufficient, and $(s/\epsilon) \geq 2$, the Monte Carlo variant of Algorithm 3 referred to by Corollary 4 on page 74 of [19] may be used. This Monte Carlo variant reads only a randomly chosen subset of the noisy samples utilized by the deterministic algorithm,

$$\{f(\tilde{x}_k) + \tilde{n}_k\}_{k=1}^{\tilde{m}} \subseteq \{f(x_k) + n_k\}_{k=1}^m,$$

yet it still outputs a subset $S \subseteq B$ which is guaranteed to simultaneously satisfy both of the following properties with probability at least $1 - \delta$:

- (i) S will contain all $\omega \in B$ satisfying (4), and

(ii) all $\omega \in S$ will have an associated coefficient estimate $z_\omega \in \mathbb{C}$ satisfying (5).

Finally, both this Monte Carlo variant's number of required samples, \tilde{m} , as well as its operation count will also always be

$$(7) \quad \mathcal{O}\left(\frac{s}{\epsilon} \cdot \log^3(N) \cdot \log\left(\frac{N}{\delta}\right)\right).$$

Using the preceding lemma one can easily prove the following noise robust variant of Theorem 7 (and Corollary 4) from §5 of [19]. The proofs of both results are outlined in Appendix B for the sake of completeness.

Theorem 2. *Suppose $f : [-\pi, \pi] \rightarrow \mathbb{C}$ has $\hat{f} \in \ell^1 \cap \ell^2$. Let $s, \epsilon^{-1} \in \mathbb{N} \setminus \{1\}$ with $(s/\epsilon) \geq 2$, and $\mathbf{n} \in \mathbb{C}^m$ be an arbitrary noise vector. Then, there exists a set of m points $\{x_k\}_{k=1}^m \subset [-\pi, \pi]$ together with a simple deterministic algorithm $\mathcal{A} : \mathbb{C}^m \rightarrow \mathbb{C}^{4s}$ such that $\mathcal{A}(\{f(x_k) + n_k\}_{k=1}^m)$ is always guaranteed to output (the nonzero coefficients of) a degree $\leq N/2$ trigonometric polynomial $y_s : [-\pi, \pi] \rightarrow \mathbb{C}$ satisfying*

$$(8) \quad \|f - y_s\|_2 \leq \|\hat{\mathbf{f}} - \hat{\mathbf{f}}_s^{\text{opt}}\|_2 + \frac{22\epsilon \cdot \|\hat{\mathbf{f}} - \hat{\mathbf{f}}_{(s/\epsilon)}^{\text{opt}}\|_1}{\sqrt{s}} + 22\sqrt{s} \left(\|\hat{f} - \hat{f}|_B\|_1 + \|\mathbf{n}\|_\infty \right).$$

Both the number of required samples, m , and the algorithm's operation count are always

$$(9) \quad \mathcal{O}\left(\frac{s^2 \cdot \log^4(N)}{\log\left(\frac{s}{\epsilon}\right) \cdot \epsilon^2}\right).$$

If succeeding with probability $(1 - \delta) \in [2/3, 1)$ is sufficient, and $(s/\epsilon) \geq 2$, a Monte Carlo variant of the deterministic algorithm may be used. This Monte Carlo variant reads only a randomly chosen subset of the noisy samples utilized by the deterministic algorithm,

$$\{f(\tilde{x}_k) + \tilde{n}_k\}_{k=1}^{\tilde{m}} \subseteq \{f(x_k) + n_k\}_{k=1}^m,$$

yet it still outputs (the nonzero coefficients of) a degree $\leq N/2$ trigonometric polynomial, $y_s : [-\pi, \pi] \rightarrow \mathbb{C}$, that satisfies (8) with probability at least $1 - \delta$. Both its number of required samples, \tilde{m} , as well as its operation count will always be

$$(10) \quad \mathcal{O}\left(\frac{s}{\epsilon} \cdot \log^3(N) \cdot \log\left(\frac{N}{\delta}\right)\right).$$

We now have the necessary prerequisites in order to discuss our general strategy for constructing several new fully discrete SFTs.

3. DESCRIPTION OF THE PROPOSED APPROACH

In this section we assume that we have access to an SFT algorithm \mathcal{A} which requires m function evaluations of a 2π -periodic function $f : [-\pi, \pi] \rightarrow \mathbb{C}$ in order to produce an s -sparse approximation to \hat{f} . For any non-adaptive SFT algorithm \mathcal{A} the m points $\{x_k\}_{k=1}^m \subset [-\pi, \pi]$ at which \mathcal{A} needs to evaluate f can be determined before \mathcal{A} is actually executed. As a result, the function evaluations $\{f(x_k)\}_{k=1}^m$ required by \mathcal{A} can also be evaluated before \mathcal{A} is ever run. Indeed, if the SFT algorithm \mathcal{A} is both nonadaptive *and* robust to noise it suffices to *approximate* the function evaluations $\{f(x_k)\}_{k=1}^m$ required by \mathcal{A} before it is executed.⁵ These simple ideas form the basis for the proposed computational approach outlined in Algorithm 1.

⁵We hasten to point out, moreover, that similar ideas can also be employed for *adaptive* and noise robust SFT algorithms in order to approximately evaluate f in an “on demand” fashion as well. We leave the details to the interested reader.

Algorithm 1: A Generic Method for Discretizing a Given SFT Algorithm \mathcal{A}

Input : Pointer to vector $\mathbf{f} \in \mathbb{C}^N$, sparsity $s \leq N$, nodes $\{x_k\}_{k=1}^m \subset [-\pi, \pi]$ at which the given SFT algorithm \mathcal{A} needs function evaluations, and α, β satisfying Lemma 3

Output: \widehat{R}^s , a sparse approximation of $\widehat{\mathbf{f}} \in \mathbb{C}^N$

- 1 Initialize $\widehat{R}, \widehat{R}^s \leftarrow \emptyset$
 - 2 Set $c_1 = \frac{\beta\sqrt{\ln N}}{N}$ in the definition of periodic Gaussian g from (3), and $c_2 = \frac{\alpha\sqrt{\ln N}}{2}$
 - 3 **for** j from 1 to $\lceil c_2 \rceil$ **do**
 - 4 $q = -\lceil \frac{N}{2} \rceil + 1 + (2j - 1) \lceil \frac{N}{\alpha\sqrt{\ln N}} \rceil$
 - 5 Modulate g to be $\tilde{g}_q(x) := e^{-iqx} g(x)$
 - 6 **for each point** $x \in \{x_k\}_{k=1}^m$ **do**
 - 7 | Use \mathbf{f} to approximately compute $(\tilde{g}_q * f)(x)$ as per §3.1
 - 8 **end**
 - 9 Run given SFT algorithm \mathcal{A} using the approximate function evaluations $\{(\tilde{g}_q * f)(x_k)\}_{k=1}^m$ in order to find an s -sparse Fourier approximation, $\widehat{R}_{temp} \subset \mathbb{Z} \times \mathbb{C}$, of $\widehat{\tilde{g}_q * f}$.
 - 10 **for each (frequency, Fourier coefficient) pair** $(\omega, c_\omega) \in \widehat{R}_{temp}$ **do**
 - 11 | **if** $\omega \in \left[q - \lceil \frac{N}{\alpha\sqrt{\ln N}} \rceil, q + \lceil \frac{N}{\alpha\sqrt{\ln N}} \rceil \right) \cap B$ **then**
 - 12 | $\widehat{R} = \widehat{R} \cup \left\{ \left(\omega, c_\omega / \left(\widehat{\tilde{g}_q} \right)_\omega \right) \right\}$
 - 13 | **end**
 - 14 **end**
 - 15 **end**
 - 16 Choose the s frequencies ω with $(\omega, \tilde{c}_\omega) \in \widehat{R}$ having the largest $|\tilde{c}_\omega|$, and put those $(\omega, \tilde{c}_\omega)$ in \widehat{R}^s
 - 17 Return \widehat{R}^s
-

The objective of Algorithm 1 is to use a nonadaptive and noise robust SFT algorithm \mathcal{A} which requires off-grid function evaluations in order to approximately compute the DFT of a given vector $\mathbf{f} \in \mathbb{C}^N$, $\widehat{\mathbf{f}} = F\mathbf{f}$. Note that computing $\widehat{\mathbf{f}}$ is equivalent to computing the Fourier series coefficients of the degree N trigonometric interpolant of \mathbf{f} . Hereafter the 2π -periodic function $f : [-\pi, \pi] \rightarrow \mathbb{C}$ under consideration will always be this degree N trigonometric interpolant of \mathbf{f} . Our objective then becomes to approximately compute \widehat{f} using \mathcal{A} . Unfortunately, our given input vector \mathbf{f} only contains equally spaced function evaluations of f , and so does not actually contain the function evaluations $\{f(x_k)\}_{k=1}^m$ required by \mathcal{A} . As a consequence, we are forced to try to interpolate these required function evaluations $\{f(x_k)\}_{k=1}^m$ from the available equally spaced function evaluations \mathbf{f} .

Directly interpolating the required function evaluations $\{f(x_k)\}_{k=1}^m$ from \mathbf{f} for an arbitrary degree N trigonometric polynomial f using standard techniques appears to be either too inaccurate, or else too slow to work well in our setting.⁶ As a result, Algorithm 1 instead uses \mathbf{f} to rapidly approximate samples from the convolution of the unknown trigonometric polynomial f with (several modulations of) a known filter function g . Thankfully, all of the evaluations $\{(g * f)(x_k)\}_{k=1}^m$ can be approximated very accurately using only the data in \mathbf{f} in just $\mathcal{O}(m \log N)$ -time when g is chosen carefully enough (see §3.1 below). The given SFT algorithm \mathcal{A} is then used to approximate the Fourier coefficients of $g * f$ for each modulation of g using these approximate evaluations. Finally, $\widehat{\mathbf{f}}$ is then approximated using the recovered sparse approximation for each $\widehat{g * f}$ combined with our a priori knowledge of \widehat{g} .

⁶Each function evaluation $f(x_k)$ needs to be accurately computed in just $\mathcal{O}(\log^c N)$ -time in order to allow us to achieve our overall desired runtime for Algorithm 1.

3.1. Rapidly and Accurately Evaluating $f * g$. In this section we will carefully consider the approximation of $(f * g)(x)$ by a *severely truncated version* of the semi-discrete convolution sum

$$(11) \quad \frac{1}{N} \sum_{j=0}^{N-1} f\left(-\pi + \frac{2\pi j}{N}\right) g\left(x + \pi - \frac{2\pi j}{N}\right)$$

for any given value of $x \in [-\pi, \pi]$. Our goal is to determine exactly how many terms of this finite sum we actually need in order to obtain an accurate approximation of $f * g$ at an arbitrary x -value. More specifically, we aim to use as few terms from this sum as absolutely possible in order to ensure, e.g., an approximation error of size $\mathcal{O}(N^{-2})$.

Without loss of generality, let us assume that $N = 2M + 1$ is odd – this allows us to express B , the set of N Fourier modes about zero, as

$$B := \left(-\left\lfloor \frac{N}{2} \right\rfloor, \left\lfloor \frac{N}{2} \right\rfloor\right) \cap \mathbb{Z} = [-M, M] \cap \mathbb{Z}.$$

In the lemmas and theorems below the function $f : [-\pi, \pi] \rightarrow \mathbb{C}$ will always denote a degree- N trigonometric polynomial of the form

$$f(x) = \sum_{\omega \in B} \widehat{f}_\omega e^{i\omega x}.$$

Furthermore, g will always denote the periodic Gaussian as defined above in (3). Finally, we will also make use of the Dirichlet kernel $D_M : \mathbb{R} \rightarrow \mathbb{C}$, defined by

$$D_M(y) = \frac{1}{2\pi} \sum_{n=-M}^M e^{iny} = \frac{1}{2\pi} \sum_{n \in B} e^{iny}.$$

The relationship between trigonometric polynomials such as f and the Dirichlet kernel D_M is the subject of the following lemma.

Lemma 5. *Let $h : [-\pi, \pi] \rightarrow \mathbb{C}$ have $\widehat{h}_\omega = 0$ for all $\omega \notin B$, and define the set of points $\{y_j\}_{j=0}^{2M} = \left\{-\pi + \frac{2\pi j}{N}\right\}_{j=0}^{2M}$. Then,*

$$2\pi (h * D_M)(x) = h(x) = \frac{2\pi}{N} \sum_{j=0}^{2M} h(y_j) D_M(x - y_j)$$

holds for all $x \in [-\pi, \pi]$.

Proof. By the definition of D_M , we trivially have $2\pi \left(\widehat{D_M}\right)_\omega = \chi_B(\omega) \forall \omega \in \mathbb{Z}$. Thus,

$$\widehat{h} = 2\pi \cdot \widehat{h} \circ \widehat{D_M} = 2\pi \cdot \widehat{h * D_M}$$

where, as before, \circ denotes the Hadamard product, and $*$ denotes convolution. This yields $h(x) = 2\pi (h * D_M)(x)$ and so establishes the first equality above. To establish the second equality above, recall from (2) that for any $\omega \in B$ we will have

$$\widehat{h}_\omega = \frac{(-1)^\omega}{N} \sum_{j=0}^{2M} h\left(-\pi + \frac{2\pi j}{N}\right) e^{-\frac{2\pi i j \omega}{N}} = \frac{1}{N} \sum_{j=0}^{2M} h(y_j) e^{-i\omega y_j},$$

since h is a trigonometric polynomial. Thus, given $x \in [-\pi, \pi]$ one has

$$h(x) = \sum_{\omega \in B} \widehat{h}_\omega e^{i\omega x} = \frac{1}{N} \sum_{j=0}^{2M} \left(h(y_j) \sum_{\omega \in B} e^{i\omega(x-y_j)} \right) = \frac{2\pi}{N} \sum_{j=0}^{2M} h(y_j) D_M(x - y_j).$$

We now have the desired result. \square

We can now write a formula for $g * f$ which only depends on N evaluations of f in $[-\pi, \pi]$.

Lemma 6. *Given the set of equally spaced points $\{y_j\}_{j=0}^{2M} = \left\{-\pi + \frac{2\pi j}{N}\right\}_{j=0}^{2M}$ one has that*

$$(g * f)(x) = \frac{1}{N} \sum_{j=0}^{2M} f(y_j) \int_{-\pi}^{\pi} g(x - u - y_j) D_M(u) du$$

for all $x \in [-\pi, \pi]$.

Proof. By Lemma 5, we have

$$\begin{aligned} (g * f)(x) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} g(x - y) f(y) dy = \frac{1}{N} \int_{-\pi}^{\pi} g(x - y) \sum_{j=0}^{2M} f(y_j) D_M(y - y_j) dy \\ &= \frac{1}{N} \sum_{j=0}^{2M} f(y_j) \int_{-\pi}^{\pi} g(x - u - y_j) D_M(u) du. \end{aligned}$$

The last equality holds after a change of variables since g and D_M are both 2π -periodic. \square

The next two lemmas will help us bound the error produced by discretizing the integral weights present in the finite sum provided by Lemma 6 above. More specifically, they will ultimately allow us to approximate the sum in Lemma 6 by the sum in (11).

Lemma 7. *Let $x \in [-\pi, \pi]$ and $y_j = -\pi + \frac{2\pi j}{N}$ for some $j = 0, \dots, 2M$. Then,*

$$\int_{-\pi}^{\pi} g(x - u - y_j) D_M(u) du = \sum_{n \in B} \widehat{g}_n e^{in(x - y_j)}.$$

Proof. Recalling that $2\pi \widehat{D_M}_\omega = \chi_B(\omega)$ for all $\omega \in \mathbb{Z}$ we have that

$$\int_{-\pi}^{\pi} g(x - u - y_j) D_M(u) du = 2\pi (D_M * g)(x - y_j) = \sum_{n \in \mathbb{Z}} \widehat{g}_n \chi_B(n) e^{in(x - y_j)} = \sum_{n \in B} \widehat{g}_n e^{in(x - y_j)}.$$

\square

Lemma 8. *Denote $I(a) := \int_{-a}^a e^{-x^2} dx$ for $a > 0$; then*

$$\pi \left(1 - e^{-a^2}\right) < I^2(a) < \pi \left(1 - e^{-2a^2}\right).$$

Proof. Let $a > 0$ and observe that

$$I^2(a) = \int_{-a}^a \int_{-a}^a e^{-x^2 - y^2} dx dy > \iint_{\{x^2 + y^2 \leq a^2\}} e^{-(x^2 + y^2)} dx dy = \pi \left(1 - e^{-a^2}\right).$$

The first equality holds by Fubini's theorem, and the inequality follows simply by integrating a positive function over a disk of radius a as opposed to a square of sidelength $2a$. A similar argument yields the upper bound. \square

We are now ready to bound the difference between the integral weights present in the finite sum provided by Lemma 6, and the $g(x - y_j)$ -weights present in the sum (11).

Lemma 9. Choose any $\tau \in \left(0, \frac{1}{\sqrt{2\pi}}\right)$, $\alpha \in \left[1, \frac{N}{\sqrt{\ln N}}\right]$, and $\beta \in \left(0, \alpha \sqrt{\frac{\ln(1/\tau\sqrt{2\pi})}{2}}\right]$. Let $c_1 = \frac{\beta\sqrt{\ln N}}{N}$ in the definition of the periodic Gaussian g so that

$$g(x) = \frac{N}{\beta\sqrt{\ln N}} \sum_{n=-\infty}^{\infty} e^{-\frac{(x-2n\pi)^2 N^2}{2\beta^2 \ln N}}.$$

Then for all $x \in [-\pi, \pi]$ and $y_j = -\pi + \frac{2\pi j}{N}$,

$$\left| g(x - y_j) - \int_{-\pi}^{\pi} g(x - u - y_j) D_M(u) du \right| < \frac{N^{1-\frac{\beta^2}{18}}}{\beta\sqrt{\ln N}}.$$

Proof. Using Lemma 7 we calculate

$$\begin{aligned} \left| g(x - y_j) - \int_{-\pi}^{\pi} g(x - u - y_j) D_M(u) du \right| &= \left| g(x - y_j) - \sum_{n \in B} \widehat{g}_n e^{in(x-y_j)} \right| = \left| \sum_{n \in B^c} \widehat{g}_n e^{in(x-y_j)} \right| \\ &\leq \frac{1}{\sqrt{2\pi}} \sum_{|n| > M} e^{-\frac{c_1^2 n^2}{2}} \quad (\text{Using Lemma 2}) \\ &\leq \frac{2}{\sqrt{2\pi}} \int_M^{\infty} e^{-\frac{c_1^2 n^2}{2}} dn \\ &= \sqrt{\frac{2}{\pi}} \int_M^{\infty} e^{-\frac{\beta^2 n^2 \ln N}{2N^2}} dn. \end{aligned}$$

Upon the change of variable $v = \frac{\beta n \sqrt{\ln N}}{\sqrt{2N}}$, we get that

$$\begin{aligned} \left| g(x - y_j) - \int_{-\pi}^{\pi} g(x - u - y_j) D_M(u) du \right| &\leq \sqrt{\frac{2}{\pi}} \frac{\sqrt{2N}}{\beta\sqrt{\ln N}} \int_{\frac{\beta M \sqrt{\ln N}}{\sqrt{2N}}}^{\infty} e^{-v^2} dv \\ &= \frac{2N}{\beta\sqrt{\pi \ln N}} \frac{1}{2} \left(\int_{-\infty}^{\infty} e^{-v^2} dv - \int_{-\frac{\beta M \sqrt{\ln N}}{\sqrt{2N}}}^{\frac{\beta M \sqrt{\ln N}}{\sqrt{2N}}} e^{-v^2} dv \right) \\ &< \frac{N}{\beta\sqrt{\pi \ln N}} \left(\sqrt{\pi} - \sqrt{\pi \left(1 - e^{-\frac{\beta^2 M^2 \ln N}{2N^2}}\right)} \right) \\ &= \frac{N}{\beta\sqrt{\ln N}} \left(1 - \sqrt{1 - N^{-\frac{\beta^2 M^2}{2N^2}}} \right) \end{aligned}$$

where the last inequality follows from Lemma 8. Noting now that

$$y \in [0, 1] \implies 1 - \sqrt{1 - y} \leq y,$$

and that $\frac{N}{M} = 2 + \frac{1}{M} \in (2, 3]$ for all $M \in \mathbb{Z}^+$, we can further see that

$$\frac{N}{\beta\sqrt{\ln N}} \left(1 - \sqrt{1 - N^{-\frac{\beta^2 M^2}{2N^2}}} \right) \leq \frac{N}{\beta\sqrt{\ln N}} N^{-\frac{\beta^2 M^2}{2N^2}} \leq \frac{N^{1-\frac{\beta^2}{18}}}{\beta\sqrt{\ln N}}$$

also always holds. \square

With the lemmas above we can now prove that (11) can be used to approximate $(g * f)(x)$ for all $x \in [-\pi, \pi]$ with controllable error.

Theorem 3. Let $p \geq 1$. Using the same values of the parameters from Lemma 9 above, one has

$$\left| (g * f)(x) - \frac{1}{N} \sum_{j=0}^{2M} f(y_j) g(x - y_j) \right| \leq \frac{\|\mathbf{f}\|_p}{\beta \sqrt{\ln N}} N^{1 - \frac{\beta^2}{18} - \frac{1}{p}}$$

for all $x \in [-\pi, \pi]$.

Proof. Using Lemmas 6 and 9 followed by Holder's inequality, we have

$$\begin{aligned} \left| (g * f)(x) - \frac{1}{N} \sum_{j=0}^{2M} f(y_j) g(x - y_j) \right| &= \left| \frac{1}{N} \sum_{j=0}^{2M} f(y_j) \left(g(x - y_j) - \int_{-\pi}^{\pi} g(x - u - y_j) D_M(u) du \right) \right| \\ &\leq \frac{1}{N} \sum_{j=0}^{2M} |f(y_j)| \frac{N^{1 - \frac{\beta^2}{18}}}{\beta \sqrt{\ln N}} \leq \frac{N^{-\frac{\beta^2}{18}}}{\beta \sqrt{\ln N}} \|\mathbf{f}\|_p N^{1 - \frac{1}{p}}. \end{aligned}$$

□

To summarize, Theorem 3 tells us that $(g * f)(x)$ can be approximately computed in $\mathcal{O}(N)$ -time for any $x \in [-\pi, \pi]$ using (11). This linear runtime cost may be reduced significantly, however, if one is willing to accept an additional trade-off between accuracy and the number of terms needed in the sum (11). This trade-off is characterized in the next lemma.

Lemma 10. Let $x \in [-\pi, \pi]$, $p \geq 1$, $\gamma \in \mathbb{R}^+$, and $\kappa := \lceil \gamma \ln N \rceil + 1$. Set $j' := \arg \min_j |x - y_j|$. Using the same values of the other parameters from Lemma 9 above, one has

$$\left| \frac{1}{N} \sum_{j=0}^{2M} f(y_j) g(x - y_j) - \frac{1}{N} \sum_{j=j'-\kappa}^{j'+\kappa} f(y_j) g(x - y_j) \right| \leq 2 \|\mathbf{f}\|_p N^{-\frac{2\pi^2 \gamma^2}{\beta^2}}$$

for all $\beta \geq 4$ and $N \geq \beta^2$.

Proof. Appealing to Lemma 1 and recalling that $c_1 = \frac{\beta \sqrt{\ln N}}{N}$ we can see that

$$g(x) \leq \left(\frac{3N}{\beta \sqrt{\ln N}} + \frac{1}{\sqrt{2\pi}} \right) e^{-\frac{x^2 N^2}{2\beta^2 \ln N}}.$$

Using this fact we have that

$$g(x - y_{j' \pm k}) \leq \left(\frac{3N}{\beta \sqrt{\ln N}} + \frac{1}{\sqrt{2\pi}} \right) e^{-\frac{(x - y_{j' \pm k})^2 N^2}{2\beta^2 \ln N}} \leq \left(\frac{3N}{\beta \sqrt{\ln N}} + \frac{1}{\sqrt{2\pi}} \right) e^{-\frac{(2k-1)^2 \pi^2}{2\beta^2 \ln N}}$$

for all $k \in \mathbb{Z}_N$. As a result, one can now bound

$$\left| \frac{1}{N} \sum_{j=0}^{2M} f(y_j) g(x - y_j) - \frac{1}{N} \sum_{j=j'-\kappa}^{j'+\kappa} f(y_j) g(x - y_j) \right|$$

above by

$$(12) \quad \left(\frac{3}{\beta \sqrt{\ln N}} + \frac{1}{N \sqrt{2\pi}} \right) \sum_{k=\kappa+1}^{N-2\kappa-1} (|f(y_{j'-k})| + |f(y_{j'+k})|) e^{-\frac{(2k-1)^2 \pi^2}{2\beta^2 \ln N}},$$

where the y_j -indexes are considered modulo N as appropriate.

Our goal is now to employ Holder's inequality on (12). Toward that end, we will now bound the

q -norm of the vector $\mathbf{h} := \left\{ e^{-\frac{(\kappa+\ell-\frac{1}{2})^2 2\pi^2}{\beta^2 \ln N}} \right\}_{\ell=1}^{N-2\kappa-1}$. Letting $a := q \left(\frac{4}{\beta^2 \ln N} \right)$ we have that

$$\begin{aligned} \|\mathbf{h}\|_q^q &= \sum_{\ell=1}^{N-2\kappa-1} e^{-\frac{\pi^2}{2}(\kappa+\ell-\frac{1}{2})^2 a} < \sum_{\ell=\kappa}^{\infty} e^{-\frac{\pi^2}{2}\ell^2 a} \leq \int_{\kappa-1}^{\infty} e^{-\frac{\pi^2 x^2}{2} a} dx \\ &\leq \sqrt{\frac{1}{2\pi a}} - \frac{1}{\pi\sqrt{2a}} \int_{-\pi(\kappa-1)\sqrt{\frac{a}{2}}}^{\pi(\kappa-1)\sqrt{\frac{a}{2}}} e^{-u^2} du \leq \sqrt{\frac{1}{2\pi a}} e^{-\frac{a\pi^2}{2}(\kappa-1)^2} \leq \frac{\beta}{2} \sqrt{\frac{\ln N}{2\pi q}} N^{-\frac{2q\pi^2\gamma^2}{\beta^2}}, \end{aligned}$$

where we have used Lemma 8 once again. As a result we have that

$$\|\mathbf{h}\|_q \leq \left(\frac{\beta^2 \ln N}{8\pi} \right)^{\frac{1}{2q}} q^{-\frac{1}{2q}} N^{-\frac{2\pi^2\gamma^2}{\beta^2}} \leq \left(\frac{\beta^2 \ln N}{8\pi} \right)^{\frac{1}{2q}} N^{-\frac{2\pi^2\gamma^2}{\beta^2}}$$

for all $q \geq 1$. Applying Holder's inequality on (12) we can now see that (12) is bounded above by

$$2 \left(\frac{3}{\beta\sqrt{\ln N}} + \frac{1}{N\sqrt{2\pi}} \right) \|\mathbf{f}\|_p \left(\frac{\beta^2 \ln N}{8\pi} \right)^{\frac{1}{2} - \frac{1}{2p}} N^{-\frac{2\pi^2\gamma^2}{\beta^2}}.$$

The result now follows. \square

We may now finally combine the truncation and estimation errors in Theorem 3 and Lemma 10 above in order to bound the total error one incurs by approximating $(g * f)(x)$ via a truncated portion of (11) for any given $x \in [-\pi, \pi]$.

Theorem 4. Fix $x \in [-\pi, \pi]$, $p \geq 1$ (or $p = \infty$), $\frac{N}{36} \geq r \geq 1$, and $g : [-\pi, \pi] \rightarrow \mathbb{R}^+$ to be the 2π -periodic Gaussian (3) with $c_1 := \frac{6\sqrt{\ln(N^r)}}{N}$. Set $j' := \arg \min_j |x - y_j|$ where $y_j = -\pi + \frac{2\pi j}{N}$ for all $j = 0, \dots, 2M$. Then,

$$\left| (g * f)(x) - \frac{1}{N} \sum_{j=j'-\lceil \frac{6r}{\sqrt{2\pi}} \ln N \rceil - 1}^{j'+\lceil \frac{6r}{\sqrt{2\pi}} \ln N \rceil + 1} f(y_j) g(x - y_j) \right| \leq 3 \frac{\|\mathbf{f}\|_p}{N^r}.$$

As a consequence, we can see that $(g * f)(x)$ can always be computed to within $\mathcal{O}(\|\mathbf{f}\|_{\infty} N^{-r})$ -error in just $\mathcal{O}(r \log N)$ -time for any given $\mathbf{f} \in \mathbb{C}^N$ once the $\{g(x - y_j)\}_{j=j'-\lceil \frac{6r}{\sqrt{2\pi}} \ln N \rceil - 1}^{j'+\lceil \frac{6r}{\sqrt{2\pi}} \ln N \rceil + 1}$ have been precomputed.

Proof. Combining Theorem 3 and Lemma 10 we can see that

$$\left| (g * f)(x) - \frac{1}{N} \sum_{j=j'-\lceil \frac{6r}{\sqrt{2\pi}} \ln N \rceil - 1}^{j'+\lceil \frac{6r}{\sqrt{2\pi}} \ln N \rceil + 1} f(y_j) g(x - y_j) \right| \leq \|\mathbf{f}\|_p \left(\frac{1}{\beta\sqrt{\ln N}} N^{1 - \frac{\beta^2}{18} - \frac{1}{p}} + 2 N^{-\frac{2\pi^2\gamma^2}{\beta^2}} \right)$$

where $\beta = 6\sqrt{r} \geq 6$, $N \geq 36r = \beta^2$, and $\gamma = \frac{6r}{\sqrt{2\pi}} = \frac{\beta\sqrt{r}}{\sqrt{2\pi}}$. \square

We are now prepared to bound the error of the proposed approach when utilizing the SFTs developed in [19].

4. AN ERROR GUARANTEE FOR ALGORITHM 1 WHEN USING THE SFTs PROPOSED IN [19]

Given the 2π -periodic Gaussian $g : [-\pi, \pi] \rightarrow \mathbb{R}^+$ (3), consider the periodic modulation of g , $\tilde{g}_q : [-\pi, \pi] \rightarrow \mathbb{C}$, for any $q \in \mathbb{Z}$ defined by

$$\tilde{g}_q(x) = e^{-iqx} g(x).$$

One can see that

$$\tilde{g}_q(x) = e^{-iqx} \sum_{\omega=-\infty}^{\infty} \hat{g}_\omega e^{i\omega x} = \sum_{\omega=-\infty}^{\infty} \hat{g}_\omega e^{i(\omega-q)x} = \sum_{\tilde{\omega}=-\infty}^{\infty} \hat{g}_{\tilde{\omega}+q} e^{i\tilde{\omega}x},$$

so that the Fourier series coefficients of \tilde{g}_q are those of g , shifted by q ; that is,

$$\left(\widehat{\tilde{g}_q}\right)_\omega = \hat{g}_{\omega+q}.$$

In line 9 of Algorithm 1, we provide the SFT Algorithm in [19] with the approximate evaluations of $\{(\tilde{g}_q * f)(x_k)\}_{k=1}^m$, namely, $\{(\tilde{g}_q * f)(x_k) + n_k\}_{k=1}^m$, where, by Theorem 4, the perturbations n_k are bounded, for instance, by

$$|n_k| \leq 3 \frac{\|f\|_\infty}{N^r} \quad \forall k = 1, \dots, m.$$

With this in mind, let us apply Lemma 4 to the function $\tilde{g}_q * f$. We have the following lemma.

Lemma 11. *Let $s \in [2, N] \cap \mathbb{N}$, and $\mathbf{n} \in \mathbb{C}^m$ be the vector containing the total errors incurred by approximating $\tilde{g}_q * f$ via a truncated version of (11), as per Theorem 4. There exists a set of m points $\{x_k\}_{k=1}^m \subset [-\pi, \pi]$ such that Algorithm 3 on page 72 of [19], when given access to the corrupted samples $\{(\tilde{g}_q * f)(x_k) + n_k\}_{k=1}^m$, will identify a subset $S \subseteq B$ which is guaranteed to contain all $\omega \in B$ with*

$$\left| \left(\widehat{\tilde{g}_q * f}\right)_\omega \right| > 4 \left(\frac{1}{s} \cdot \left\| \widehat{\tilde{g}_q * f} - \left(\widehat{\tilde{g}_q * f}\right)_s^{\text{opt}} \right\|_1 + 3 \|f\|_\infty N^{-r} \right) =: 4\tilde{\delta}.$$

Furthermore, every $\omega \in S$ returned by Algorithm 3 will also have an associated Fourier series coefficient estimate $z_\omega \in \mathbb{C}$ which is guaranteed to have

$$\left| \left(\widehat{\tilde{g}_q * f}\right)_\omega - z_\omega \right| \leq \sqrt{2}\tilde{\delta}.$$

Next, we need to guarantee that the estimates of $\widehat{\tilde{g}_q * f}$ returned by Algorithm 3 of [19] will yield good estimates of \hat{f} itself. We have the following.

Lemma 12. *Let $s \in [2, N] \cap \mathbb{N}$. Given a 2π -periodic function $f : [-\pi, \pi] \rightarrow \mathbb{C}$, the periodic Gaussian g , and any of its modulations $\tilde{g}_q(x) = e^{-iqx} g(x)$, one has*

$$\left\| \widehat{\tilde{g}_q * f} - \left(\widehat{\tilde{g}_q * f}\right)_s^{\text{opt}} \right\|_1 \leq \frac{1}{2} \left\| \hat{f} - \hat{f}_s^{\text{opt}} \right\|_1.$$

Proof. Recall the definition of $R_s^{\text{opt}}(\hat{f})$ as the subset of B containing the s most energetic frequencies of \hat{f} , and observe that

$$\frac{1}{2} \left\| \hat{f} - \hat{f}_s^{\text{opt}} \right\|_1 = \frac{1}{2} \sum_{\omega \in B \setminus R_s^{\text{opt}}(\hat{f})} |\hat{f}_\omega| \geq \sum_{\omega \in B \setminus R_s^{\text{opt}}(\hat{f})} \left| \left(\widehat{\tilde{g}_q}\right)_\omega \cdot \hat{f}_\omega \right|$$

since, by Lemma 2, $\widehat{g}_\omega < \frac{1}{2}$ for all ω , and consequently, $\left(\widehat{g}_q\right)_\omega = \widehat{g}_{\omega+q} < \frac{1}{2}$ for all ω . Moreover,

$$\sum_{\omega \in B \setminus R_s^{\text{opt}}(\widehat{f})} \left| \left(\widehat{g}_q\right)_\omega \cdot \widehat{f}_\omega \right| \geq \sum_{\omega \in B \setminus R_s^{\text{opt}}(\widehat{g}_q * f)} \left| \left(\widehat{g}_q\right)_\omega \cdot \widehat{f}_\omega \right| = \left\| \widehat{g}_q * f - \left(\widehat{g}_q * f\right)_s^{\text{opt}} \right\|_1.$$

□

Let us combine the guarantees above into the following lemma.

Lemma 13. *Let $s \in [2, N] \cap \mathbb{N}$, and $\mathbf{n} \in \mathbb{C}^m$ be the vector containing the total errors incurred by approximating $\widehat{g}_q * f$ via a truncated version of (11), as per Theorem 4. There exists a set of m points $\{x_k\}_{k=1}^m \subset [-\pi, \pi]$ such that Algorithm 3 on page 72 of [19], when given access to the corrupted samples $\{(\widehat{g}_q * f)(x_k) + n_k\}_{k=1}^m$, will identify a subset $S \subseteq B$ which is guaranteed to contain all $\omega \in B$ with*

$$\left| \left(\widehat{g}_q * f\right)_\omega \right| > 4 \left(\frac{1}{2s} \cdot \left\| \widehat{f} - \widehat{f}_s^{\text{opt}} \right\|_1 + 3 \|\mathbf{f}\|_\infty N^{-r} \right) =: 4\delta.$$

Furthermore, every $\omega \in S$ returned by Algorithm 3 will also have an associated Fourier series coefficient estimate $z_\omega \in \mathbb{C}$ which is guaranteed to have

$$\left| \left(\widehat{g}_q\right)_\omega \cdot \widehat{f}_\omega - z_\omega \right| \leq \sqrt{2}\delta.$$

The lemma above implies that for any choice of q in line 4 of Algorithm 1, we are guaranteed to find all $\omega \in \left[q - \left\lceil \frac{N}{\alpha\sqrt{\ln N}} \right\rceil, q + \left\lceil \frac{N}{\alpha\sqrt{\ln N}} \right\rceil \right) \cap B$ with

$$\left| \widehat{f}_\omega \right| > \max_{\tilde{\omega}} \frac{4\delta}{\left(\widehat{g}_q\right)_{\tilde{\omega}}} \geq \frac{4\delta}{\tau}$$

where α and τ are as defined in Lemma 3. Moreover, the Fourier series coefficient estimates z_ω returned by Algorithm 3 will satisfy

$$\left| \widehat{f}_\omega - \frac{z_\omega}{\left(\widehat{g}_q\right)_\omega} \right| \leq \max_{\tilde{\omega}} \frac{\sqrt{2}\delta}{\left(\widehat{g}_q\right)_{\tilde{\omega}}} \leq \frac{\sqrt{2}\delta}{\tau}.$$

Following Theorem 3, which guarantees a decay of N^{-r} in the total approximation error, let us set $\beta = 6\sqrt{r}$ for $1 \leq r \leq \frac{N}{36}$. Recall from Lemma 3 the choice of $\beta \in \left(0, \alpha\sqrt{\frac{\ln(1/\tau\sqrt{2\pi})}{2}} \right]$ where τ is to be chosen from $\left(0, \frac{1}{\sqrt{2\pi}} \right)$. Thus, we must choose $\alpha \in \left[1, \frac{N}{\sqrt{\ln N}} \right]$ so that

$$6\sqrt{r} \leq \alpha\sqrt{\frac{\ln(1/\tau\sqrt{2\pi})}{2}} \iff \alpha \geq \frac{6\sqrt{2r}}{\ln(1/\tau\sqrt{2\pi})}.$$

We may remove the dependence on τ simply by setting, e.g., $\tau = \frac{1}{3}$. Then $\alpha = \mathcal{O}(\sqrt{r})$.

We are now ready to state the recovery guarantee of Algorithm 1 and its operation count.

Theorem 5. *Let $N \in \mathbb{N}$, $s \in [2, N] \cap \mathbb{N}$, and $1 \leq r \leq \frac{N}{36}$ as in Theorem 4. If Algorithm 3 of [19] is used in Algorithm 1 then Algorithm 1 will always deterministically identify a subset $S \subseteq B$ and a sparse vector $\mathbf{v}|_S \in \mathbb{C}^N$ satisfying*

$$(13) \quad \left\| \widehat{\mathbf{f}} - \mathbf{v}|_S \right\|_2 \leq \left\| \widehat{\mathbf{f}} - \widehat{\mathbf{f}}_s^{\text{opt}} \right\|_2 + \frac{33}{\sqrt{s}} \cdot \left\| \widehat{\mathbf{f}} - \widehat{\mathbf{f}}_s^{\text{opt}} \right\|_1 + 198\sqrt{s} \|\mathbf{f}\|_\infty N^{-r}.$$

Algorithm 1's operation count is then

$$\mathcal{O}\left(\frac{s^2 \cdot r^{\frac{3}{2}} \cdot \log^{\frac{11}{2}}(N)}{\log(s)}\right).$$

If returning a sparse vector $\mathbf{v}|_S \in \mathbb{C}^N$ that satisfies (13) with probability at least $(1-p) \in [2/3, 1)$ is sufficient, a Monte Carlo variant of the deterministic Algorithm 3 in [19] may be used in line 9 of Algorithm 1. In this case Algorithm 1's operation count is

$$\mathcal{O}\left(s \cdot r^{\frac{3}{2}} \cdot \log^{\frac{9}{2}}(N) \cdot \log\left(\frac{N}{p}\right)\right).$$

Proof. Redefine δ in the proof of Theorem 7 in [19] as

$$\delta = \frac{1}{\tau} \left(\frac{1}{2s} \cdot \left\| \hat{f} - \hat{f}_s^{\text{opt}} \right\|_1 + 3 \|\mathbf{f}\|_{\infty} N^{-r} \right) = 3 \left(\frac{1}{2s} \cdot \left\| \hat{\mathbf{f}} - \hat{\mathbf{f}}_s^{\text{opt}} \right\|_1 + 3 \|\mathbf{f}\|_{\infty} N^{-r} \right),$$

and observe that any $\omega \in B = [-\lceil \frac{N}{2} \rceil, \lfloor \frac{N}{2} \rfloor] \cap \mathbb{Z}$ that is reconstructed by Algorithm 1 will have a Fourier series coefficient estimate v_{ω} that satisfies

$$\left| v_{\omega} - \hat{\mathbf{f}}_{\omega} \right| = \left| v_{\omega} - \hat{f}_{\omega} \right| \leq \sqrt{2} \cdot \delta.$$

We can thus bound the approximation error by

$$\begin{aligned} \left\| \hat{\mathbf{f}} - \mathbf{v}|_S \right\|_2 &\leq \left\| \hat{\mathbf{f}} - \hat{\mathbf{f}}|_S \right\|_2 + \left\| \hat{\mathbf{f}}|_S - \mathbf{v}|_S \right\|_2 \leq \left\| \hat{\mathbf{f}} - \hat{\mathbf{f}}|_S \right\|_2 + 2\sqrt{s} \cdot \delta \\ (14) \quad &= \sqrt{\left\| \hat{\mathbf{f}} - \hat{\mathbf{f}}_s^{\text{opt}} \right\|_2^2 + \sum_{\omega \in R_s^{\text{opt}}(\hat{f}) \setminus S} \left| \hat{f}_{\omega} \right|^2 - \sum_{\tilde{\omega} \in S \setminus R_s^{\text{opt}}(\hat{f})} \left| \hat{f}_{\tilde{\omega}} \right|^2} + 2\sqrt{s} \cdot \delta. \end{aligned}$$

In order to make additional progress on (14) we must consider the possible magnitudes of $\hat{\mathbf{f}}$ entries at indices in $S \setminus R_s^{\text{opt}}(\hat{f})$ and $R_s^{\text{opt}}(\hat{f}) \setminus S$. Careful analysis (in line with the techniques employed in the proof of Theorem 7 of [19]) indicates that

$$\sum_{\omega \in R_s^{\text{opt}}(\hat{f}) \setminus S} \left| \hat{f}_{\omega} \right|^2 - \sum_{\tilde{\omega} \in S \setminus R_s^{\text{opt}}(\hat{f})} \left| \hat{f}_{\tilde{\omega}} \right|^2 \leq s \cdot (8\sqrt{2} + 8)^2 \cdot \delta^2.$$

Therefore, in the worst possible case equation (14) will remain bounded by

$$\left\| \hat{\mathbf{f}} - \mathbf{v}|_S \right\|_2 \leq \sqrt{\left\| \hat{\mathbf{f}} - \hat{\mathbf{f}}_s^{\text{opt}} \right\|_2^2 + s \cdot (8\sqrt{2} + 8)^2 \cdot \delta^2} + 2\sqrt{s} \cdot \delta \leq \left\| \hat{\mathbf{f}} - \hat{\mathbf{f}}_s^{\text{opt}} \right\|_2 + 22\sqrt{s} \cdot \delta.$$

The error bound stated in (13) follows.

The runtimes follow by observing that $c_2 = \mathcal{O}\left(\alpha \cdot \log^{\frac{1}{2}}(N)\right) = \mathcal{O}\left(r^{\frac{1}{2}} \cdot \log^{\frac{1}{2}}(N)\right)$ as chosen in line 2 of Algorithm 1, and for every choice of q in line 4 of Algorithm 1, all of the evaluations $\{(\tilde{g}_q * f)(x_k)\}_{k=1}^m$ can be approximated very accurately in just $\mathcal{O}(mr \log N)$ -time, where the number of samples m is on the orders described in Theorem 2. \square

We are now ready to empirically evaluate Algorithm 1 with several different SFT algorithms \mathcal{A} used in its line 9.

5. NUMERICAL EVALUATION

In this section we evaluate the performance of three new discrete SFT Algorithms resulting from Algorithm 1: DMSFT-4, DMSFT-6,⁷ and CLW-DSFT.⁸ All of them were developed by utilizing different SFT algorithms in line 9 of Algorithm 1. Here DMSFT stands for the **D**iscrete **M**ichigan **S**tate **F**ourier **T**ransform algorithm. Both DMSFT-4 and DMSFT-6 are implementations of Algorithm 1 that use a randomized version of the SFT algorithm GFFT [25] in their line 9.⁹ The only difference between DMSFT-4 and DMSFT-6 is how accurately each one estimates the convolution in line 7 of Algorithm 1: for DMSFT-4 we use $\kappa = 4$ in the partial discrete convolution in Lemma 10 when approximating $\tilde{g}_q * f$ at each x_k , while for DMSFT-6 we always use $\kappa = 6$. The CLW-DSFT stands for the **C**hristlieb **L**awlor **W**ang **D**iscrete **S**parsely **F**ourier **T**ransform algorithm. It is an implementation of Algorithm 1 that uses the SFT developed in [5] in its line 9, and κ varying between 12 and 20 for its line 7 convolution estimates (depending on each input vector’s Fourier sparsity, etc.). All of DMSFT-4, DMSFT-6 and CLW-DSFT were implemented in C++ in order to empirically evaluate their runtime and noise robustness characteristics.

We also compare these new implementations’ runtime and robustness characteristics with FFTW 3.3.4¹⁰ and sFFT 2.0¹¹. FFTW is the highly optimized FFT implementation which runs in $\mathcal{O}(N \log N)$ -time for input vectors of length N . All the standard discrete Fourier Transforms in the numerical experiments are performed using FFTW 3.3.4 with FFTW_MEASURE plan. The sFFT 2.0 is a randomized discrete sparse Fourier Transform algorithm written in C++ which is both stable and robust to noise. It was developed by Indyk et al. in [12]. Note that DMSFT-4, DMSFT-6, CLW-DSFT, and sFFT 2.0 are all randomized algorithms designed to approximate discrete DFTs that are approximately s -sparse. This means that all of them take both sparsity s and size N of the DFT’s $\hat{\mathbf{f}} \in \mathbb{C}^N$ they aim to recover as parameters. In contrast, FFTW can not utilize existing sparsity to its advantage. Finally, all experiments are run on a Linux CentOS machine with 2.50GHz CPU and 16 GB of RAM.

5.1. Experiment Setup. For the execution time experiments each trial input vector $\mathbf{f} \in \mathbb{C}^N$ was generated as follows: First s frequencies were independently selected uniformly at random from $[0, N) \cap \mathbb{Z}$, and then each of these frequencies was assigned a uniform random phase with magnitude 1 as its Fourier coefficient. The remaining frequencies’ Fourier coefficients were then set to zero to form $\hat{\mathbf{f}} \in \mathbb{C}^N$. Finally, the trial input vector \mathbf{f} was then formed via an inverse DFT.

For each pair of s and N the parameters in each randomized algorithm were chosen so that the probability of correctly recovering all s energetic frequencies was at least 0.9 per trial input. Every data point in a figure below corresponds to an average over 100 runs on 100 different trial input vectors of this kind. It is worth mentioning that the parameter tuning process for DMSFT-4 and DMSFT-6 requires significantly less effort than for both CLW-DSFT and sFFT 2.0 since the DMSFT variants only have two parameters (whose default values are generally near-optimal).

5.2. Runtime as Input Vector Size Varies. In Figure 1 we fixed the sparsity to $s = 50$ and ran numerical experiments on 8 different input vector lengths N : $2^{16}, 2^{18}, \dots, 2^{30}$. We then plotted the running time (averaged over 100 runs) for DMSFT-4, DMSFT-6, CLW-DSFT, sFFT 2.0, and FFTW.

As expected, the runtime slope of all the SFT algorithms (i.e. DMSFT-4, DMSFT-6, CLW-DSFT, and sFFT 2.0) is less than the slope of FFTW as N increases. Although FFTW is fastest

⁷The code for both DMSFT variants is available at <https://sourceforge.net/projects/aafftannarborfa/>.

⁸The CLW-DSFT code is available at www.math.msu.edu/~markiwen/Code.html.

⁹Code for GFFT is also available at www.math.msu.edu/~markiwen/Code.html.

¹⁰This code is available at <http://www.fftw.org/>

¹¹This code is available at <https://groups.csail.mit.edu/netmit/sFFT/>

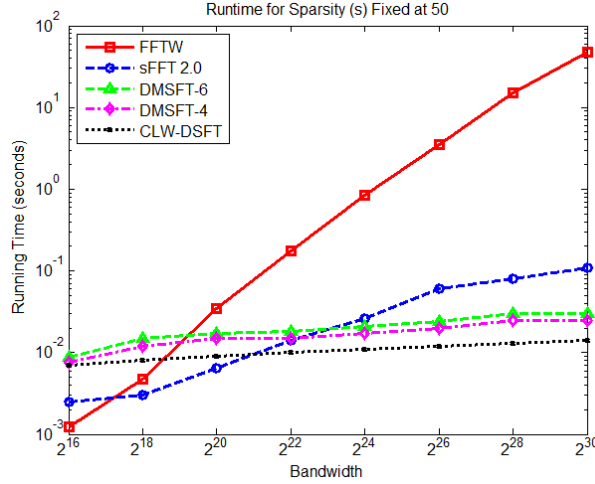


FIGURE 1. Runtime Comparison at Sparsity (s) Fixed at 50

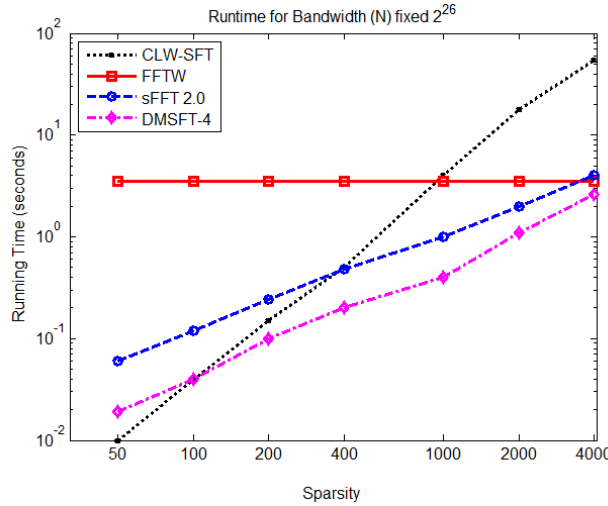


FIGURE 2. Runtime Comparison at Bandwidth (N) Fixed at 2^{26}

for vectors of small size, it becomes the slowest algorithm when the vector size N is greater than 2^{20} . Among the randomized algorithms, sFFT 2.0 is the fastest one when N is less than 2^{22} , but DMSFT-4, DMSFT-6, and CLW-DSFT all outperform sFFT 2.0 with respect to runtime when the input vector's sizes are large enough. The CLW-DSFT implementation becomes faster than sFFT 2.0 when N is approximately 2^{21} while DMSFT-4 and DMSFT-6 have better runtime performance than sFFT 2.0 when N is greater than 2^{23} .

5.3. Runtime as Sparsity Varies. In Figure 2 we fix the input vector lengths to $N = 2^{26}$ and run the numerical experiments on 7 different values of sparsity s : 50, 100, 200, 400, 1000, 2000, and 4000. As expected, the FFTW's runtime is constant as we increase the sparsity. The runtimes of DMSFT-4, CLW-DSFT, and sFFT 2.0 are all essentially linear in s . Here DMSFT-6 has been excluded for ease of viewing/reference – its runtimes lie directly above those of DMSFT-4 when included in the plot. Looking at Figure 2 we can see the CLW-DSFT's runtime increases more rapidly with s than that of DMSFT-4 and sFFT 2.0. The runtime of CLW-DSFT becomes the slowest one when sparsity is around 1000. DMSFT-4 and sFFT 2.0 have approximately the

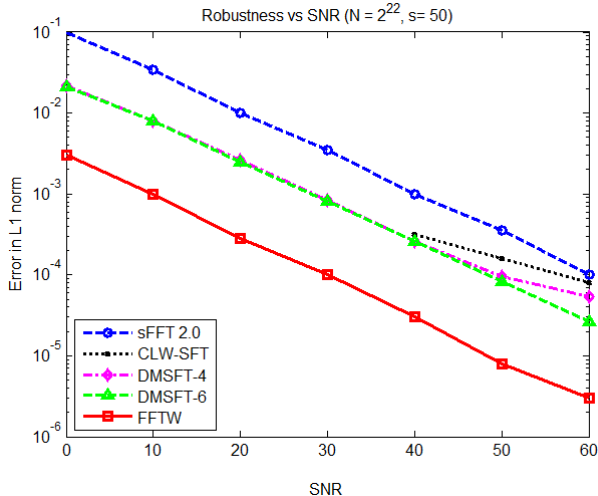


FIGURE 3. Robustness to Noise (Bandwidth ($N = 2^{22}$), Sparsity ($s = 50$)).

same runtime slope as s increases, and they both have good performance when the sparsity is large. However, DMSFT-4 maintains consistently better runtime performance than sFFT 2.0 for all sparsity values, and is the only algorithm in the plot that still faster than FFTW when the sparsity is 4000. Indeed, when the sparsity is 4000 the average runtime of DMSFT-4 is 2.68s and the average runtime of DMSFT-6 is 2.9s. Both of them remain faster than FFTW (3.47s) and sFFT 2.0 (3.96s) for this large sparsity (though only DMSFT-4 has been included in the plot above).

5.4. Robustness to Noise. In our final set of experiments we test the noise robustness of DMSFT-4, DMSFT-6, CLW-DSFT, sFFT 2.0, and FFTW for different levels of Gaussian noise. Here the size of each input vector is $N = 2^{22}$ and sparsity is fixed at $s = 50$. The test signals are then generated as before, except that Gaussian noise is added to \mathbf{f} after it is constructed. More specifically, we first generate \mathbf{f} and then set $\mathbf{f} = \mathbf{f} + \mathbf{n}$ where each entry of \mathbf{n} , n_j , is an i.i.d. mean 0 random complex Gaussian value. The noise vector \mathbf{n} is then rescaled to achieve each desired signal-to-noise ratio (SNR) considered in the experiments.¹²

Recall that the randomized algorithms compared herein (DMSFT-4, DMSFT-6, CLW-DSFT, and sFFT 2.0) are all tuned to guarantee exact recovery of s -sparse functions with probability at least 0.9 in all experiments. For our noise robustness experiments this ensures that the correct frequency support, S , is found for at least 90 of the 100 trial signals used to generate each point plotted in Figure 3. We use average L_1 error to measure the noise robustness of each algorithm for each of these at least 90 trial runs. The average L_1 error is defined as

$$\text{Average } L_1 \text{ Error} = \frac{1}{s} \sum_{\omega \in S} |\hat{f}_\omega - z_\omega|$$

where S is the true frequency support of the input vector \mathbf{f} , \hat{f}_ω are the true input Fourier coefficients for all frequencies $\omega \in S$, and z_ω are their recovered approximations from each algorithm. Figure 3 graphs the averaged average L_1 error over the at least 90 trial signals where each method correctly identified S .

It can be seen in Figure 3 that DMSFT-4, DMSFT-6, sFFT 2.0, and FFTW are all robust to noise. As expected, FFTW has the best performance in this test. DMSFT-4 and DMSFT-6 are

¹²The SNR is defined as $SNR = 20 \log \frac{\|\mathbf{f}\|_2}{\|\mathbf{n}\|_2}$, where \mathbf{f} is the length N input vector and \mathbf{n} is the length N noise vector.

both more robust to noise when compared to sFFT 2.0. As for CLW-DSFT, it cannot guarantee a 0.9 probability of correctly recovering S when the SNR is less than 40 and so is not plotted for those SNR values. This is due to the base energetic frequency identification methods of [21, 5] being inherently ill conditioned, though the CLW-DSFT results look better when compared to the true $\hat{\mathbf{f}}$ with respect to, e.g., earth mover’s distance. Frequencies are often estimated incorrectly by CLW-DSFT at higher noise levels, but when they are they are usually at least close enough to the true frequencies to be informative.

6. CONCLUSION

Let \mathcal{A} be a sublinear-time sparse FFT algorithm which utilizes unequally spaced samples from a given periodic function $f : [-\pi, \pi] \rightarrow \mathbb{C}$ in order to rapidly approximate its sequence of Fourier series coefficients $\hat{f} \in \ell^2$. In this paper we propose a generic method of transforming any such algorithm \mathcal{A} into a sublinear-time sparse DFT algorithm which rapidly approximates $\hat{\mathbf{f}}$ from a given input vector $\mathbf{f} \in \mathbb{C}^N$. As a result we are able to construct several new sublinear-time sparse DFT algorithms from existing sparse Fourier algorithms which utilize unequally spaced function samples [25, 19, 21, 5]. The best of these new algorithms is shown to outperform existing discrete sparse Fourier transform methods with respect to both runtime and noise robustness for large vector lengths N . In addition, we also present several new theoretical discrete sparse FFT robust recovery guarantees. These include the first known theoretical guarantees for entirely deterministic and discrete sparse DFT algorithms which hold for arbitrary input vectors $\mathbf{f} \in \mathbb{C}^N$.

REFERENCES

- [1] J. Bailey, M. A. Iwen, and C. V. Spencer. On the design of deterministic matrices for fast recovery of fourier compressible functions. *SIAM Journal on Matrix Analysis and Applications*, 33(1):263–289, 2012.
- [2] S. Bittens. Sparse fft for functions with short frequency support. *University of Göttingen*, 2016.
- [3] S. Bittens, R. Zhang, and M. A. Iwen. A deterministic sparse fft for functions with structured fourier sparsity. *arXiv preprint arXiv:1705.05256*, 2017.
- [4] L. Bluestein. A linear filtering approach to the computation of discrete fourier transform. *IEEE Transactions on Audio and Electroacoustics*, 18(4):451–455, 1970.
- [5] A. Christlieb, D. Lawlor, and Y. Wang. A multiscale sub-linear time fourier algorithm for noisy data. *Applied and Computational Harmonic Analysis*, 40:553 – 574, 2016.
- [6] A. Cohen, W. Dahmen, and R. DeVore. Compressed sensing and best k-term approximation. *Journal of the American mathematical society*, 22(1):211–231, 2009.
- [7] J. W. Cooley and J. W. Tukey. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301, 1965.
- [8] S. Foucart and H. Rauhut. *A mathematical introduction to compressive sensing*. Birkhäuser Basel, 2013.
- [9] A. C. Gilbert, P. Indyk, M. Iwen, and L. Schmidt. Recent developments in the sparse fourier transform: a compressed fourier transform for big data. *IEEE Signal Processing Magazine*, 31(5):91–100, 2014.
- [10] A. C. Gilbert, S. Muthukrishnan, and M. Strauss. Improved time bounds for near-optimal sparse fourier representations. In *Optics & Photonics 2005*, pages 59141A–59141A. International Society for Optics and Photonics, 2005.
- [11] A. C. Gilbert, M. J. Strauss, and J. A. Tropp. A tutorial on fast fourier sampling. *IEEE Signal Processing Magazine*, 25(2):57–66, 2008.
- [12] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Simple and practical algorithm for sparse fourier transform. *Proc. of SODA*, January, 2012.
- [13] X. Hu, M. Iwen, and H. Kim. Rapidly computing sparse legendre expansions via sparse fourier transforms. *Numerical Algorithms*, 74(4):1029 – 1059, 2017.
- [14] M. Iwen, A. Gilbert, M. Strauss, et al. Empirical evaluation of a sub-linear time sparse dft algorithm. *Communications in Mathematical Sciences*, 5(4):981–998, 2007.
- [15] M. A. Iwen. A deterministic sub-linear time sparse fourier algorithm via non-adaptive compressed sensing methods. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 20–29. Society for Industrial and Applied Mathematics, 2008.

- [16] M. A. Iwen. Simple deterministically constructible rip matrices with sublinear fourier sampling requirements. In *CISS*, pages 870–875, 2009.
- [17] M. A. Iwen. Combinatorial sublinear-time Fourier algorithms. *Foundations of Computational Mathematics*, 10:303 – 338, 2010.
- [18] M. A. Iwen. Notes on lemma 6. *Preprint at www.math.msu.edu/~markiwen/Papers/Lemma6_FOCM_10.pdf*, 2012.
- [19] M. A. Iwen. Improved approximation guarantees for sublinear-time Fourier algorithms. *Applied and Computational Harmonic Analysis*, 34:57–82, 2013.
- [20] J. Laska, S. Kirolos, Y. Massoud, R. Baraniuk, A. Gilbert, M. Iwen, and M. Strauss. Random sampling for analog-to-information conversion of wideband signals. In *Design, Applications, Integration and Software, 2006 IEEE Dallas/CAS Workshop on*, pages 119–122. IEEE, 2006.
- [21] D. Lawlor, Y. Wang, and A. Christlieb. Adaptive sub-linear time fourier algorithms. *Advances in Adaptive Data Analysis*, 5(01):1350003, 2013.
- [22] L. Morotti. Explicit universal sampling sets in finite vector spaces. *Applied and Computational Harmonic Analysis*, 2016.
- [23] G. Plonka and K. Wannenwetsch. A deterministic sparse fft algorithm for vectors with small support. *Numerical Algorithms*, 71(4):889–905, 2016.
- [24] L. Rabiner, R. Schafer, and C. Rader. The chirp z-transform algorithm. *IEEE transactions on audio and electroacoustics*, 17(2):86–92, 1969.
- [25] I. Segal and M. Iwen. Improved sparse fourier approximation results: Faster implementations and stronger guarantees. *Numerical Algorithms*, 63:239 – 263, 2013.

APPENDIX A. PROOF OF LEMMAS 1, 2 AND 3

We will restate each lemma before its proof for ease of reference.

Lemma 14 (Restatement of Lemma 1). *The 2π -periodic Gaussian $g : [-\pi, \pi] \rightarrow \mathbb{R}^+$ has*

$$g(x) \leq \left(\frac{3}{c_1} + \frac{1}{\sqrt{2\pi}} \right) e^{-\frac{x^2}{2c_1^2}}$$

for all $x \in [-\pi, \pi]$.

Proof. Observe that

$$\begin{aligned} c_1 g(x) &= \sum_{n=-\infty}^{\infty} e^{-\frac{(x-2n\pi)^2}{2c_1^2}} = e^{-\frac{x^2}{2c_1^2}} + e^{-\frac{(x-2\pi)^2}{2c_1^2}} + e^{-\frac{(x+2\pi)^2}{2c_1^2}} + \sum_{|n| \geq 2} e^{-\frac{(x-2n\pi)^2}{2c_1^2}} \\ &\leq 3e^{-\frac{x^2}{2c_1^2}} + \int_1^{\infty} e^{-\frac{(x-2n\pi)^2}{2c_1^2}} dn + \int_1^{\infty} e^{-\frac{(x+2n\pi)^2}{2c_1^2}} dn \end{aligned}$$

holds since the series above have monotonically decreasing positive terms, and $x \in [-\pi, \pi]$.

Now, if $x \in [0, \pi]$ and $n \geq 1$, one has

$$e^{-\frac{(2n+1)^2\pi^2}{2c_1^2}} \leq e^{-\frac{(x+2n\pi)^2}{2c_1^2}} \leq e^{-\frac{4n^2\pi^2}{2c_1^2}} \leq e^{-\frac{(x-2n\pi)^2}{2c_1^2}} \leq e^{-\frac{(2n-1)^2\pi^2}{2c_1^2}},$$

which yields

$$\begin{aligned} c_1 g(x) &\leq 3e^{-\frac{x^2}{2c_1^2}} + 2 \int_1^{\infty} e^{-\frac{\pi^2(2n-1)^2}{2c_1^2}} dn = 3e^{-\frac{x^2}{2c_1^2}} + \frac{1}{2} \left(\int_{-\infty}^{\infty} e^{-\frac{\pi^2 m^2}{2c_1^2}} dm - \int_{-1}^1 e^{-\frac{\pi^2 m^2}{2c_1^2}} dm \right) \\ &= 3e^{-\frac{x^2}{2c_1^2}} + \frac{c_1}{\sqrt{2\pi}} - \frac{1}{2} \int_{-1}^1 e^{-\frac{\pi^2 m^2}{2c_1^2}} dm. \end{aligned}$$

Using Lemma 8 to bound the last integral we can now get that

$$\begin{aligned} c_1 g(x) &\leq 3e^{-\frac{x^2}{2c_1^2}} + \frac{c_1}{\sqrt{2\pi}} - \frac{1}{2} \frac{\sqrt{2}c_1}{\pi} \sqrt{\pi \left(1 - e^{-\frac{\pi^2}{2c_1^2}}\right)} = 3e^{-\frac{x^2}{2c_1^2}} + \frac{c_1}{\sqrt{2\pi}} \left(1 - \sqrt{\left(1 - e^{-\frac{\pi^2}{2c_1^2}}\right)}\right) \\ &\leq 3e^{-\frac{x^2}{2c_1^2}} + \frac{c_1}{\sqrt{2\pi}} e^{-\frac{\pi^2}{2c_1^2}} \leq 3e^{-\frac{x^2}{2c_1^2}} + \frac{c_1}{\sqrt{2\pi}} e^{-\frac{x^2}{2c_1^2}}. \end{aligned}$$

Recalling now that g is even we can see that this inequality will also hold for all $x \in [-\pi, 0]$ as well. \square

Lemma 15 (Restatement of Lemma 2). *The 2π -periodic Gaussian $g : [-\pi, \pi] \rightarrow \mathbb{R}^+$ has*

$$\widehat{g}_\omega = \frac{1}{\sqrt{2\pi}} e^{-\frac{c_1^2 \omega^2}{2}}$$

for all $\omega \in \mathbb{Z}$. Thus, $\widehat{g} = \{\widehat{g}_\omega\}_{\omega \in \mathbb{Z}} \in \ell^2$ decreases monotonically as $|\omega|$ increases, and also has $\|\widehat{g}\|_\infty = \frac{1}{\sqrt{2\pi}}$.

Proof. Starting with the definition of the Fourier transform, we calculate

$$\begin{aligned} \widehat{g}_\omega &= \frac{1}{c_1} \sum_{n=-\infty}^{\infty} \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-\frac{(x-2n\pi)^2}{2c_1^2}} e^{-i\omega x} dx \\ &= \frac{1}{c_1} \sum_{n=-\infty}^{\infty} \frac{1}{2\pi} \int_{-\pi}^{\pi} e^{-\frac{(x-2n\pi)^2}{2c_1^2}} e^{-i\omega(x-2n\pi)} dx \\ &= \frac{1}{c_1} \sum_{n=-\infty}^{\infty} \frac{1}{2\pi} \int_{-\pi-2n\pi}^{\pi-2n\pi} e^{-\frac{u^2}{2c_1^2}} e^{-i\omega u} du \\ &= \frac{1}{2\pi c_1} \int_{-\infty}^{\infty} e^{-\frac{u^2}{2c_1^2}} e^{-i\omega u} du \\ &= \frac{c_1 \sqrt{2\pi}}{2\pi c_1} e^{-\frac{c_1^2 \omega^2}{2}} \\ &= \frac{e^{-\frac{c_1^2 \omega^2}{2}}}{\sqrt{2\pi}}. \end{aligned}$$

The last two assertions now follow easily. \square

Lemma 16 (Restatement of Lemma 3). *Choose any $\tau \in \left(0, \frac{1}{\sqrt{2\pi}}\right)$, $\alpha \in \left[1, \frac{N}{\sqrt{\ln N}}\right]$, and $\beta \in \left(0, \alpha \sqrt{\frac{\ln(1/\tau\sqrt{2\pi})}{2}}\right]$. Let $c_1 = \frac{\beta\sqrt{\ln N}}{N}$ in the definition of the periodic Gaussian g from (3). Then $\widehat{g}_\omega \in \left[\tau, \frac{1}{\sqrt{2\pi}}\right]$ for all $\omega \in \mathbb{Z}$ with $|\omega| \leq \left\lceil \frac{N}{\alpha\sqrt{\ln N}} \right\rceil$.*

Proof. By Lemma 2 above it suffices to show that

$$\frac{1}{\sqrt{2\pi}} e^{-\frac{c_1^2 \left(\left\lceil \frac{N}{\alpha\sqrt{\ln N}} \right\rceil\right)^2}{2}} \geq \tau,$$

which holds if and only if

$$c_1^2 \left(\left\lceil \frac{N}{\alpha\sqrt{\ln N}} \right\rceil \right)^2 \leq 2 \ln \left(\frac{1}{\tau\sqrt{2\pi}} \right)$$

$$c_1 \leq \frac{\sqrt{2 \ln \left(\frac{1}{\tau\sqrt{2\pi}} \right)}}{\left\lceil \frac{N}{\alpha\sqrt{\ln N}} \right\rceil}.$$

Thus, it is enough to have

$$c_1 \leq \frac{\sqrt{2 \ln \left(\frac{1}{\tau\sqrt{2\pi}} \right)}}{\frac{N}{\alpha\sqrt{\ln N}} + 1} = \frac{\alpha\sqrt{2 \ln \left(\frac{1}{\tau\sqrt{2\pi}} \right)} \ln N}{N + \alpha\sqrt{\ln N}},$$

or,

$$c_1 = \frac{\beta\sqrt{\ln N}}{N} \leq \frac{\alpha\sqrt{2 \ln \left(\frac{1}{\tau\sqrt{2\pi}} \right)} \ln N}{2N} \leq \frac{\alpha\sqrt{2 \ln \left(\frac{1}{\tau\sqrt{2\pi}} \right)} \ln N}{N + \alpha\sqrt{\ln N}}.$$

This, in turn, is guaranteed by our choice of β . \square

APPENDIX B. PROOF OF LEMMA 4 AND THEOREM 2

We will restate Lemma 4 before its proof for ease of reference.

Lemma 17 (Restatement of Lemma 4). *Let $s, \epsilon^{-1} \in \mathbb{N} \setminus \{1\}$ with $(s/\epsilon) \geq 2$, and $\mathbf{n} \in \mathbb{C}^m$ be an arbitrary noise vector. There exists a set of m points $\{x_k\}_{k=1}^m \subset [-\pi, \pi]$ such that Algorithm 3 on page 72 of [19], when given access to the corrupted samples $\{f(x_k) + n_k\}_{k=1}^m$, will identify a subset $S \subseteq B$ which is guaranteed to contain all $\omega \in B$ with*

$$(15) \quad \left| \hat{f}_\omega \right| > 4 \left(\frac{\epsilon \cdot \left\| \hat{\mathbf{f}} - \hat{\mathbf{f}}_{(s/\epsilon)}^{\text{opt}} \right\|_1}{s} + \left\| \hat{f} - \hat{f}|_B \right\|_1 + \|\mathbf{n}\|_\infty \right).$$

Furthermore, every $\omega \in S$ returned by Algorithm 3 will also have an associate Fourier series coefficient estimate $z_\omega \in \mathbb{C}$ which is guaranteed to have

$$(16) \quad \left| \hat{f}_\omega - z_\omega \right| \leq \sqrt{2} \left(\frac{\epsilon \cdot \left\| \hat{\mathbf{f}} - \hat{\mathbf{f}}_{(s/\epsilon)}^{\text{opt}} \right\|_1}{s} + \left\| \hat{f} - \hat{f}|_B \right\|_1 + \|\mathbf{n}\|_\infty \right).$$

Both the number of required samples, m , and Algorithm 3's operation count are

$$(17) \quad \mathcal{O} \left(\frac{s^2 \cdot \log^4(N)}{\log \left(\frac{s}{\epsilon} \right) \cdot \epsilon^2} \right).$$

If succeeding with probability $(1 - \delta) \in [2/3, 1)$ is sufficient, and $(s/\epsilon) \geq 2$, the Monte Carlo variant of Algorithm 3 referred to by Corollary 4 on page 74 of [19] may be used. This Monte Carlo variant reads only a randomly chosen subset of the noisy samples utilized by the deterministic algorithm,

$$\{f(\tilde{x}_k) + \tilde{n}_k\}_{k=1}^{\tilde{m}} \subseteq \{f(x_k) + n_k\}_{k=1}^m,$$

yet it still outputs a subset $S \subseteq B$ which is guaranteed to simultaneously satisfy both of the following properties with probability at least $1 - \delta$:

- (i) S will contain all $\omega \in B$ satisfying (15), and
- (ii) all $\omega \in S$ will have an associated coefficient estimate $z_\omega \in \mathbb{C}$ satisfying (16).

Finally, both this Monte Carlo variant's number of required samples, \tilde{m} , as well as its operation count will also always be

$$(18) \quad \mathcal{O}\left(\frac{s}{\epsilon} \cdot \log^3(N) \cdot \log\left(\frac{N}{\delta}\right)\right).$$

Proof. The proof of this lemma involves a somewhat tedious and uninspired series of minor modifications to various results from [19]. In what follows we will outline the portions of that paper which need to be changed in order to obtain the stated lemma. Algorithm 3 on page 72 of [19] will provide the basis of our discussion.

In the first paragraph of our lemma we are provided with m -contaminated evaluations of f , $\{f(x_k) + n_k\}_{k=1}^m$, at the set of m points $\{x_k\}_{k=1}^m \subset [-\pi, \pi]$ required by line 4 of Algorithm 1 on page 67 of [19]. These contaminated evaluations of f will then be used to approximate the vector $\mathcal{G}_{\lambda, K} \tilde{\psi} \mathbf{A} \in \mathbb{C}^m$ in line 4 of Algorithm 3. More specifically, using (18) on page 67 of [19] one can see that each $(\mathcal{G}_{\lambda, K} \tilde{\psi} \mathbf{A})_j \in \mathbb{C}$ is effectively computed via a DFT

$$(19) \quad (\mathcal{G}_{\lambda, K} \tilde{\psi} \mathbf{A})_j = \frac{1}{s_j} \sum_{k=0}^{s_j-1} f\left(-\pi + \frac{2\pi k}{s_j}\right) e^{\frac{-2\pi i k h_j}{s_j}}$$

for some integers $0 \leq h_j < s_j$. Note that we are guaranteed to have noisy evaluations of f at each of these points by assumption. That is, we have $f(x_{j,k}) + n_{j,k}$ for all $x_{j,k} := -\pi + \frac{2\pi k}{s_j}$, $k = 0, \dots, s_j - 1$.

We therefore approximate each $(\mathcal{G}_{\lambda, K} \tilde{\psi} \mathbf{A})_j$ via an approximate DFT as per (19) by

$$E_j := \frac{1}{s_j} \sum_{k=0}^{s_j-1} (f(x_{j,k}) + n_{j,k}) e^{\frac{-2\pi i k h_j}{s_j}}.$$

One can now see that

$$(20) \quad \left| E_j - (\mathcal{G}_{\lambda, K} \tilde{\psi} \mathbf{A})_j \right| = \left| \frac{1}{s_j} \sum_{k=0}^{s_j-1} n_{j,k} e^{\frac{-2\pi i k h_j}{s_j}} \right| \leq \frac{1}{s_j} \sum_{k=0}^{s_j-1} |n_{j,k}| \leq \|\mathbf{n}\|_\infty$$

holds for all j . Every entry of both $\mathcal{E}_{s_1, K} \tilde{\psi} \mathbf{A}$ and $\mathcal{G}_{\lambda, K} \tilde{\psi} \mathbf{A}$ referred to in Algorithm 3 will therefore be effectively replaced by its corresponding E_j estimate. Thus, the lemma we seek to prove is essentially obtained by simply incorporating the additional error estimate (20) into the analysis of Algorithm 3 in [19] wherever an $\mathcal{E}_{s_1, K} \tilde{\psi} \mathbf{A}$ or $\mathcal{G}_{\lambda, K} \tilde{\psi} \mathbf{A}$ currently appears.

To show that lines 6 – 14 of Algorithm 3 will identify all $\omega \in B$ satisfying (15) we can adapt the proof of Lemma 6 on page 72 of [19]. Choose any $\omega \in B$ you like. Lemmas 3 and 5 from [19] together with (20) above ensure that both

$$(21) \quad \begin{aligned} |E_j - \hat{f}_\omega| &\leq \left| E_j - (\mathcal{G}_{\lambda, K} \tilde{\psi} \mathbf{A})_j \right| + \left| (\mathcal{G}_{\lambda, K} \tilde{\psi} \mathbf{A})_j - \hat{f}_\omega \right| \\ &\leq \|\mathbf{n}\|_\infty + \frac{\epsilon \cdot \|\hat{\mathbf{f}} - \hat{\mathbf{f}}_{(s/\epsilon)}^{\text{opt}}\|_1}{s} + \|\hat{f} - \hat{f}|_B\|_1 \end{aligned}$$

and

$$\begin{aligned}
|E_{j'} - \widehat{f}_\omega| &\leq \left| E_{j'} - \left(\mathcal{E}_{s_1, K} \tilde{\psi} \mathbf{A} \right)_{j'} \right| + \left| \left(\mathcal{E}_{s_1, K} \tilde{\psi} \mathbf{A} \right)_{j'} - \widehat{f}_\omega \right| \\
(22) \qquad &\leq \|\mathbf{n}\|_\infty + \frac{\epsilon \cdot \left\| \widehat{\mathbf{f}} - \widehat{\mathbf{f}}_{(s/\epsilon)}^{\text{opt}} \right\|_1}{s} + \left\| \widehat{f} - \widehat{f}|_B \right\|_1
\end{aligned}$$

hold for more than half of the j and j' -indexes that Algorithm 3 uses to approximate \widehat{f}_ω . The rest of the proof of Lemma 6 now follows exactly as in [19] after the δ at the top of page 73 is redefined to be $\delta := \frac{\epsilon \cdot \left\| \widehat{\mathbf{f}} - \widehat{\mathbf{f}}_{(s/\epsilon)}^{\text{opt}} \right\|_1}{s} + \left\| \widehat{f} - \widehat{f}|_B \right\|_1 + \|\mathbf{n}\|_\infty$, each $\left(\mathcal{G}_{\lambda, K} \tilde{\psi} \mathbf{A} \right)_j$ entry is replaced by E_j , and each $\left(\mathcal{E}_{s_1, K} \tilde{\psi} \mathbf{A} \right)_{j'}$ entry is replaced by $E_{j'}$.

Similarly, to show that lines 15 – 18 of Algorithm 3 will produce an estimate $z_\omega \in \mathbb{C}$ satisfying (16) for every $\omega \in S$ one can simply modify the first few lines of the proof of Theorem 7 in Appendix F of [19]. In particular, one can redefine δ as above, replace the appearance of each $\left(\mathcal{G}_{\lambda, K} \tilde{\psi} \mathbf{A} \right)_j$ entry by E_j , and then use (21). The bounds on the runtime follow from the last paragraph of the proof of Theorem 7 in Appendix F of [19] with no required changes. To finish, we note that the second paragraph of the lemma above follows from a completely analogous modification of the proof of Corollary 4 in Appendix G of [19]. □

B.1. Proof of Theorem 2. To get the first paragraph of Theorem 2 one can simply utilize the proof of Theorem 7 exactly as it is written in Appendix F of [19] after redefining δ as above, and then replacing the appearance of each $\left(\mathcal{G}_{\lambda, K} \tilde{\psi} \mathbf{A} \right)_j$ entry with its approximation E_j . Once this has been done, equation (42) in the proof of Theorem 7 can then be taken as a consequence of Lemma 4 above. In addition, all references to Lemma 6 of [19] in the proof can then also be replaced with appeals to Lemma 4 above. To finish, the proof of Corollary 4 in Appendix G of [19] can now be modified in a completely analogous fashion in order to prove the second paragraph of Theorem 2.

ACKNOWLEDGEMENTS

M.A. Iwen, R. Zhang, and S. Merhi were all supported in part by NSF DMS-1416752. The authors would like to thank Aditya Viswanathan for helpful comments and feedback on the first draft of the paper.