A sparse multidimensional FFT for real positive vectors*

Letourneau, Pierre-David[†] Langston, M. Harper Meister, Benoit
Lethin, Richard [‡]

July 20, 2016

arXiv:1604.06682v4 [cs.DS] 19 Jul 2016

Abstract

We present a sparse multidimensional FFT (sMFFT) randomized algorithm for positive real vectors. The algorithm works in any fixed dimension, requires an (almost)-optimal number of samples ($\mathcal{O}(R\log(N))$) and runs in $\mathcal{O}(R\log(N))$ complexity (where N is the total size of the vector in d dimensions and R is the number of nonzeros) which we claim is optimal (up to first order). It is stable to noise, exhibits an exponentially small probability of failure and is generalizable to general complex vectors.

*This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA). The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Patent Pending. (U.S. Patent Application No. 62/286,732)

[†]Corresponding author: letourneau@reservoir.com

[‡]Reservoir Labs, 632 Broadway Suite 803, New York, NY 10012

1 Introduction

The Fast Fourier Transform (FFT) algorithm reduces the computational cost of computing the Discrete Fourier Transform (DFT) of a general complex N-vector from $\mathcal{O}(N^2)$ to $\mathcal{O}(N\log(N))$. Since its popularization in the 1960s [8], the FFT algorithm has played a crucial role in multiple areas including scientific computing [9], signal processing [27] and computer science [10]. Such complexity has been shown to be optimal in the general case [30]. In more restricted cases however, such as when the vector to be recovered is sparse, it is possible to significantly improve on the latter.

Indeed, the past decade has seen the design and study of various algorithms that can compute the DFT of sparse vectors using significantly less time and measurements than traditionally required [1–3,5,11–21,23,24,28,32–34]. That is, if f is an $N\times 1$ vector corresponding to the DFT of an $N\times 1$ vector $\hat{f}$ containing at most R nonzero elements, it is possible to recover $\hat{f}$ using significantly fewer samples than the traditional "Nyquist rate" ($\ll \mathcal{O}(N)$) and in computational complexity much lower than that of the FFT ($\ll \mathcal{O}(N\log(N))$).

These schemes are generally referred to as "sparse Fast Fourier Transform" (sFFT) algorithms, and they fall within two main categories: deterministic [1,2,20,21,24] or randomized [2,11–13,15–19,21,28]. Of the two, randomized algorithms have had the most success in practice thus far; although deterministic algorithms do exhibit polylogarithmic complexity in $\log(N)$ and R (i.e., a computational cost $\sim C_{\text{det}}\,R\,\log^a(N)$), the algorithmic constant C_{det} is so large that they are only competitive when N is impractically large or when the sparsity satisfies stringent conditions. On the other hand, the algorithmic constant of randomized sparse FFTs, C_{rand}, is in general much smaller. This constant, however, depends on a parameter not present in a deterministic context: the probability of failure p in recovering the actual solution; i.e., $C_{\text{rand}} \sim C_{\text{rand}}(p)$. Sparse FFT algorithms can further be split into a few more categories: 1D versus multidimensional (d), and exact versus noisy measurements. In particular, this paper is interested solely in cases that are stable to noise as encountered in most applications.

Table 1 summarizes the properties of the most recent sparse FFT algorithms according to the most important characteristics. Note that all scalings are reported to first order, that we neglect any $\log(\log(\cdot))$ (or slower) factor throughout the paper, and that the behavior of the algorithmic constant is only reported when it is available.

Reference	Time	Samples	N	Randomization	dimension (d)	Comment
[19]	$\mathcal{O}\left(R\log^2(N)\right)$	$\mathcal{O}\left(R\log(N)\right)$	2^L	Random $C\sim p^{-2}$	$1D$	accuracy ϵ, $C\sim\epsilon^{-1}$ No implementation
[17]	$\mathcal{O}\left(N\log^3(N)\right)$	$\mathcal{O}\left(R\log(N)\right)$	2^L	Random $p\sim\frac{1}{N^{\mathcal{O}\left(\frac{R}{\gamma}\right)}}$	Any $C\sim d^d$	
[11]	$\mathcal{O}\left(R\log^2(N)\right)$	$\mathcal{O}\left(R\log(N)\right)$	2^L	Random	$1D$, $2D$	Average case.
[13]	$\mathcal{O}(R\,\mathrm{poly}(\log(N)))$	$\mathcal{O}(R\,\mathrm{poly}(\log(N)))$	Most	Random $C\sim\log(p)$	Any $C\sim 2^{\mathcal{O}(d)}$	accuracy ϵ, $C\sim\epsilon^{-1}$
[18]	$\mathcal{O}(R\log^{d+2}(N)$	$\mathcal{O}(R\log(N)$	2^L	Random $p\sim\frac{1}{\log^2(N)}$	Any $C\sim 2^{\mathcal{O}(d^2)}$	
[33]	$\mathcal{O}(R\log(N))$	$\mathcal{O}(R)$	2^L	Deterministic	$1D$	a priori knowledge of sparsity. Support lies in cyclic interval.
[34]	$\mathcal{O}(R\log(N))$	$\mathcal{O}(R)$	2^L	Deterministic	$1D$	Real positive vector. Support lies in cyclic interval.
[31]	$\mathcal{O}(R\log^4(N))$	$\mathcal{O}(R\log^3(N))$	Most	Random $p\sim\frac{1}{\#\text{samples}}$	$2D$	Theory for uniformly random support only
[22]	$\mathcal{O}(R\log^{d+3}(N))$	$\mathcal{O}(R\log(N))$	2^L	Random $p\sim\frac{1}{\log(N)}$	Any $C\sim 2^{\mathcal{O}(d^2)}$	accuracy ϵ, $C\sim\epsilon^{-1}$
This paper	$\mathcal{O}\left(R\log\left(N\right)\right)$	$\mathcal{O}\left(R\log\left(N\right)\right)$	Most	Random $C\sim\log(p)$	Any $C\sim\mathcal{O}(1)$	accuracy ϵ, $C\sim\log(\epsilon)$ No implementation

Table 1: Computational characteristics of recent sparse FFT algorithms. C is the algorithmic constant, d is the ambient dimension and p is the probability of failure.

In this paper, we introduce a sparse multidimensional FFT (sMFFT) with properties as shown in the final row of Table 1. The sMFFT algorithm we present in this paper has the following properties:

- $\mathcal{O}\left(R\sqrt{\log(R)}\log\left(N\right)\right)$ samples and $\mathcal{O}\left(R\log^{\frac{3}{2}}(R)\log\left(N\right)\right)$ computational complexity (Section 3).

- Works in any dimension d with algorithmic constant *independent* of dimension, i.e., dependence of the form $\log(N) \sim \log(M^d) = d\log(M)$ only (Section 4).
- Exponentially small probability of failure and most desirable behavior among all modern randomized sparse FFTs ($C_{\text{sMFFT}} \sim \log(p)$, Section 3).
- Stable to noise (Section 3.3).
- Works for any positive integer N (Section 3).
- Works for positive real vectors only[1].

In this sense, our sMFFT algorithm matches the optimal sampling complexity [15] (within a $\log(\log(N))^{-1}$ factor), and the best time complexity achieved so far [33, 34], which we conjecture is in fact optimal.

Conjecture. Consider the family $\mathcal{F}_d(R, M)$ of a sparse d-dimensional signal possessing R nonzero elements distributed arbitrarily within a a regular lattice possessing M points per dimensions ($N = M^d$ points total). Then, the sampling and computational complexity for recovering each element of $\mathcal{F}_d(R, M)$ from the knowledge of noisy Fourier samples is bounded below by $\mathcal{O}(R\log(M^d))$ and $\mathcal{O}(R\log(R)\log(M^d))$ respectively.

Rationale. Sampling complexity has already been treated [15] and the bound is known to be optimal. Now, for signal of size R, one must use the FFT which has a provable lower bound of $\mathcal{O}(R\log(R))$ [30]. Since sparse FFTs are a generalization, one cannot expect to achieve a better complexity. In addition, it is reasonable to expect that any reconstruction cost will exhibit some dependency on N^d; i.e., $\sim \mathcal{O}(R\log(R)\, f(N^d))$, and $f(N^d) = \log(N^d)$ is a very weak (and achievable) dependence. In this sense, it is reasonable to claim that $\mathcal{O}(R\log(R)\log(N^d))$ should be optimal.

In this sense, *our algorithm possesses optimal sampling, and optimal computational complexity within a factor* $\sqrt{\log(R)}$. We believe, however, that this factor is an artifact of the theory (see Remark 1, Appendix A.1), and that the parameters can be chosen such that the sampling and computational complexity are $\mathcal{O}(R\log(N))$ and $\mathcal{O}(R\log(R)\log(N))$, respectively. Also note that the algorithms introduced in [33, 34] also reach this bound. However, both algorithms were designed solely for the 1D case, requiring the support of the sparse vector be contained in some *cyclic interval* of size $\mathcal{O}(R)$, an hypothesis that is extremely restrictive. In opposition, our sMFFT algorithm presented here makes *no assumption* on the support of the sparse vector other than being sparse. In addition, our algorithm is the first multidimensional version of a sparse FFT which exhibit such scaling per regards to dimension. Indeed, most previous endeavors were targeted at the 1D sparse FFT, whereas the only currently-existing multidimensional versions [13, 17, 18, 22] have a cost that scales at least exponentially with dimension. As for guarantees of recovering the correct solution (with probability $\geq 1 - p$), it is seen from Table 1 that the dependence of the algorithmic constant on the probability of failure is often quite unfavorable compared to that of our scheme ($C \sim \log(p)$). Finally, our sMFFT algorithm does currently share with [34] the characteristic of being designed for real positive vectors only.[2] However, in many applications this is a valid hypothesis (e.g., radar imaging [26], image processing [25]).

This document is structured as follows: in Section 2, we introduce the notation and a quantitative description of the problem. In Section 3, we describe the algorithm in the one-dimensional case. The case of noisy data is discussed in Section 3.3. We describe how to go from one dimension to multiple dimensions in Section 4. Finally, numerical results are provided in Section 5. All proofs can be found in Appendix A, and a discussion of the generalization to complex vectors can be found in Appendix B.

2 Notation and description of the problem

In this section, we introduce the notation used throughout the remainder of the paper. Unless otherwise stated, the (1D) function $f(x)$ will be assumed to take the form,

$$f(x) = \sum_{j=0}^{N-1} e^{-2\pi i\, x\, j}\, \hat{f}_j\,, \tag{1}$$

[1] General complex vectors are treated in a sister paper: *"A sparse multidimensional fast Fourier transform for general complex vectors."* The work presented here is a precursor to the latter, or a faster alternative when the positivity hypotheses are met. See also Appendix B.

[2] ibid.

for some finite $0 < N \in \mathbb{N}$, which represents the total number of unknowns. Furthermore, it is assumed that the vector $\hat{f}$ has real and positive elements as well as sparsity level smaller than or equal to $R < N$. That is, if we define the *support* of $\hat{f}$ as: $\mathcal{S} := \left\{ n \in \{0, 1, ..., N-1\} : |\hat{f}_n| \neq 0 \right\}$, then $0 \leq \#\mathcal{S} \leq R < N < \infty$, and

$$\hat{f}_j \begin{cases} > 0 & \text{if } j \in \mathcal{S} \\ = 0 & \text{else,} \end{cases} \tag{2}$$

where # indicates the cardinality of a set. In particular, we are interested in the case where $R \ll N$. We shall denote by $\mathcal{F}$ the Fourier transform (and $\mathcal{F}^*$ its inverse/adjoint); i.e., $\mathcal{F}\left[\hat{f}(\xi)\right](x) = \int_{\mathbb{R}^d} e^{-2\pi i\, x\cdot\xi}\, \hat{f}(\xi)\, \mathrm{d}\xi$, where d represents the ambient dimension. The size-N Discrete Fourier Transform (DFT) is defined as

$$f_{n;N} = \left[F_N\, \hat{f}\right]_n = \sum_{j=0}^{N-1} e^{-2\pi i \frac{n\,j}{N}}\, \hat{f}_j,\ n = 0, 1, ..., N-1. \tag{3}$$

The problem can now be stated as follows: let $\hat{f}$ be some vector satisfying Eq.(2) and assume samples take the form of a "clean" signal plus some additive noise: $\hat{f} + \nu$, where the ν is "small". Then, an approximation $\hat{f}$ is recovered within the noise level from as few measurements and in as few computational steps as possible. We refer to the case where $||\nu|| = 0$ as the *noiseless* case; otherwise, the data is noisy. Finally, we assume that one has access to a decomposition of N, the total size of the spectrum, of the form: $N = K \prod_{i=1}^{P} \rho_i$, where $2 \leq \rho_i \leq \rho = \mathcal{O}(1)$ for $i = 1, 2, ..., P$, $K = \mathcal{O}(\rho R \sqrt{\log(R)})$ and $P = \mathcal{O}\left(\log\left(\frac{N}{K}\right)\right)$. If no such decomposition exists, it is always possible to slightly increase the value of N and "pad with zeros" such that the resulting integer possesses such a decomposition. Because the dependence on N is logarithmic, the impact on computational cost is negligible.

3 A sparse FFT in 1D

In this section, we describe a fast way to compute the one-dimensional DFT of a bandlimited and periodic function $f(x)$ satisfying Eq.(1)-(2). Our approach to this problem can be broken into two separate steps: in the first step, the support $\mathcal{S}$ of the vector $\hat{f}$ is recovered, and in the second step, the nonzero values of $\hat{f}$ are computed using the knowledge of the recovered support. We describe the algorithm in the noiseless case in this section, followed by a discussion of its stability to noise in Section 3.3. Pseudo-code is provided in Algorithms 1-4.

Algorithm 1 1DSFFT(R, N, p)

```
1: Let μ, Δ and η be estimates for min_{j∈S} |f̂_j|, ||f̂||_∞/μ and the noise √N ||ν||_2 respectively.
2: (In the noiseless case, let η be the desired level of accuracy)
3: S ← FIND_SUPPORT(R, N, p, μ, Δ)
4: f̂ ← COMPUTE_VALUES(S, R, N, p, μ, Δ, η)
5: Output: f̂, S.
```

Finding the support. For the remainder of this section, refer to the example in Figure 1. From a high-level perspective, our support-finding scheme uses three major ingredients: 1)sub-sampling, 2)shuffling and 3)low-pass filtering. Sub-sampling reduces the size of the problem to a manageable level, but leads to aliasing. Nonetheless, when Eq.(2) is satisfied, an aliased Fourier coefficient is nonzero *if and only if* its corresponding aliased lattice contains an element of the true support (note that positivity is crucial here to avoid cancellation). This provides a useful criterion to discriminate between elements that belong to the support and elements that do not.

For the example in Figure 1, let $k, N, M_k \in \mathbb{N}$, $0 < \alpha < 1$ and $\mathcal{S}_k, \mathcal{W}_k, \mathcal{M}_k \subset \{0, 1, ..., N-1\}$. Then,

- the *aliased support* $\mathcal{S}_k$ at step k corresponds to the indices of the elements of the true support $\mathcal{S}$ modulo M_k;
- the *working support* at step k corresponds to the set $\mathcal{W}_k := \{0, 1, ..., M_k - 1\}$;

- a *candidate support* $\mathcal{M}_k$ at step k is any set satisfying $\mathcal{S}_k \subset \mathcal{M}_k \subset \mathcal{W}_k$ of size $\mathcal{O}(\rho R\sqrt{\log(R)})$.

Line **0)** (Figure 1) represents a lattice (thin tickmarks) of size: $N = 40 = 5\prod_{i=1}^{3} 2 = K\prod_{i=1}^{P}\rho_i$, which contains only 3 positive frequencies (black dots; $\mathcal{S} = \{1, 23, 35\}$). In the beginning, (step $k = 0$) only the fact that $\mathcal{S} \subset \{0, 1, ..., N-1\}$ is known. The first step ($k = 1$) is performed as follows: letting $M_1 = \frac{N}{\prod_{i=2}^{P}\rho_i} = \rho_1 K = \mathcal{O}(R\sqrt{\log(R)})$, sample $f(x)$ at $x_{n_1\prod_{i=2}^{P}\rho_i;\,N} = \frac{n_1\prod_{i=2}^{P}\rho_i}{N} = \frac{n_1}{M_1} = x_{n_1;\,M_1}$ to obtain

$$f_{n_1\prod_{i=2}^{P}\rho_i;\,N} = \sum_{j=0}^{N-1} e^{-2\pi i\,\frac{n_1\prod_{i=2}^{P}\rho_i\,j}{N}}\hat{f}_j = \sum_{l=0}^{M_1-1} e^{-2\pi i\,\frac{n_1 l}{M_1}}\left(\sum_{j:j \bmod M_1 = l}\hat{f}_j\right) = \sum_{l=0}^{M_1-1} e^{-2\pi i\,\frac{n_1 l}{M_1}}\hat{f}_j^{(1)} = f_{n_1;\,M_1} \tag{4}$$

for $n_1 \in \mathcal{M}_1 := \{0, 1, ..., M_1 - 1\}$ defined as the candidate support in the first step. The samples correspond to a DFT of size M_1 of the vector $\hat{f}^{(1)}$ with entries that are an *aliased version* of those of the original vector $\hat{f}$, as described previously. These can be computed through the FFT in order $\mathcal{O}(M_1\log(M_1)) = \mathcal{O}(R\log^{\frac{3}{2}}(R))$. In this first step, it is further possible to rapidly identify the aliased support $\mathcal{S}_1$ from the knowledge of $\hat{f}^{(1)}$ since the former correspond to the set: $\{l \in \{0, 1, ..., M_1 - 1\} : \hat{f}_l^{(1)} \neq 0\}$ (due to $\hat{f}_l^{(1)} := \sum_{j:j \bmod M_1 = l}\hat{f}_j > 0 \Leftrightarrow l \in \mathcal{S}_1$ by Eq.(2)). In our example, $M_1 = \rho_1 K = 2\cdot 5 = 10$, which leads to

$$\mathcal{S}_1 = \{1 \bmod 10,\ 23 \bmod 10,\ 35 \bmod 10\} = \{1, 3, 5\} = \{l \in \{0, 1, ..., 9\} : \hat{f}_l^{(1)} \neq 0\};\ \ \mathcal{W}_1 = \mathcal{M}_1 = \{0, 1, ..., 9\}.$$

This is shown on line **1)** of Figure 1. For this first step, the working support $\mathcal{W}_1$ is equal to the candidate support $\mathcal{M}_1$.

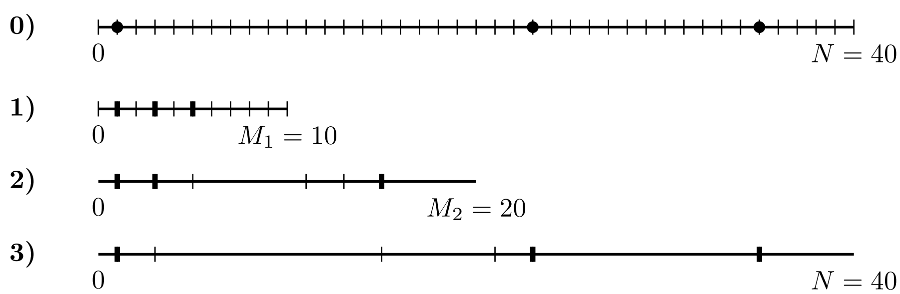

Figure 1: Computing the support $\mathcal{S}$. Line **0)**: Initialization; (unknown) elements of $\mathcal{S}$ correspond to black dots and lie in the grid $\{0, 1, ..., N-1\}$. Line **1)**: First step; elements of the candidate support $\mathcal{M}_1$ are represented by thin tickmarks and those of the aliased support $\mathcal{S}_1$ by thick tickmarks. $\mathcal{S}_1$ is a subset of $\mathcal{M}_1$ and both lie in the working support $\{0, 1, ..., M_1 - 1\}$. Line **2)**: Second step; elements of the candidate support $\mathcal{M}_2$ correspond to thin tickmarks and are obtained through de-aliasing of $\mathcal{S}_1$. Elements of the aliased support $\mathcal{S}_2$ correspond to thick tickmarks. Both lie in the working support $\{0, 1, ..., M_2 - 1\}$. M_2 is a constant factor of M_1. Line **3)**: The final step correspond to the step when the working is equal to $\{0, 1, ..., N-1\}$.

Then, proceed to the next step ($k = 2$) as follows: let $M_2 = \rho_2 M_1 = K\prod_{i=1}^{2}\rho_i = 5\cdot 2^2 = 20$ and consider the samples

$$f_{n_2\prod_{i=3}^{P}\rho_i;\,N} = \sum_{l=0}^{M_2-1} e^{-2\pi i\,\frac{n_2 l}{M_2}}\left(\sum_{j:j \bmod M_2 = l}\hat{f}_j\right) = \sum_{l=0}^{M_2-1} e^{-2\pi i\,\frac{n_2 l}{M_2}}\hat{f}_l^{(2)} = f_{n_2;\,M_2}$$

for $n_2 = 0, 1, ..., M_2 - 1$ as before. Here however, knowledge of $\mathcal{S}_1$ is incorporated. Indeed, since M_2 is a multiple of M_1, it follows upon close examination that: $\mathcal{S}_2 \subset \cup_{k=0}^{\rho_1 - 1}(\mathcal{S}_1 + kM_1) := \mathcal{M}_2$. That is, the set $\mathcal{M}_2$, defined as the union of $\rho_1 = \mathcal{O}(1)$ translated copies of $\mathcal{S}_1$, must itself contain $\mathcal{S}_2$. Furthermore, it is of size $\mathcal{O}(\rho_1\,\#\mathcal{S}_1) = \mathcal{O}(\rho R)$ by construction. It is thus a proper candidate support (by definition). In our example, one obtains

$$\cup_{k=0}^{1}(\mathcal{S}_1 + kM_1) = \{1, 3, 5\} \cup \{1 + 10, 3 + 10, 5 + 10\} = \{1, 3, 5, 11, 13, 15\} = \mathcal{M}_2,$$

which contains the aliased support: $\mathcal{S}_2 = \{1 \bmod 20,\, 23 \bmod 20,\, 35 \bmod 20\} = \{1, 3, 15\}$ as shown on on line **2)** of Figure 1. The working support becomes $\mathcal{W}_2 := \{0, 1, ..., 19\}$. Once again, it is possible to recover $\mathcal{S}_2$ by leveraging the fact that $\{l \in \{0, 1, ..., M_2 - 1\} : \hat{f}_l^{(2)} \neq 0\} = \mathcal{S}_2$. Here however, the cost is higher since computing $\hat{f}^{(2)}$ involves performing an FFT of size $M_2 = 20$. Continuing in this fashion, the cost will increase exponentially with k, so additional steps are required to contain the cost. Such steps involve a special kind of shuffling as well as a filtering of the samples followed by an FFT, and we describe this in detail below. All together, $\mathcal{S}_k$ can now be recovered from the knowledge of $\mathcal{M}_k$ at any step k using merely $\mathcal{O}(R\sqrt{\log(R)})$ samples and $\mathcal{O}(R \log^{\frac{3}{2}}(R))$ computations.

Following the rapid recovery of $\mathcal{S}_2$, proceed in a similar fashion until $\mathcal{W}_k := \{0, 1, ..., N-1\}$ ($P = \mathcal{O}\left(\log\left(\frac{N}{R}\right)\right)$ times) at which point $\mathcal{S}_k = \mathcal{S}$. Throughout this process, the size of the aliased support $\mathcal{S}_k$ and candidate support $\mathcal{M}_k$ remain of order $\mathcal{O}(R)$ while the size of the working support increases exponentially fast; i.e., $\#\mathcal{W}_k = \mathcal{O}(K \prod_{i=1}^{k} \rho_i) \geq 2^k \cdot R$. Since going from step k to step $k+1$ (computing $\mathcal{S}_k$ from $\mathcal{M}_k$ and de-aliasing) has the scaling just described, this implies the claimed cost. This therefore implies $\mathcal{O}(\log(N))$ "dealiasing" steps and a total cost of $\mathcal{O}\left(R\sqrt{\log(R)}\log(N)\right)$ samples and $\mathcal{O}\left(R \log^{\frac{3}{2}}(R) \log\left(\frac{N}{R}\right)\right)$ computational steps to identify $\mathcal{S}$. The steps of this support-recovery algorithm are described in Algorithm 2, the correctness of which is guaranteed by the following proposition:

Proposition 1. *In the noiseless case, Algorithm 2 outputs $\mathcal{S}$, the support of the vector $\hat{f}$ satisfying Eq.*(2)*, with probability at least* $(1-p)$.

Proof. Refer to Algorithm 2 and Proposition 5 (Appendix A.1). □

From the knowledge of $\mathcal{S}$, it is possible to recover the actual values of $\hat{f}$ rapidly and with few samples. This is the second major step of the sMFFT which we describe below in Section 3.1.

Algorithm 2 FIND_SUPPORT($R, \tilde{N}, p, \mu, \Delta$)

Pick $2 \leq \rho$ and $0 < \delta \ll 1$.
Let $\alpha = \frac{1}{\rho}$, $K = \frac{\max\{8, \frac{2}{\alpha}\}}{\pi} R \sqrt{\log\left(\frac{2R\Delta}{\delta}\right)\log\left(\frac{2\Delta}{\delta}\right)}$ and choose $N = K \prod_{i=1}^{P} \rho_i \geq \tilde{N}$ where $2 \leq \rho_i \leq \rho\ \forall\ i$
Let $M_1 = K$ and $\mathcal{M}_1 := \{0, 1, ..., M_1 - 1\}$.
for k from 1 to P **do**
    $\mathcal{S}_k \leftarrow$ FIND_ALIASED_SUPPORT($\mathcal{M}_k, M_k, K, \alpha, p, \delta, \mu, \Delta$)
    $\mathcal{M}_{k+1} := \cup_{m=0}^{\rho_k - 1} (\mathcal{S}_k + m M_k)$.
    $M_{k+1} := \rho_k M_k$
    $\alpha \leftarrow \frac{1}{\rho_i}$
end for
Output: $\mathcal{S}_P$.

3.1 Rapid recovery of $\mathcal{S}_k$ from knowledge of $\mathcal{M}_k$.

Details are given here as to how to solve the problem of rapidly recovering the aliased support $\mathcal{S}_k$ from the knowledge of a candidate support $\mathcal{M}_k$. Before proceeding, a few definitions are introduced.

Definition 1. *Let $1 \leq K \leq M \in \mathbb{N}$. Then, define the set $\mathcal{A}(K; M)$ as,*

$$\mathcal{A}(K; M) := \left\{ m \in \{0, 1, ..., M-1\} \,:\, m \leq \frac{K}{2} \text{ or } |m - M| < \frac{K}{2} \right\}$$

Definition 2. *Let $0 < M \in \mathbb{N}$. Then, the set $\mathcal{Q}(M) := \{q \in [0, M) \cap \mathbb{Z} : q \perp M\}$, where the symbol $\perp$ between two integers indicates they are coprime.*

Algorithm 3 shows how to solve the aliased support recovery problem rapidly; correctness is guaranteed by Proposition 2, which relies on Proposition 5 (Appendix A.1). Proposition 5 states that if the elements of an aliased vector of size M_k with aliased support $\mathcal{S}_k$ containing at most R nonzeros are shuffled (according to appropriate random permutation) and subsequently convoluted with a (low-frequency) Gaussian, then the probability that the resulting value at a location $m \in \mathcal{S}_k^c$ is of order $\mathcal{O}(1)$ is small. If $m \in \mathcal{S}_k$, the value at m is of order $\mathcal{O}(1)$ with probability 1. This realization allows us to develop an *efficient statistical test* to identify $\mathcal{S}_k$ from the knowledge of $\mathcal{M}_k$. The process is shown schematically in Figure 2. Specifically, the four following steps are performed: 1) permute samples randomly, 2) apply a diagonal Gaussian filter, 3) compute a small FFT, 4) eliminate elements that do not belong to aliased support. To help the reader better understand, we once again proceed through an example. To begin, assume $M_k = 40$, and

$$\mathcal{S}_k = \{1, 23, 35\},\ \mathcal{M}_k = \{1, 3, 15, 21, 23, 35\},\ \mathcal{W}_k = \{1, 2, ..., 39\}$$

as in step $k = 3$ of the previous section. Refer to Figure 2. The first step is to randomly shuffle the elements of $\mathcal{M}_k$ within $\mathcal{W}_k$ by applying a permutation operator $\Pi_Q(\cdot)$ in sample space

$$\Pi_Q\left(f_{n;\,M_k}\right) = f_{(nQ) \bmod M_k;\,M_k} \tag{5}$$

for some integer $Q \in \mathcal{Q}(M_k)$. By Lemma 3 (AppendixA.1), this is equivalent to shuffling in frequency space as: $\hat{f}_l^{(k)} \rightarrow \hat{f}^{(k)}_{(l[Q]^{-1}_{M_k}) \bmod M_k}$ ($[Q]^{-1}_{M_k}$ being the unique inverse of Q modulo M_k). Furthermore, Lemma 4 (Appendix A.1) shows that if Q is chosen uniformly at random within $\mathcal{Q}(M_k)$, the mapped elements of the candidate support $\mathcal{M}_k$ will be uniformly distributed within the working support $\mathcal{W}_k$,

$$\mathbb{P}\left(\left|(i[Q]^{-1}_{M_k}) \bmod M_k - (j[Q]^{-1}_{M_k}) \bmod M_k\right| \leq C \,\middle|\, i \neq j\right) \leq \mathcal{O}\left(\frac{C}{M_k}\right).$$

For the illustrative example, assume $[Q]^{-1}_{M_k} = 13$. The sets $\mathcal{S}_k$ and $\mathcal{M}_k$ are then mapped by $\Pi_Q(\cdot)$ to (line **B)**, Figure 2).

$$\begin{aligned}
\mathcal{S}_k^{\text{shuffled}} &= \{(1 \cdot 13) \bmod 40, (23 \cdot 13) \bmod 40,\ (35 \cdot 13) \bmod 40\} = \{13, 15, 19\} \\
\mathcal{M}_k^{\text{shuffled}} &= \{(1 \cdot 13) \bmod 40,\ (3 \cdot 13) \bmod 40,\ (15 \cdot 13) \bmod 40,\ (21 \cdot 13) \bmod 40,\ (23 \cdot 13) \bmod 40,\ (35 \cdot 13) \bmod 40\} \\
&= \{13, 15, 19, 33, 35, 39\}
\end{aligned}$$

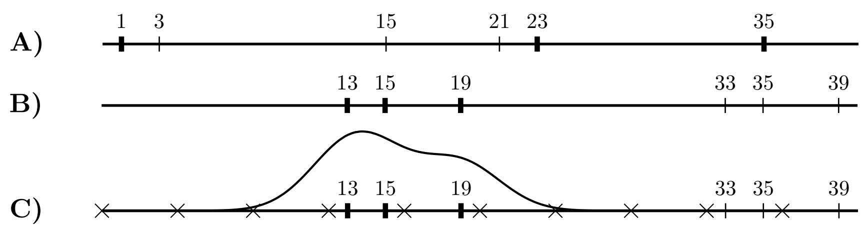

Figure 2: Finding the aliased support $\mathcal{S}_k$ from knowledge of $\mathcal{M}_k$ (line **A)**). First, indices are shuffled in sample space leading to a shuffling in frequency space (line **B)**). A Gaussian filter is applied followed by a small FFT (line **C)**) on a grid G. The points of $\mathcal{M}_k$ for which the value of the result of the last step at their closest neighbor in G is small are discarded leaving only the aliased support $\mathcal{S}_k$.

This step is followed by the application of a diagonal Gaussian filtering operator $\Psi_\sigma(\cdot)$ having elements

$$\frac{1}{M_k}\, g_\sigma\left(\frac{m}{M_k}\right) = \frac{\sqrt{\pi}\sigma}{M_k} \sum_{h \in \mathbb{Z}} e^{-\pi^2\sigma^2\left(\frac{m+hM}{M}\right)^2} \tag{6}$$

in sample space (step 2). By the properties of the Fourier transform, this is equivalent to a convolution in frequency space (line **C)**, Figure 2), implying the equality:

$$\left[\Psi_\sigma\left(\Pi_Q\left(f_{n;\,M_k}\right)\right)\right](\xi) = \mathcal{F}^*\left[\sum_{j \in \mathcal{S}_k} \hat{f}_j^{(k)}\, e^{-\frac{\left|x - \left(j[Q]^{-1}_{M_k} \bmod M_k\right)\right|^2}{\sigma^2}}\right](\xi). \tag{7}$$

The function is now *bandlimited* (with bandwidth of order $\mathcal{O}(K)$ thanks to our choice for σ; Algorithm 3), so this expression can be discretized (samples $\times$, Figure 2) to produce our main expression,

$$\phi_n^{(k)}(Q) = F_{\mathcal{A}(K,M_k)}\left[\Psi_\sigma\left(\Pi_Q\left(f_{n;\,M_k}\right)\right)\right]_n = \frac{1}{M_k} \sum_{m \in \mathcal{A}(K,M_k)} e^{2\pi i \frac{nm}{K}}\, g_\sigma\left(\frac{m}{M_k}\right) f_{(mQ) \bmod M_k;\,M_k} \tag{8}$$

In particular, we note that if n if of the form $j\frac{M_k}{K}$ for $j = 0, ..., K-1$, the last step can be performed through a small size-K FFT. This corresponds to step 3 of the aliased support recovery algorithm. The knowledge of $\left\{\phi^{(k)}_{j\frac{M_k}{K}}\right\}_{j=0}^{K=1}$ can be used to recover $\mathcal{S}_k$ from $\mathcal{M}_k$ rapidly, seen intuitively as follows: by construction $\phi^{(k)}_n$ is "large" only if the distance between a shuffled element $\left\{l[Q]^{-1}_{M_k}) \,\mathrm{mod} M_k\right\}_{l\in\mathcal{S}_k}$ of the aliased support, and $(n[Q]^{-1}_{M_k}) \,\mathrm{mod} M_k$ is smaller than $\mathcal{O}(\sigma)$, which in turn occurs only if the distance between $\left[\left(n[Q]^{-1}_{M_k} \,\mathrm{mod} M_k\right) \frac{K}{M_k}\right] \frac{M_k}{K}$ and the shuffled elements of the aliased support is smaller than $\mathcal{O}(\sigma)$ as well. However, because of the randomness introduced by the shuffling, and because of the particular choice of σ, it can be shown (Proposition 5) that for any fixed $i \in \mathcal{M}_k$, the probability that a computed element $\phi^{(k)}_{\left[i\frac{K}{M_k}\right]\frac{M_k}{K}}$ is "large" for multiple independent trials is small if $i \in \mathcal{M}_k \cap \mathcal{S}_k^c$ and equal to 1 if $i \in \mathcal{M}_k \cap \mathcal{S}_k$. This fact allows for the construction of an efficient statistical test based on the knowledge of the quantities found in Eq.(8) to discriminate between the points of $\mathcal{M}_k \cap \mathcal{S}_k^c$ and those of $\mathcal{M}_k \cap \mathcal{S}_k$ (step 4). Such a test constitutes the core of Algorithm 3, and its correctness follows from the following proposition (Appendix A.1).

Proposition 2. *In the noiseless case, Algorithm 3 outputs $\mathcal{S}_k$, the aliased support of the vector $\hat{f}$ at step k, with probability at least $(1-p)$.*

Proof. Refer to Algorithm 3 and Proposition 5. □

As for the computational cost, the permutation and filtering (multiplication) step (1 and 2) both incur a cost of $\mathcal{O}(R\sqrt{\log(R)})$ since only the samples for which the filter is of order $\mathcal{O}(1)$ are considered (and there are $\mathcal{O}(K) = \mathcal{O}(R\sqrt{\log(R)})$ of them following our choice of σ and K). These are followed by an FFT (step 3) of size $\mathcal{O}(K)$. Finally, step 4 involves checking a simple property on each of the M_k elements of $\mathcal{M}_k$ incurring a cost of $\mathcal{O}(R)$. This is repeated $\mathcal{O}(\log(p))$ times for a probability $(1-p)$ of success. In conclusion, this implies that extracting $\mathcal{S}_k$ from $\mathcal{M}_k$ requires merely $\mathcal{O}(\log(p) R\sqrt{\log(R)})$ samples and $\mathcal{O}(\log(p) R \log^{\frac{3}{2}}(R))$ computational time for fixed p, as claimed.

Algorithm 3 FIND_ALIASED_SUPPORT($\mathcal{M}_k, M_k, K, \alpha, p, \delta, \mu, \Delta$)

1: Let $\sigma = \frac{\alpha \frac{M_k}{2R}}{\sqrt{\log\left(\frac{2R\Delta}{\delta}\right)}}$ and $L = \log_\alpha(p)$.
2: $\mathcal{S}_k \leftarrow \mathcal{M}_k$
3: **for** l **from** 1 **to** L **do**
4: Pick $Q^{(l)} \in \mathcal{Q}(M_k)$ uniformly at random.
5: Compute: $\phi^{(k)}_{j\frac{M_k}{K}}(Q^{(l)}),\ j = 0, 1, ..., K-1$ (Eq.(8)).
6: **for** $j \in \mathcal{M}_k$ **do**
7: **if** $\left|\phi^{(k)}_{\left[\left(j[Q^{(l)}]^{-1}_{M_k} \bmod M_k\right)\frac{K}{M_k}\right]\frac{M_k}{K}}(Q^{(l)})\right| < \delta\,\mu$ **then**
8: Remove j from $\mathcal{S}_k$.
9: **end if**
10: **end for**
11: **end for**
12: Output: $\mathcal{S}_k$.

3.2 Recovering values from knowledge of the support

In this section, assume a set size $\mathcal{O}(R)$ containing the support $\mathcal{S} \subset \{0, 1, 2, ..., N-1\}$ has been recovered. The values of the nonzero Fourier coefficients of $\hat{f}$ in Eq. (1) can be rapidly computed u sing this information. For this purpose, assume $f(x)$ can be sampled at locations: $\left\{\frac{qQ^{(t)} \bmod P^{(t)}}{P^{(t)}}\right\}_{q=1}^{P^{(t)}}$ for $t = 0, 1, ..., T$, and $\{P^{(t)}\}_{t=1}^T$ some random prime numbers on the order of $\mathcal{O}(R\log_R(N))$ (see Algorithm 4). It follows that

$$f^{(t)}_{qQ^{(t)} \bmod P^{(t)};\, P^{(t)}} = \sum_{j\in\mathcal{S}} e^{-2\pi i \frac{q\,((jQ^{(t)}) \bmod P^{(t)})}{P^{(t)}}} \hat{f}_j = \sum_{l=0}^{P^{(t)}-1} e^{-2\pi i \frac{q\,l}{P^{(t)}}} \left(\sum_{j\in\mathcal{S}: j[Q^{(t)}]^{-1}_{P^{(t)}} \bmod P^{(t)} = l} \hat{f}_j \right) \quad (9)$$

for $t = 0, 1, ..., T$. The outer sum is seen to be a DFT of size $P^{(t)}$ of a shuffled and aliased vector, whereas the inner sum can be expressed as the application of a binary matrix $B^{(t)}_{q,j}$ with entries

$$B^{(t)}_{q,j} = \begin{cases} 1 & \text{if } j[Q^{(t)}]^{-1}_{P^{(t)}} \bmod P^{(t)} = q \\ 0 & \text{else} \end{cases}$$

to the vector with entries' index corresponding to the support of $\hat{f}$. In particular, each such matrix is sparse with exactly $\#\mathcal{S} = \mathcal{O}(R)$ nonzero entries. Eq. (9) can further be written in matrix form as

$$[FB]\,\hat{f} = \begin{bmatrix} F^{(1)} & 0 & ... & 0 \\ 0 & F^{(2)} & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & F^{(T)} \end{bmatrix} \begin{bmatrix} B^{(1)} \\ B^{(2)} \\ ... \\ B^{(T)} \end{bmatrix} \hat{f} = \begin{bmatrix} f^{(1)} \\ f^{(2)} \\ ... \\ f^{(T)} \end{bmatrix} = f_0, \tag{10}$$

where $F^{(t)}$ is a standard DFT matrix of size $P^{(t)}$. Proposition 6 states that if $T \geq \mathcal{O}(R \log_R(N))$, then with nonzero probability $\frac{1}{T}(FB)^*(FB) = I + \mathcal{P}$, where I is the identity and $\mathcal{P}$ is a perturbation with 2-norm smaller than $\frac{1}{2}$. When this occurs, one can solve the linear system through the Neumann series, $\hat{f} = \sum_{n=0}^{\infty} \mathcal{P}^n \,(FB)^* f_0$ as performed in Algorithm 4 with correctness in Proposition 3.

Proposition 3. *(Correctness of Algorithm 4) Assume the support $\mathcal{S}$ of $\hat{f}$ is known. Then Algorithm 4 outputs an approximation to the nonzero elements of $\hat{f}$ with error bounded by η in the ℓ^2-norm, with probability greater than or equal to $1 - p$.*

Since each matrix $B^{(t)}$ contains exactly R nonzero entries, both B and B^*B can be applied in order $RT = \mathcal{O}(R \log_R(N))$ steps and performed $\mathcal{O}(\log(\eta))$ times (truncation of Neumann series). In addition, since F is a block diagonal matrix with $T = \mathcal{O}(1)$ blocks consisting of DFT matrices of size $\mathcal{O}(R \log_R(N))$, it can be applied in order $\mathcal{O}(R \log(R) \log_R(N))$ thanks to the FFT. Finally, this is performed at most $\log(p)$ times for a probability p of success. Therefore, the cost of computing the nonzero values of $\hat{f}$ is bounded by $\mathcal{O}\left(\log(p)\, R \log_R(N)\, (\log(R) + \log(\eta))\right)$ and uses at most $\mathcal{O}\left(\log(p)\, R \log_R(N)\right)$ samples as claimed.

Algorithm 4 COMPUTE_VALUES($\mathcal{S}, R, N, p, \mu, \Delta, \eta$)

Let $T = 4$, $Z = \lceil \log_{\frac{1}{2}}(\eta) \rceil$ and $L = \lceil \log_{\frac{1}{2}}(p) \rceil$.
for t **from** 1 **to** L **do**
    Pick $\{P^{(t)}\}_{t=1}^{T}$ i.i.d. uniform r.v. chosen among the set containing the smallest $4R \log_R(N)$ prime numbers greater than R.
    Sample $\left\{ f_{n \bmod P^{(t)}; P^{(t)}} \right\}_{n=0}^{P^{(t)}-1}$, $t = 1, ..., T$
    Compute $\hat{f}_0 \leftarrow (FB)^* f_0$ (Eq.(9)).
    if $||(B^*B)\,\hat{f}_0||_2 < \frac{1}{2}$ **then**
        $\hat{f} \leftarrow \sum_{n=0}^{Z} (I - B^*B)^n \hat{f}_0$
        Output: $\hat{f}$.
        Exit.
    end if
end for

3.3 Stability

As discussed previously, the theory underlying the algorithms introduced in Section 3 have been designed for vectors which are exactly sparse. In this section, we discuss the effect of noise and approximate sparsity. In fact, if the sparse vector of Fourier coefficients take the form $\hat{f} + \hat{\nu}$, where $||\hat{\nu}||_2 < \frac{\eta}{\sqrt{N}}$ for some "small" η, the sMFFT algorithm recovers the support and values of $\hat{f}$ with the same guarantees as described earlier.

Support. The most important quantity for the fast recovery of the support is Eq.(8), so in the presence of noise,

$$\phi^{(k)}_n(Q) = \frac{1}{M_k} \sum_{m \in \mathcal{A}(K, M_k)} e^{2\pi i \frac{nm}{K}}\, g_\sigma\left(\frac{m}{M_k}\right) \left(f_{(mQ) \bmod M_k;\, M_k} + \nu_{(mQ) \bmod M_k;\, M_k}\right). \tag{11}$$

The second term in this expression is the error term and can be *uniformly bounded* by the following lemma:

Lemma 1. *Assuming the noise term $\hat{\nu}$ is such that $||\hat{\nu}||_2 < \frac{\eta}{\sqrt{N}}$, the error term of the computed value in Eq.(11) is uniformly bounded by*

$$\left|\left|\psi_n^{(k)}(Q)\right|\right|_\infty = \left|\left|\frac{1}{M_k}\sum_{m\in\mathcal{A}(K,M_k)} e^{2\pi i\frac{nm}{K}}\, g_\sigma\left(\frac{m}{M_k}\right)\nu_{(mQ)\bmod M_k;\, M_k}\right|\right|_\infty < \mathcal{O}(\eta).$$

Algorithm 2 tests whether $\left|\phi^{(k)}_{\left[i\frac{K}{M_k}\right]\frac{M_k}{K}}(Q^{(l)})\right| > \delta\,\mu$ in order to discriminate between elements of the candidate and aliased supports. The presence of noise can skew this test in two ways: 1) by bringing the computed value below the threshold when $i \in \mathcal{S}_k$ or 2) by bringing the value above the threshold multiple times when $i \notin \mathcal{S}_k$. Either way, if η is small enough such that $\left|\left|\psi_n^{(k)}(Q)\right|\right|_\infty \le \frac{\delta\mu}{2}$, then $\left|\phi_n^{(k)}(Q^{(l)})\right| > \delta\,\mu - \frac{\delta\mu}{2} = \frac{\delta\mu}{2}$ by the triangular inequality and Lemma 1. Under these circumstances, it can be shown that the conclusion of Proposition 5 follows through with similar estimate, by simply replacing δ with $\frac{\delta}{2}$ in the proof.

Recovering values from knowledge of the support. It is quickly observed that the recovery of the values is a well-conditioned problem. Indeed, since $\frac{1}{T}(FB)^*(FB) = I - \mathcal{P}$, and $||\mathcal{P}||_2 \le \frac{1}{2}$ with high probability by Proposition 6, a simple argument based on the singular value decomposition produces the following corollary,

Corollary 1. *Under the hypothesis of Proposition 6, $\left(\frac{1}{T}(FB)^*(FB)\right)^{-1}$ exists, and $\left|\left|\left(\frac{1}{T}(FB)^*(FB)\right)^{-1}\right|\right|_2 \le 2$ with probability greater than or equal to $\frac{1}{2}$.*

Therefore, the output of Algorithm 4 is such that

$$||\hat{f}^{\text{sMFFT}} - \hat{f}||_2 \le \left|\left|\left(\frac{1}{T}(FB)^*(FB)\right)^{-1}\right|\right|_2 ||(FB)^*\nu||_2 \le 2||B||_2\,||\nu||_2 = \mathcal{O}(\eta),$$

which, together with Proposition 3, demonstrates the stability of Algorithm 4 in the noisy case.

4 The multi-dimensional sparse FFT

Whenever dealing with the multidimensional DFT/FFT, it is assumed that the function of interest is both periodic and bandlimited with fundamental period $[0,1)^d$, i.e., $f(x) = \sum_{j\in([0,M)\cap\mathbb{Z})^d} e^{-2\pi i\, x\cdot j}\,\hat{f}_j$ for some finite $M \in \mathbb{N}$ and $j \in \mathbb{Z}^d$ (up to some rescaling). Computing the Fourier coefficients is then equivalent to computing the d-dimensional integrals $\hat{f}_n = \int_{[0,1]^d} e^{-2\pi i\, n\cdot x}\, f(x)\,\mathrm{d}x$, and this is traditionally achieved through a "dimension-by-dimension" trapezoid rule [7, 36]

$$\hat{f}_{(j_1,j_2,...,j_d)} = \sum_{n_1=0}^{M-1}\frac{e^{2\pi i\,\frac{j_1 n_1}{M}}}{M}\left(...\left(\sum_{n_{d-1}=0}^{M-1}\frac{e^{2\pi i\,\frac{j_{d-1}n_{d-1}}{M}}}{M}\left(\sum_{n_d=0}^{M-1}\frac{e^{2\pi i\,\frac{j_d n_d}{M}}}{M} f_{(n_1,n_2,...,n_d)}\right)\right)\right). \tag{12}$$

However, Proposition 4 shows that it is also possible to re-write the d-dimensional DFT as that of a 1D function with Fourier coefficients equal to those of the original function, but with different ordering.

Proposition 4. *(Rank-1 d-dimensional DFT) Assume the function $f : [0,1)^d \to \mathbb{C}$ has form* (12)*. Then,*

$$\int_{[0,1]^d} e^{-2\pi i\, j\cdot x}\, f(x)\,\mathrm{d}x = \frac{1}{N}\sum_{n=0}^{N-1} e^{-2\pi i\, j\cdot x_n}\, f(x_n) \tag{13}$$

for all $j \in [0,M)^d \cap \mathbb{Z}^d$, where $x_n = \frac{ng \bmod N}{N}$, $g = (1, M, M^2, ..., M^{d-1})$ and $N = M^d$.

Now, Eq. (13) can be written in two different ways (due to periodicity); namely,

$$\frac{1}{N}\sum_{n=0}^{N-1} e^{-2\pi i\, j\cdot \frac{ng \bmod N}{N}}\, f\left(\frac{ng \bmod N}{N}\right) = \frac{1}{N}\sum_{n} e^{-2\pi i\, \frac{(j\cdot g)n}{N}}\, f\left(\frac{ng}{N}\right)$$

Geometrically, the left-hand side represents a quadrature rule with points $x_n = \frac{ng \bmod N}{N}$ distributed (more-or-less uniformly) in $[0,1)^d$ (Figure 3, left; grey dots). The right-hand side represents an equivalent quadrature where the points $x_n = \frac{ng}{N}$ now lie on a *line* embedded in $\mathbb{R}^d$ (Figure 3, right; grey dots). The location at which the lattice (thin black lines) intersects represents the standard multidimensional DFT samples. In short, Proposition 4 allows one to write *any* d-dimensional DFT as a one-dimensional DFT

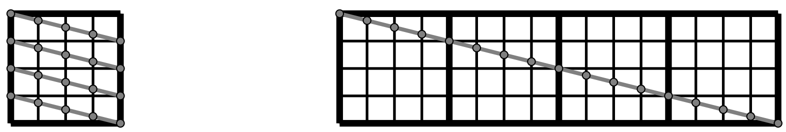

Figure 3: Geometric interpretation of rank-1 d-dimensional DFT in 2D. The thick black box represents fundamental periodic domain. The grey dots represent rank-1 discretization points. The 2D grid represents standard discretization points. Left: the rank-1 d-dimensional quadrature interpreted as a 2D discretization over the fundamental periodic region. Right: the rank-1 d-dimensional quadrature interpreted as a uniform discretization over a line in $\mathbb{R}^2$.

by picking the appropriate sample points (Proposition 4) and proceeding to a re-ordering of the Fourier coefficients through the isomorphism

$$\tilde{n} : \left\{n \in ([0,M)\cap\mathbb{Z})^D\right\} \to n\cdot g = n_0 + n_1\, M + ... + n_{D-1}\, M^{D-1} \in [0, M^D)\cap\mathbb{Z}^D.$$

In Section 5, we use this scheme for achieving our sMFFT numerical results.

5 Numerical results

We have implemented our sMFFT algorithm in `MATLAB`[3] and present a few numerical results which exhibit the claimed scaling. All simulations were carried out on a small cluster possessing 4 Intel Xeon E7-4860 v2 processors and 256GB of RAM, with the `MATLAB` flag $-$`singleCompThread` to ensure fairness through the use of a single computational thread. The numerical experiments presented here fall in two categories: 1) dependence of running time as a function of the total number of unknowns N for a fixed number of nonzero frequencies R, and 2) dependence of running time as a function of the number of nonzero frequencies R for a fixed total number of unknowns N. All experiments were carried out in three dimensions (3D) with additive Gaussian noise with variance η. The nonzero values of $\hat{f}$ were picked randomly and uniformly at random in $[0.5, 1.5]$, and the remaining parameters were set according to Table 2. All comparisons are perdormed with the `MATLAB` `fftn`$(\cdot)$ function, which uses a dimension-wise decomposition of the DFT (see Section 4) and a 1D FFT routine along each dimension. For case 1), we picked $R = 50$ nonzero frequencies

Table 2: Values of parameters required by Algorithm 1-4 and used for numerical experiments

Parameter	Description	Value (Case 1)	Value (Case 2)
N	Total number of unknowns	variable	10^8
R	Number of nonzero frequencies	50	variable
α	Gaussian filter parameter	0.15	0.15
δ	Statistical test parameter	0.1	0.1
p	Probability of failure	10^{-4}	10^{-4}
d	Ambient dimension	3	3
η	Noise level	10^{-2}	10^{-2}

distributed uniformly at random on a 3D lattice having $N^{1/3}$ elements in each dimension for different values

[3]`MATLAB` is a trademark of Mathworks

of $N \in [10^3, 10^{10}]$. The results are shown in Figure 4 (left). As can be observed, the cost of computing the DFT through the sMFFT remains more or less constant with N, whereas that the the MATLAB $\texttt{fftn}(\cdot)$ function increases linearly. This is the expected behavior and demonstrates the advantages of the sMFFT over the FFT. Also note that the largest relative ℓ^2-error observed was $9.3 \cdot 10^{-3}$ which is on the order of the noise level, as predicted by the theory.

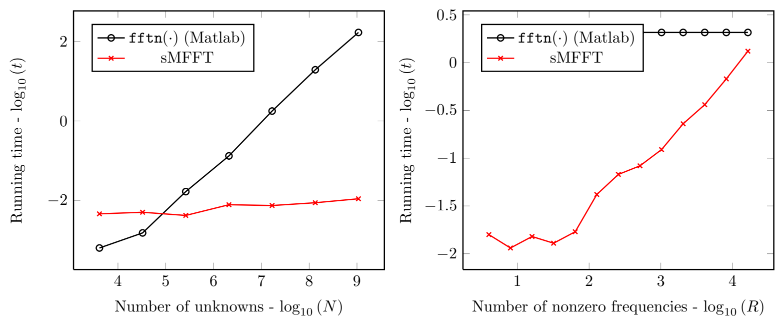

Figure 4: Left: Running time vs number of unknowns (N) for the MATLAB $\texttt{fftn}(\cdot)$ (black) and the sMFFT (red) in three dimensions (3D), with $R = 50$ nonzeros and noise $\eta = 10^{-3}$. Right: Running time vs number of nonzero frequencies (R) for the MATLAB $\texttt{fftn}(\cdot)$ (black) and the sMFFT (red) in three dimensions (3D) and for $N = 10^8$ and noise $\eta = 10^{-3}$.

For case 2), we fixed $N = \mathcal{O}(10^8)$ and proceeded to compare the sMFFT algorithm with the MATLAB $\texttt{fftn}(\cdot)$ function as before (w/ parameters found in Table 2). The results are shown in Figure 4 (right). In this case, the theory states that the sMFFT algorithm should scale quasi-linearly with the number of nonzero frequencies R. A close look shows that it is indeed the case. For this case, the largest relative ℓ^2-error observed was $1.1 \cdot 10^{-2}$, again on the order of the noise level and in agreement the theory. Finally, the cost $\texttt{fftn}(\cdot)$ function remains constant as the FFT scales like $\mathcal{O}(N \log(N))$ and is oblivious to R.

6 Conclusion

We have introduce a sparse multidimensional FFT (sMFFT) for computing the DFT of a $N \times 1$ sparse, real-positive vector (having R nonzeros) exhibiting the best scaling up to date per regards to sampling ($\mathcal{O}(R\sqrt{\log(R)}\log(N))$) and time ($\mathcal{O}(R\log^{\frac{3}{2}}(R)\log(N))$) complexities, as well as probability of success ($\mathcal{O}(\log(p))$), accuracy ($\mathcal{O}(\log(\eta))$) and dimension ($\mathcal{O}(1)$). The scheme is also stable to noise. We have provided a rigorous theoretical analysis of our approach demonstrating each claim. Finally, we have implemented our algorithm and provided numerical examples in 3D succesfully demonstrating the claimed scaling.

A Proofs

In this appendix, we present all proofs and accompanying results related to the statements presented in the main body of the work.

A.1 Proofs of Section 3

Lemma 2. *Let* $0 < Q \leq N \in \mathbb{N}$, $Q \perp N$. *Then the map,*

$$n \in \{0, 1, ..., N-1\} \to nQ \bmod N \subset \{0, 1, ..., N-1\}$$

is an isomorphism.

Proof. Since the range is discrete and a subset of the domain, it suffices to show that the map is injective. Surjectivity will then follow from the pigeon hole principle. To show injectivity, consider $i, j \in \{0, 1, ..., N-1\}$, and assume,

$$iQ \bmod N = jQ \bmod N$$

This implies (by definition) that there exists some integer p such that,

$$(i-j)Q = pN$$

so that N divides $(i-j)Q$. However, $N \perp Q$ so N must be a factor $(i-j)$. Now, i, j are restricted to $\{0, 1, ..., N-1\}$ so,

$$|i-j| < N,$$

and the only integer divisible by N that satisfies this equation is 0. Thus,

$$i - j = 0 \Leftrightarrow i = j$$

which demonstrates injectivity. □

Lemma 3. *Let* $0 < Q < M$ *be an integer coprime to* M *and,*

$$f_n = \sum_{l=0}^{M-1} e^{-2\pi i \frac{n\,l}{M}} \hat{f}_l$$

Then,

$$\frac{1}{M} \sum_{n=0}^{M-1} e^{2\pi i \frac{m\,n}{M}} f_{(nQ) \bmod M} = \hat{f}_{(m[Q]_M^{-1}) \bmod M}$$

where $0 < [Q]_M^{-1} < M$ *is the unique integer such that* $[Q]_M^{-1} Q \bmod M = 1 \bmod M$.

Proof. Consider

$$\begin{aligned}
\frac{1}{M} \sum_{n=0}^{M-1} e^{2\pi i \frac{m\,n}{M}} f_{(nQ) \bmod M} &= \sum_{l=0}^{M-1} \left(\frac{1}{M} \sum_{n=0}^{M-1} e^{2\pi i \frac{n}{M}(m - Ql \bmod M)} \right) \hat{f}_l \\
&= \sum_{l=0}^{M-1} \left(\frac{1}{M} \sum_{n=0}^{M-1} e^{2\pi i \frac{nQ}{M}\left(m[Q]_M^{-1} \bmod M - l\right)} \right) \hat{f}_l
\end{aligned}$$

However,

$$\frac{1}{M} \sum_{n=0}^{M-1} e^{2\pi i \frac{nQ}{M}\left(m[Q]_M^{-1} \bmod M - l\right)} = \frac{1}{M} \sum_{j=0}^{M-1} e^{2\pi i \frac{j}{M}\left(m[Q]_M^{-1} \bmod M - l\right)} = \delta_{m[Q]_M^{-1} \bmod M, l},$$

where the second equality follows from the fact that $m \to m[Q]_M^{-1} \bmod M$ is an isomorphism (Lemma 2). This implies that

$$\frac{1}{M}\sum_{n=0}^{M-1} e^{2\pi i\,\frac{m\,n}{M}}\, f_{(nQ)\bmod M} = \hat{f}_{(m[Q]_M^{-1})\bmod M}$$

as claimed. ☐

Lemma 4. *Let $M \in \mathbb{N}/\{0\}$ and let Q be a uniform random variable over $\mathcal{Q}(M)$ (Definition 2). Then,*

$$\mathbb{P}\left(|jQ\,\mathrm{mod}M| \leq C\right) \leq \mathcal{O}\left(\frac{C}{M}\right)$$

for all $0 < j < M$ (up to a $\log(\log(M))$ factor).

Proof. Fix $0 < j, k < M$ and let $\gamma = \gcd(j, M)$. Consider,

$$\mathbb{P}\left(jQ \bmod M = k\right) = \sum_{q\in\mathcal{Q}(M)} \mathbb{P}\left(jq \bmod M = k | Q = q\right)\,\mathbb{P}(Q = q) = \sum_{q\in\mathcal{Q}(M)} \mathbb{I}_{jq\,\mathrm{mod}M=k}(q)\,\mathbb{P}(Q = q)$$

and note that,

$$\mathbb{P}(Q = q) = \frac{1}{\#\mathcal{Q}(M)} = \frac{1}{\phi(M)} \leq \frac{e^{\zeta}\log(\log(M)) + \frac{3}{\log(\log(M))}}{M}$$

following bounds on the Euler totient function $\phi(\cdot)$ ([35]), where ζ is the Euler-Mascheroni constant, and since Q is uniformly distributed in $\mathcal{Q}(M)$. Therefore,

$$\mathbb{P}\left(jQ \bmod M = k\right) \leq \frac{e^{\zeta}\log(\log(M)) + \frac{3}{\log(\log(M))}}{M}\sum_{q}\mathbb{I}_{jq\,\mathrm{mod}M=k}(q)$$

We now show that the quantity $\sum_q \mathbb{I}_{jq\,\mathrm{mod}M=k}(q)$ is bounded above and below by,

$$\gamma - 1 \leq \sum_{q}\mathbb{I}_{jq\,\mathrm{mod}M=k}(q) \leq \gamma$$

To see this, first note that this quantity corresponds to the number of integers q which hash to the integer k through the map $q \to (jq) \bmod M$. Now, assume there exists some q such that

$$jq\,\mathrm{mod}M \equiv k, \tag{14}$$

which implies that

$$jq + iM = k \tag{15}$$

for some integer $i \in \mathbb{Z}$. This is a Diophantine equation which has infinitely many solutions if and only if $\gcd(j, M) = \gamma$ divides k ([29]). Otherwise, it has no solution. Assuming it does and (q_0, i_0) is a particular solution, all remaining solutions must take the form

$$q = q_0 + u\frac{M}{\gamma},\ i_0 - u\frac{j}{\gamma}$$

where $u \in \mathbb{Z}$. However, since $0 \leq q < M$ the number of possible solutions must be such that,

$$\gamma - 1 \leq \#\left\{q \in [0, M) : q = q_0 + u\frac{M}{\gamma}\right\} \leq \gamma.$$

which proves the claim. Thus,

$$\mathbb{P}\left(jQ \bmod M = k\right) \leq \left(e^{\zeta}\log(\log(M)) + \frac{3}{\log(\log(M))}\right)\frac{\gamma}{M}$$

We can now treat $\mathbb{P}\left(|jQ \bmod M| \leq C\right)$. Before we proceed however, recall that Eq.(14) has a solution if and only if $\gamma | k$. We then write,

$$\mathbb{P}\left(|jQ \bmod M| \leq C\right) = \sum_{0 \leq k \leq C} \mathbb{I}_{\gamma | k}(k)\, \mathbb{P}\left(nQ \bmod M = k\right)$$

from which it follows that,

$$\begin{aligned}
\mathbb{P}\left(|jQ \bmod M| \leq C\right) &\leq \left(e^{\zeta} \log(\log(M)) + \frac{3}{\log(\log(M))}\right) \frac{\gamma}{M} \sum_{0 \leq k \leq C} \mathbb{I}_{\gamma | k}(k) \\
&\leq \left(e^{\zeta} \log(\log(M)) + \frac{3}{\log(\log(M))}\right) \frac{\gamma}{M} \frac{C}{\gamma} \\
&\leq \left(e^{\zeta} \log(\log(M)) + \frac{3}{\log(\log(M))}\right) \frac{C}{M}
\end{aligned}$$

since the number of integers in $0 \leq k \leq C$ that are divisible by γ is bounded above by $\frac{C}{\gamma}$. Finally, since this holds regardless of our choice of j, this proves the desired result. □

Lemma 5. *Consider a function $f(x)$ of the form of Eq.(1) and satisfying the constraint Eq.(2). Let $0 < \sigma = \mathcal{O}\left(\frac{M}{R\sqrt{\log(R)}}\right)$, $0 < \delta < 1$ and,*

$$\mu = \min_{j \in \mathcal{S}} |\hat{f}_j|$$
$$\Delta = \frac{\max_{j \in \mathcal{S}} |\hat{f}_j|}{\min_{j \in \mathcal{S}} |\hat{f}_j|} = \frac{||\hat{f}||_\infty}{\mu}$$

Finally, let $F_{\mathcal{A}(K;M)}(\cdot)$ and $\Psi_\sigma(\cdot)$ be the operators found in Eq.(8). Then, there exists a constant $1 < C < \infty$ such that if,

$$K \geq C\, \frac{M\sqrt{\log\left(\frac{2\Delta}{\delta}\right)}}{\pi\, \sigma} \tag{16}$$

the inequality,

$$\left[F_{\mathcal{A}(K;M)}\left(\Psi_\sigma\left(f_{k;\,M}\right)\right)\right]_{n\frac{M}{K}} \geq \delta\, \mu$$

implies that

$$\inf_{j \in \mathcal{S}} \left| n\frac{M}{K} - j \right| \leq \sigma \sqrt{\log\left(\frac{2R\Delta}{\delta}\right)} \tag{17}$$

and,

$$\inf_{j \in \mathcal{S}} \left| n\frac{M}{K} - j \right| \leq \sigma \sqrt{\log\left(\frac{1}{\delta}\right)},$$

implies that

$$\left[F_{\mathcal{A}(K;M)}\left(\Psi_\sigma\left(f_{k;\,M}\right)\right)\right]_{n\frac{M}{K}} \geq \delta\, \mu$$

for all $n \in \{0, 1, ..., K-1\}$.

Proof. Consider the quantity

$$\frac{1}{M} \sum_{m \in \mathcal{A}(K;M)} e^{2\pi i \frac{n\frac{M}{K} m}{M}}\, \hat{g}_\sigma\left(\frac{m}{M}\right) f_{m;M} = \sum_{j \in \mathcal{S}} \left(\frac{1}{M} \sum_{m \in \mathcal{A}(K;M)} e^{2\pi i \frac{m}{K}(n-j)} \hat{g}_\sigma\left(\frac{m}{M}\right) \right) \hat{f}_j \tag{18}$$

and recall that,

$$\frac{1}{M}\hat{g}_\sigma\left(\frac{m}{M}\right) = \frac{\sqrt{\pi}\sigma}{M} \sum_{h \in \mathbb{Z}} e^{-\pi^2\sigma^2\left(\frac{m+hM}{M}\right)^2}$$

where $\mathcal{A}(K;M) := \left\{m \in \{0,1,...,M-1\} \,:\, m \leq \frac{K}{2} \text{ or } |m-M| < \frac{K}{2}\right\}$ (Definition 1). From this expression, it is apparent that there exists some constant $1 < C < \infty$ such that by choosing $K \geq C\,\frac{M\sqrt{\log\left(\frac{2\Delta}{\delta}\right)}}{\pi\,\sigma}$, one has,

$$\left|\sum_{j\in\mathcal{S}}\left(\frac{1}{M}\sum_{m\in\mathcal{A}^c(K;M)} e^{2\pi i\frac{m}{K}(n-j)}\hat{g}_\sigma\left(\frac{m}{M}\right)\right)\hat{f}_j\right| \leq \sum_{m\in\mathcal{A}^c(K;m)}\left(\frac{\sqrt{\pi}\sigma}{M}\sum_{h\in\mathbb{Z}} e^{-\pi^2\sigma^2\left(\frac{m+hM}{M}\right)^2}\right)||\hat{f}||_\infty \leq \frac{\delta\mu}{2}$$

Indeed, by the integral test,

$$\begin{aligned}
\sum_{m\in\mathcal{A}^c(K;m)}\left(\frac{\sqrt{\pi}\sigma}{M}\sum_{h\in\mathbb{Z}} e^{-\pi^2\sigma^2\left(\frac{m+hM}{M}\right)^2}\right) &\leq A\,\frac{\sqrt{\pi}\sigma}{M}\sum_{m\geq\frac{K}{2}} e^{-\pi^2\sigma^2\left(\frac{m}{M}\right)^2}\\
&\leq A\,\frac{\sqrt{\pi}\sigma}{M}\,e^{-\pi^2\sigma^2\left(\frac{K}{2M}\right)^2} + \frac{\sqrt{\pi}\sigma}{M}\int_{\frac{K}{2}}^{\infty} e^{-x^2\left(\frac{\pi\sigma}{M}\right)^2}\,\mathrm{d}x\\
&\leq A\,\frac{\sqrt{\pi}\sigma}{M}\,e^{-\pi^2\sigma^2\left(\frac{K}{2M}\right)^2} + \frac{1}{\sqrt{\pi}}\,\mathrm{erfc}\left(\frac{\pi K\sigma}{2M}\right)\\
&\leq B\,e^{-\pi^2\sigma^2\left(\frac{K}{2M}\right)^2}
\end{aligned}$$

for some positive constants A, B, and where the last inequality follows from estimates on the complementary error function [6] and the fact that $\frac{\sqrt{\pi}\sigma}{M} = \mathcal{O}\left(\frac{1}{R\sqrt{\log(R)}}\right)$ by assumption. Therefore,

$$\begin{aligned}
\left[F_{\mathcal{A}(K;M)}\left(\Psi_\sigma\left(f_{m;\,M}\right)\right)\right]_{n\frac{M}{K}} &= \frac{1}{M}\sum_{m\in\mathcal{A}(K;M)} e^{2\pi i\frac{n\,m}{K}}\,\hat{g}_\sigma\left(\frac{m}{M}\right)\,f_{m;M}\\
&= \frac{1}{M}\sum_{m=0}^{M-1} e^{2\pi i\frac{n\frac{M}{K}m}{M}}\,\hat{g}_\sigma\left(\frac{m}{M}\right)\,f_{m;M} + \epsilon_n\\
&= \sum_{j\in\mathcal{S}} e^{-\frac{\left(n\frac{M}{K}-j\right)}{\sigma^2}}\,\hat{f}_j + \epsilon_n
\end{aligned}$$

where $\max_n|\epsilon_n| \leq \frac{\delta\,\mu}{2}$. Now assume: $|\left[F_{\mathcal{A}(K;M)}\left(\Psi_\sigma\left(f_{k;\,M}\right)\right)\right]_{n\frac{M}{K}}| \geq \delta\,\mu$. Then, the triangle inequality and the previous equation imply that,

$$\sum_{j\in\mathcal{S}} e^{-\frac{\left(n\frac{M}{K}-j\right)}{\sigma^2}}\,\hat{f}_j \geq \left|\left[F_{\mathcal{A}(K;M)}\left(\Psi_\sigma\left(f_{m;\,M}\right)\right)\right]_{n\frac{M}{K}}\right| - \frac{\delta\,\mu}{2} \geq \left(\delta - \frac{\delta}{2}\right)\mu = \frac{\delta}{2}\,\mu. \tag{19}$$

We claim that this cannot occur unless,

$$\inf_{j\in\mathcal{S}}\left|n\frac{M}{K} - j\right| \leq \sigma\,\sqrt{\log\left(\frac{2R\Delta}{\delta}\right)}. \tag{20}$$

We proceed by contradiction. Assume the opposite holds. Then,

$$\sum_{j\in\mathcal{S}} e^{-\frac{\left(n\frac{M}{K}-j\right)}{\sigma^2}}\,\hat{f}_j \leq ||\hat{f}||_\infty\sum_{j\in\mathcal{S}} e^{-\frac{\left(n\frac{M}{K}-j\right)^2}{\sigma^2}} < ||\hat{f}||_\infty\,\frac{\delta}{2\Delta} = \frac{\delta\,\mu}{2}$$

by assumption. This is a contradiction. Thus, Eq.(20) must indeed hold. This proves the first part of the proposition. For the second part, assume

$$\inf_{j\in\mathcal{S}}\left|n\frac{M}{K} - j\right| \leq \sigma\,\sqrt{\log\left(\frac{1}{\delta}\right)} \tag{21}$$

holds. Letting j^* be such that $\left|n\frac{M}{K}-j^*\right| = \inf_{j\in\mathcal{S}}\left|n\frac{M}{K}-j\right|$, we note that,

$$\sum_{j\in\mathcal{S}} e^{-\frac{\left(n\frac{M}{K}-j\right)^2}{\sigma^2}}\,\hat{f}_j \geq e^{-\frac{\left(n\frac{M}{K}-j^*\right)^2}{\sigma^2}}\,\hat{f}_{j^*} \geq \delta\,\mu$$

since $\hat{f}$ and the Gaussian are all positive by assumption. This shows the second part. □

We are now ready to prove the validity of the Algorithm 1.

Proposition 5. *(Correctness of Algorithm 3) Consider a function $f(x)$ of the form of Eq.(1) and satisfying the constraint Eq.(2), and let $\Pi_Q(\cdot)$, $\Psi_\sigma(\cdot)$ and $F_K(\cdot)$ be the operators found in Eq.(8) where δ, μ, Δ and K are as in Lemma 5 and K satisfies the additional constraint $K > \frac{R}{\alpha}\sqrt{\frac{\log\left(\frac{2R\Delta}{\delta}\right)}{\log\left(\frac{1}{\delta}\right)}}$, and*

$$\sigma = \frac{\alpha\,\frac{M}{2R}}{\sqrt{\log\left(\frac{2R\Delta}{\delta}\right)}} \tag{22}$$

for some $0<\alpha<1$. Assume further that the integers $\{Q^{(l)}\}_{l=1}^L$ are chosen independently and uniformly at random within $\mathcal{Q}(M)$, for some $1 \leq L \in \mathbb{N}$. Consider

$$\phi_{\left[i\frac{K}{M}\right]\frac{M}{K}}(Q^{(l)}) := \left[F_{\mathcal{A}(K;M)}\left(\Psi_\sigma\left(\Pi_{Q^{(l)}}\left(f_{k;M}\right)\right)\right)\right]_{\left[i\frac{K}{M}\right]\frac{M}{K}} \tag{23}$$

Then,

$$\mathbb{P}\left(\cap_{l=1}^L\left\{|\phi_{\left[(i[Q]_M^{-1} \bmod M)\,\frac{K}{M}\right]\frac{M}{K}}(Q^{(l)})| \geq \delta\,\mu\right\}\right) \leq \alpha^L \tag{24}$$

for every $i \in \mathcal{S}^c$, and

$$|\phi_{\left[(i[Q]_M^{-1} \bmod M)\,\frac{K}{M}\right]\frac{M}{K}}(Q^{(l)})| \geq \delta\,\mu$$

almost surely for all $Q^{(l)}$ and every $i\in\mathcal{S}$.

Proof. From independence, the probability in Eq. (24) is equal to,

$$\prod_{l=1}^L \mathbb{P}\left(\left|\phi_{\left[(i[Q]_M^{-1} \bmod M)\,\frac{K}{M}\right]\frac{M}{K}}(Q^{(l)})\right| \geq \delta\,\mu\right).$$

So it is sufficient to consider a fixed l. As a consequence of Lemma 5 and Lemma 3 we have the inclusion,

$$\begin{aligned}\left\{|\phi_{\left[(i[Q]_M^{-1} \bmod M)\,\frac{K}{M}\right]\frac{M}{K}}(Q^{(l)})| \geq \delta\,\mu\right\} &\subset \left\{\inf_{j\in\mathcal{S}}\left|\left[(i[Q^{(l)}]_M^{-1} \bmod M)\,\frac{K}{M}\right]\frac{M}{K} - (j[Q^{(l)}]_M^{-1}) \bmod M\right| \leq \sigma\sqrt{\log\left(\frac{2R\Delta}{\delta}\right)}\right\}\\ &\subset \cup_{j\in\mathcal{S}}\left\{\left|((i-j)[Q^{(l)}]_M^{-1}) \bmod M\right| \leq \sigma\sqrt{\log\left(\frac{2R\Delta}{\delta}\right)} + \frac{M}{2K}\right\},\end{aligned}$$

which implies that the probability for each fixed l is bounded by,

$$\begin{aligned}\mathbb{P}\left(|\phi_{\left[(i[Q]_M^{-1} \bmod M)\frac{K}{M}\right]\frac{M}{K}}(Q^{(l)})| \geq \delta\,\mu\right) &\leq \sum_{j\in\mathcal{S}}\mathbb{P}\left(\left|((i-j)[Q^{(l)}]_M^{-1}) \bmod M\right| \leq \sigma\sqrt{\log\left(\frac{2R\Delta}{\delta}\right)} + \frac{M}{2K}\right)\\ &\leq \mathcal{O}\left(R\,\left(\frac{\sigma\sqrt{\log\left(\frac{2R\Delta}{\delta}\right)} + \frac{M}{2K}}{M}\right)\right)\\ &= \mathcal{O}(\alpha),\end{aligned}$$

by the union bound, by Lemma 4 (since $i \neq j$) and by assumption. Therefore,

$$\mathbb{P}\left(\cap_{l=1}^{L}\left\{|\phi_{\left[i\frac{K}{M}\right]\frac{M}{K}}(Q^{(l)})| \geq \delta\,\mu\right\}\right) \leq \mathcal{O}\left(\alpha^{L}\right)$$

as claimed. As for the second part of the proposition, note that if $i \in \mathcal{S}$ then

$$\begin{aligned}\inf_{j\in\mathcal{S}}\left|\left[\frac{(i[Q^{(l)}]_M^{-1}) \bmod M}{\frac{M}{K}}\right]\frac{M}{K} - (j[Q^{(l)}]_M^{-1}) \bmod M\right| &\leq \inf_{j\in\mathcal{S}}\left|(i[Q^{(l)}]_M^{-1}) \bmod M - (j[Q^{(l)}]_M^{-1}) \bmod M\right| + \frac{M}{2K}\\ &= \frac{M}{2K}\\ &\leq \sigma\sqrt{\log\left(\frac{1}{\delta}\right)}\end{aligned}$$

by assumption. By Lemma 5, this implies that

$$|\phi_{\left[(i[Q]_M^{-1} \bmod M)\,\frac{K}{M}\right]\frac{M}{K}}(Q^{(l)})| \geq \delta\,\mu$$

and since this is true regardless of the value of the random variable $Q^{(l)}$, we conclude that it holds almost surely. ☐

Remark 1. A careful study of the proof of Lemma 5 and Proposition 5 shows that the order $\mathcal{O}\left(R\sqrt{\log(R)}\right)$ size of K arises from the need to bound quantities of the form $\sum_{j\in\mathcal{S}} e^{-\frac{\left(n\frac{M}{K}-j\right)^2}{\sigma^2}}$. In the worst-case scenario (the case treated by Lemma 5), this requires estimates of the form of Eq.(16), Eq.(17) and Eq.(22) which introduce an extra $\sqrt{\log(R)}$ factor in the computational cost (Section 3) relative to the (conjectured) optimal scaling. However, throughout the algorithm the elements of any aliased support $\mathcal{S}_k$ appearing in the sum are always subject to random shuffling first. Lemma 4 states that the shuffling tends to be more of less uniform. Now, were the elements *i.i.d. uniformly distributed*, it would be easy to show that these quantities are of order $\mathcal{O}(1)$ with high probability, removing the need for the extraneous factor. Unfortunately, our current theoretical apparatus does not allow us to prove the latter. However, following this argument and numerical experiments, we strongly believe that it is possible. In this sense, we believe that through a slight modification of the choice of parameters, our algorithm exhibits an (optimal) $\mathcal{O}\left(R\log(R)\log(N)\right)$ computational complexity with the same guarantees of correctness as the current scheme.

Lemma 6. *Let $\{P^{(t)}\}_{t=1}^{T}$ be prime numbers greater than or equal to $R \in \mathbb{N}$, and let $i, j \in \{0, 1, ..., N-1\}$ such that,*

$$i \bmod P^{(t)} = j \bmod P^{(t)}, \ \ t = 1, 2, ..., T.$$

If $T > \log_R(N)$, then $i = j$.

Proof. Consider $\{P^{(t)}\}_{t=1}^{T}$ as described above and $T > \log_R(N)$, and assume that

$$iQ^{(t)} \bmod P^{(t)} = jQ^{(t)} \bmod P^{(t)}$$

for $t = 0, 1, ..., T$. Since $Q^{(t)} \perp P^{(t)}$ this is an isomorphism (Lemma 2) and therefore the above statement is equivalent to

$$i \bmod P^{(t)} = j \bmod P^{(t)}$$

for $t = 0, 1, ..., T$. This implies in particular that

$$P^{(t)} \,|\, (j - i)$$

for $t = 0, 1, ..., T$, and that

$$\text{lcm}(\{P^{(t)}\}_{t=1}^{T}) \,|\, (j - i).$$

However, since the integers $\{P^{(t)}\}_{t=1}^T$ are prime (and therefore coprime),

$$\mathrm{lcm}(P^{(t)}) = \prod_{t=1}^{T} P^{(t)} \geq \left(\min_t P^{(t)}\right)^T \geq R^{\log_R(N)} = N.$$

This implies that,

$$|j - i| \geq N,$$

since $i \neq j$, and this is a contradiction since both belong to $\{0, 1, ..., N-1\}$. ☐

Corollary 2. *Let $\{P^{(t)}\}_{t=1}^T$ are as in Lemma 6 and that $i \neq j$, $k \neq l$, $i, j, k, l \in \{0, 1, ..., N-1\}$ are such that,*

$$i \bmod P^{(t)} = j \bmod P^{(t)}$$
$$k \bmod P^{(t)} = l \bmod P^{(t)}$$

for $t = 1, 2, ..., T$. Then,

$$(i - j) = (k - l)$$

Proof. The statement is equivalent to,

$$(i - j) \bmod P^{(t)} = 0 = (k - l) \bmod P^{(t)}$$

for $t = 1, 2, ...T$. By Lemma 6, this implies that $(j - i) = (k - l)$. ☐

Proposition 6. *Let $0 < R < N \in \mathbb{N}$. Further let $\{P^{(t)}\}$ be random integers uniformly distributed within the set $\mathcal{P}$ containing the $4R\log_R(N)$ smallest prime numbers strictly larger than R, and let F and B be defined as in Eq.(10) with these parameters. If $T \geq 4$, then,*

$$\mathbb{P}\left(\left|\left|(I - \frac{1}{T}(FB)^*(FB))x\right|\right|_2 > \frac{1}{2}\right) \leq \frac{1}{2}$$

Proof. First, note that,

$$(FB)^*(FB) = B^*F^*FB = B^*B$$

since F is a block-diagonal Fourier matrix, and $I - \frac{1}{T}B^*B$ has entries

$$\left[I - \frac{1}{T}B^{(t)^*}B^{(t)}\right]_{ij} = \delta_{i,j} - \frac{1}{T}\sum_s B_{si}^{(t)} B_{sj}^{(t)} = \begin{cases} \frac{1}{T} & \text{if } i[Q^{(t)}]_{P^{(t)}}^{-1} \bmod P^{(t)} = j[Q^{(t)}]_{P^{(t)}}^{-1} \bmod P^{(t)} \\ 0 & \text{o.w.} \end{cases} \quad (25)$$

Therefore, for any vector x such that $||x||_2 = 1$,

$$\begin{aligned}\mathbb{P}\left(\left|\left|(I - \frac{1}{T}(FB)^*(FB))x\right|\right|_2 > \frac{1}{2}\right) &\leq 4\,\mathbb{E}\left[\left(\sum_{i\neq j}\sum_t \bar{x}_i\,[B^{(t)^*}B^{(t)}]_{ij}\,x_j\right)^2\right] \\ &= 4\sum_{i\neq j}\sum_{k\neq l} \bar{x}_i\,x_j\,x_k\,\bar{x}_l \sum_{s,t} \mathbb{E}\left[[B^{(t)^*}B^{(t)}]_{ij}\,[B^{(s)^*}B^{(s)}]_{kl}\right]\end{aligned}$$

by Chebyshev inequality. Furthermore, thanks to Eq.(25) and independence, the expectation can be written as,

$$\mathbb{E}\left[[B^{(t)^*}B^{(t)}]_{ij}\,[B^{(s)^*}B^{(s)}]_{kl}\right] = \begin{cases} \mathbb{P}\left(\left\{(i-j) \bmod P^{(t)} = 0\right\}\right)\mathbb{P}\left(\left\{(k-l) \bmod P^{(t)} = 0\right\}\right) & \text{if } s \neq t \\ \mathbb{P}\left(\left\{(i-j) \bmod P^{(t)} = 0\right\} \cap \left\{(k-l) \bmod P^{(t)} = 0\right\}\right) & \text{if } s = t \end{cases} \quad (26)$$

Now, let $\tau(i, j)$ be defined as

$$\tau(i, j) := \left\{P^{(t)} \in \mathcal{P} : i \bmod P^{(t)} = j \bmod P^{(t)}\right\}.$$

The case $s \neq t$ is treated as follows,

$$\begin{aligned}
&\mathbb{P}\left(\left\{(i-j) \bmod P^{(t)} = 0\right\}\right) \mathbb{P}\left(\left\{(k-l) \bmod P^{(s)} = 0\right\}\right) \\
&= \left(\sum_{p_1 \in \tau(i,j)} \mathbb{P}\left(\left\{(i-j) \bmod P^{(t)} = 0\right\} \Big| P^{(t)} = p_1\right) \mathbb{P}\left(P^{(t)} = p_1\right)\right) \cdot \\
&\left(\sum_{p_2 \in \tau(k,l)} \mathbb{P}\left(\left\{(k-l) \bmod P^{(s)} = 0\right\} \Big| P^{(s)} = p_2\right) \mathbb{P}\left(P^{(s)} = p_2\right)\right) \\
&\leq \left(\frac{\#\tau(i,j)}{4R\log_R(N)} \frac{\#\tau(k,l)}{4R\log_R(N)}\right) \\
&\leq \frac{1}{16R^2}
\end{aligned}$$

since $P^{(t)}$ is uniformly distributed within a set of cardinality $4R\log_R(N)$, and because,

$$\sum_{p_1 \in \tau(i,j)} \mathbb{P}\left(\left\{(i-j) \bmod P^{(t)} = 0\right\} \Big| P^{(s)} = p_1\right) = \#\tau(i,j) = \log_R(N)$$

by Lemma 6. This leaves us the case $s = t$. To this purpose, we further split this case into two subcases: that when $i - j = k - l$ and that when $i - j \neq k - l$. When $i - j = k - l$ we obtain,

$$\begin{aligned}
&\sum_{s,t=1}^{T} \mathbb{P}\left(\{s = t\} \cap \{i - j = k - l\} \cap \left\{(i-j) \bmod P^{(t)} = 0\right\} \cap \left\{(k-l) \bmod P^{(s)} = 0\right\}\right) \\
&= \sum_{p \in \tau(k,l)} \mathbb{P}\left(\left\{(k-l) \bmod P^{(t)} = 0\right\} \Big| P^{(t)} = p\right) \mathbb{P}\left(P^{(t)} = p\right) \\
&\leq \frac{1}{4R}
\end{aligned}$$

since $k \neq l$, following an argument similar to the previous one. This leaves the case $s = t$, $i - j \neq k - l$. However, thanks to Corollary 2 it follows that the set,

$$\{s = t\} \cap \{i \neq j\} \cap \{k \neq l\} \cap \{i - j \neq k - l\} \cap \left\{(i-j) \bmod P^{(t)} = 0\right\} \cap \left\{(k-l) \bmod P^{(s)} = 0\right\}$$

must be empty. Putting everything together we find that,

$$\begin{aligned}
\mathbb{P}\left(\left\|(I - \frac{1}{T}(FB)^*(FB))x\right\|_2 > \frac{1}{2}\right) &\leq 4 \sum_{i \neq j} \sum_{k \neq l} \bar{x}_i \, x_j \, x_k \, \bar{x}_l \frac{1}{T^2} \left[\sum_{s,t=1}^{T} \left(\mathbb{E}\left[\mathbb{I}_{s \neq t}(s,t) \, [B^{(t)^*} B^{(t)}]_{ij} \, [B^{(s)^*} B^{(s)}]_{kl}\right] + \right.\right. \\
&\left.\left. \mathbb{E}\left[\mathbb{I}_{s=t}(s,t) \, \mathbb{I}_{i-j=k-l}(i,j,k,l) \, [B^{(t)^*} B^{(t)}]_{ij} \, [B^{(s)^*} B^{(s)}]_{kl}\right]\right)\right] \\
&\leq 4\left(\frac{1}{16R^2}\right)\left(\sum_{k \neq l} \bar{x}_l \, x_k\right)^2 + \frac{4}{T}\left(\frac{1}{4R}\right)\left(\sum_{k \neq l} \bar{x}_l \, x_k\right)\left(\sum_j \bar{x}_{j+k-l} \, x_j\right)
\end{aligned}$$

We further note that $\sum_{i \neq j} \bar{x}_l \, x_k$ is a bilinear form bounded by the norm of an $R \times R$ matrix with all entries equal to 1 except the diagonal which is all zeros. It is easy to work out this norm which is equal to $R - 1$ so that,

$$\frac{1}{R} \sum_{k \neq l} \bar{x}_l \, x_k < 1$$

Finally, by Cauchy-Schwartz inequality,

$$\left|\sum_j \bar{x}_{j+k-l}\, x_j\right| \leq \sqrt{\sum_j |x_{j+k-l}|^2}\,\sqrt{\sum_j |x_j|^2} = ||x||_2^2 = 1.$$

Thus,

$$\mathbb{P}\left(\left|\left|(I - \frac{1}{T}(FB)^*(FB))x\right|\right|_2 > \frac{1}{2}\right) < \frac{1}{4} + \frac{1}{T} \leq \frac{1}{2}$$

as claimed. □

Corollary 3. *Under the hypotheses of Proposition 6, the solution to the linear system*

$$FB\,\hat{f} = f_0$$

takes the form,

$$\hat{f} = \sum_{n=0}^{\infty} \left[I - \frac{1}{T}B^*B\right]^n \left(\frac{1}{\sqrt{T}}(FB)^* f_0\right)$$

with probability at least $\frac{1}{2}$.

Proof. By Proposition 6, $||I - \frac{1}{T}(FB)^*(FB)||_2 < \frac{1}{2}$ with probability at least $\frac{1}{2}$. When this is the case we write,

$$FB\,\hat{f} = f_0 \Leftrightarrow \frac{1}{T}B^*B\,\hat{f} = \frac{1}{\sqrt{T}}B^*F^*f_0 \Leftrightarrow \left[I - \left(I - \frac{1}{T}B^*B\right)\right]\,\hat{f} = \frac{1}{\sqrt{T}}(FM)^*b = \hat{f}_0$$

In this case, it is easy to verify that the Neumann series,

$$\hat{f} = \sum_{n=0}^{\infty} \left[I - \frac{1}{T}B^*B\right]^n \left(\frac{1}{\sqrt{T}}(FB)^* b\right)$$

satisfies this last equation, and that the sum converges exponentially fast. □

Proposition 3. *(Correctness of Algorithm 4) Assume the support $\mathcal{S}$ of $\hat{f}$ is known. Then Algorithm 4 outputs an approximation to the nonzero elements of $\hat{f}$ with error bounded by η in the ℓ^2-norm, with probability greater than or equal to $1-p$.*

Proof. By Proposition 6, $\frac{1}{T}(FB)(FB)^* = I - \mathcal{P}$ where $\left|\left|I - \frac{1}{T}(FB)(FB)^*\right|\right|_2 = ||\mathcal{P}||_2 < \frac{1}{2}$ with probability larger than $\frac{1}{2}$. Thus, if we consider $\mathcal{O}(\log_{\frac{1}{2}}(p))$ independent realizations of FB, the probability that at least one of them is such is greater than or equal to $(1-p)$. When this occur, Corollary 3 states that the solution is given by the Neumann series. Furthermore,

$$\begin{aligned}
\left|\left|\hat{f} - \sum_{n=0}^{\lceil \log_{\frac{1}{2}}(\eta)\rceil} \mathcal{P}^n\, f^\dagger\right|\right|_2 &= \left|\left|\sum_{\lceil \log_{\frac{1}{2}}(\eta)\rceil}^{\infty} \mathcal{P}^n\, f^\dagger\right|\right|_2 \\
&\leq \sum_{\lceil \log_{\frac{1}{2}}(\eta)\rceil}^{\infty} ||\mathcal{P}||_2^n\, ||f^\dagger||_2 \\
&\leq \mathcal{O}(\eta)
\end{aligned}$$

by the geometric series and the bound $||\mathcal{P}||_2 \leq \frac{1}{2}$.

□

A.2 Proofs of Section 3.3

Lemma 1. *Assuming the noise term $\hat{\nu}$ is such that $||\hat{\nu}||_2 < \frac{\eta}{\sqrt{N}}$, the error term of the computed value in Eq.(11) is uniformly bounded by*

$$\left|\left|\psi_n^{(k)}(Q)\right|\right|_\infty = \left|\left|\frac{1}{M_k} \sum_{m\in\mathcal{A}(K,M_k)} e^{2\pi i \frac{nm}{K}}\, g_\sigma\left(\frac{m}{M_k}\right)\, \nu_{(mQ) \bmod M_k;\, M_k}\right|\right|_\infty < \mathcal{O}(\eta).$$

Proof. First, not that since $\Pi_Q(\cdot)$ is an isomorphic permutation operator (for all $Q \in \mathcal{Q}(M_k)$) one has

$$||\Pi_Q||_\infty = 1.$$

Similarly, since the filtering operator $\Psi_\sigma(\cdot)$ is diagonal with nonzero entries $\hat{g}_\sigma(n)$, then

$$||\Psi_\sigma||_\infty = \sup_{m\in\{0,1,\dots,M_k-1\}} \left|\hat{g}_\sigma\left(\frac{m}{M_k}\right)\right| \le \sqrt{\pi}\,\sigma = \frac{\sqrt{\pi}\frac{\alpha M_k}{2R}}{\sqrt{\log\left(\frac{2R\Delta}{\delta}\right)}}.$$

Finally, we get from the triangle inequality that,

$$\begin{aligned}\le \frac{\#\mathcal{A}(K;M_k)}{M_k}\, ||\Psi_\sigma||_\infty\, ||\Pi_Q||_\infty\, ||\nu||_\infty &\le \frac{\sqrt{\pi}\frac{\alpha K}{R}}{\sqrt{\log\left(\frac{2R\Delta}{\delta}\right)}}\, ||\nu||_\infty \\ &\le \frac{\sqrt{\pi}\frac{\alpha K}{R}}{\sqrt{\log\left(\frac{2R\Delta}{\delta}\right)}}\, ||\hat{\nu}||_1\end{aligned}$$

by the Hausdorff-Young inequality [4]. Finally, we note that: $||\hat{\nu}||_1 \le \sqrt{N}\, ||\hat{\nu}||_2 < \eta$ by assumption, and recall that $K = \mathcal{O}(R\sqrt{\log(R)})$. This leads to the desired result.

□

A.3 Proof of Section 4

Proposition 4. *(Rank-1 d-dimensional DFT) Assume the function $f : [0,1)^d \to \mathbb{C}$ has form (12). Then,*

$$\int_{[0,1]^d} e^{-2\pi i\, j\cdot x}\, f(x)\,\mathrm{d}x = \frac{1}{N}\sum_{n=0}^{N-1} e^{-2\pi i\, j\cdot x_n}\, f(x_n) \tag{13}$$

for all $j \in [0,M)^d \cap \mathbb{Z}^d$, where $x_n = \frac{ng \bmod N}{N}$, $g = (1, M, M^2, ..., M^{d-1})$ and $N = M^d$.

Proof. First, note that

$$\int_{[0,1]^d} e^{-2\pi i\, j\cdot x}\, f(x)\,\mathrm{d}x = \hat{f}_j.$$

Then, substitute the samples in the quadrature to obtain

$$\begin{aligned}\frac{1}{N}\sum_{n=0}^{N-1} e^{-2\pi i\, j\cdot x_n}\, f(x_n) &= \frac{1}{N}\sum_{n=0}^{N-1} e^{-2\pi i\, j\cdot \frac{ng \bmod N}{N}} \left(\sum_{k\in[0,M)^d\cap\mathbb{Z}^d} \hat{f}_k\, e^{2\pi i\, k\cdot \frac{ng \bmod N}{N}}\right) \\ &= \sum_{k\in[0,M)^d\cap\mathbb{Z}^d} \hat{f}_k \left(\frac{1}{N}\sum_{n=0}^{N-1} e^{-2\pi i\, \frac{n((k-j)\cdot g)}{N}}\right)\end{aligned}$$

since $e^{2\pi i\,(k-j)\cdot \frac{ng \bmod N}{N}} = e^{2\pi i\,(k-j)\cdot \frac{ng}{N}}$. Note however that

$$\frac{1}{N}\sum_{n=0}^{N-1} e^{-2\pi i\, \frac{n((k-j)\cdot g)}{N}} = D_N\left((k-j)\cdot g\right),$$

which is the Dirichlet kernel and is equal to 0 unless $(k-j)\cdot g = 0 \bmod N$, in which case it is equal to 1. Thus,

$$\frac{1}{N}\sum_{n=0}^{N-1} f(x_n) = \hat{f}_j + \sum_{\substack{k\in[0,M)^d\cap\mathbb{Z}^d \\ (k-j)\cdot g \bmod N\equiv 0 \\ (k-j)\cdot g\neq 0}} \hat{f}_k.$$

Thus, in order to show that the quadrature is exact, it suffices to show that the remaining sum on the right-hand side of the previous equation is trivial. To see this, note that $(k-j)\in[-M,M)^d\cap\mathbb{Z}^d$ and consider

$$|(k-j)\cdot g| = \left|(k_1-j_1)+(k_2-j_2)M+...+(k_d-j_d)M^{d-1}\right| \leq M\sum_{l=0}^{d-1} M^l = M\,\frac{1-M^d}{1-M} < M^d = N,$$

where the inequality is strict for any finite $M\in\mathbb{N}$ strictly larger than 1. This implies that there cannot be any $(k-j)$ other than 0 in the domain of interest such that $(k-j)\cdot g \bmod N \equiv 0$. The sum is therefore empty and the result follows. □

B Generalization to general complex sparse vectors

This appendix provides a terse description of the additional steps necessary to transform the sMFFT for *real positive vectors* into a reliable algorithm for *general complex vectors.* To achieve this task, two major hurdles, both associated with the support-recovery portion of the scheme, must be overcome; the first one is associated with the initial aliasing of the signal described in Section 3. As shown Eq.(4), at each step aliasing implies Fourier coefficients of the form,

$$\hat{f}_l^{(k)} = \sum_{j:j \bmod M_k = l} \hat{f}_j,\ l=0,1,...,M_k.$$

When the original nonzero coefficients are all strictly positive, this expression is positive *if and only if* the lattice $l+iM_k,\ i=0,1,...\frac{N}{M_k}-1$ contains one of the original nonzero coefficients. When the nonzero coefficients are complex however, this is no longer true. The second potential issue pertains to the resulting filtering step found in Algorithm 3. As described by Eq.(7), the result takes the form,

$$\left[\Psi_\sigma\left(\Pi_Q\left(f_{n;\,M_k}\right)\right)\right](\xi) = \mathcal{F}^*\left[\sum_{j\in\mathcal{S}_k} \hat{f}_j^{(k)}\, e^{-\frac{\left|x-\left(j[Q]^{-1}_{M_k} \bmod M_k\right)\right|^2}{\sigma^2}}\right](\xi).$$

which corresponds to the Fourier transform of the aliased signal convoluted with a Gaussian. Once again, the crucial statistical test used in Algorithm 3 relies on this quantity being positive if and only if a point lies in the vicinity of an element of the (shuffled and aliased) support $\mathcal{S}_k$. Such statement does not hold true if we allow the coefficients to be general complex numbers (as some elements might *cancels out*).

The conclusion of these observations is that as a consequence of the lack of positivity, it is possible that elements belonging to $\mathcal{M}_k\cap\mathcal{S}_k$ might be wrongfully eliminated in Algorithm 3, i.e., the *false negative identification rate is nontrivial.* To alleviate these issues, we propose a slight modification to the scheme; we allow for the possibility of the output of Algorithm 3 be missing elements of $\mathcal{S}_k$ by launching multiple independent runs of the FIND_SUPPORT(·) routine in Algorithm 1, and taking the *union* of the outputs. In this sense, although it is possible to miss an element with a single run, we expect that the probability of a miss over multiple independent run is very small. In addition, this modification *does not have any effect on the fundamental computational complexity*; indeed, close examination shows that these additional steps *only increase the algorithmic constant by some small quantity independent of N and/or R.*

So far, this modification remains a heuristic (with some preliminary/unpublished theoretical backing). Note however that we have implemented it and can attest to excellent numerical results in line with our expectation based on the previous discussion, and very similar to those obtained in the real-positive case.

References

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

[2] Adi Akavia. Deterministic sparse Fourier approximation via approximating arithmetic progressions. *Information Theory, IEEE Transactions on*, 60(3):1733–1741, 2014.

[3] Adi Akavia, Shafi Goldwasser, and Shmuel Safra. Proving hard-core predicates using list decoding. In *FOCS*, volume 44, pages 146–159, 2003.

[4] William Beckner. Inequalities in Fourier analysis. *Annals of Mathematics*, pages 159–182, 1975.

[5] Petros Boufounos, Volkan Cevher, Anna C Gilbert, Yi Li, and Martin J Strauss. What's the frequency, Kenneth?: Sublinear Fourier sampling off the grid. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 61–72. Springer, 2012.

[6] Seok-Ho Chang, Pamela C Cosman, and Laurence B Milstein. Chernoff-type bounds for the gaussian error function. *IEEE Transactions on Communications*, 59(11):2939–2944, 2011.

[7] Eleanor Chu and Alan George. *Inside the FFT black box: serial and parallel Fast Fourier Transform algorithms*. CRC Press, 1999.

[8] James W. Cooley and John W. Tukey. An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation*, 19(90):297–301, 1965.

[9] Tom Darden, Darrin York, and Lee Pedersen. Particle mesh ewald: An $nlog(n)$ method for ewald sums in large systems. *The Journal of chemical physics*, 98(12):10089–10092, 1993.

[10] Matteo Frigo and Steven G Johnson. FFTW: An adaptive software architecture for the FFT. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 3, pages 1381–1384. IEEE, 1998.

[11] Badih Ghazi, Haitham Hassanieh, Piotr Indyk, Dina Katabi, Erik Price, and Lixin Shi. Sample-optimal average-case sparse Fourier Transform in two dimensions. In *Communication, Control, and Computing (Allerton), 2013 51st Annual Allerton Conference on*, pages 1258–1265. IEEE, 2013.

[12] Anna C. Gilbert, Sudipto Guha, Piotr Indyk, S Muthukrishnan, and Martin Strauss. Near-optimal sparse Fourier representations via sampling. In *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*, pages 152–161. ACM, 2002.

[13] Anna C Gilbert, S Muthukrishnan, and Martin Strauss. Improved time bounds for near-optimal sparse Fourier representations. In *Optics & Photonics 2005*, pages 59141A–59141A. International Society for Optics and Photonics, 2005.

[14] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 25–32. ACM, 1989.

[15] Haitham Hassanieh, Piotr Indyk, Dina Katabi, and Eric Price. Nearly optimal sparse Fourier Transform. In *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing*, pages 563–578. ACM, 2012.

[16] Haitham Hassanieh, Piotr Indyk, Dina Katabi, and Eric Price. Simple and practical algorithm for sparse Fourier Transform. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1183–1194. SIAM, 2012.

[17] Piotr Indyk and Michael Kapralov. Sample-optimal Fourier sampling in any constant dimension. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 514–523. IEEE, 2014.

[18] Piotr Indyk and Michael Kapralov. Sparse Fourier transform in any constant dimension with nearly-optimal sample complexity in sublinear time. 2014.

[19] Piotr Indyk, Michael Kapralov, and Eric Price. (nearly) sample-optimal sparse Fourier Transform. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 480–499. SIAM, 2014.

[20] Mark A Iwen. Combinatorial sublinear-time Fourier algorithms. *Foundations of Computational Mathematics*, 10(3):303–338, 2010.

[21] Mark A Iwen. Improved approximation guarantees for sublinear-time Fourier algorithms. *Applied And Computational Harmonic Analysis*, 34(1):57–82, 2013.

[22] Michael Kapralov. Sparse fourier transform in any constant dimension with nearly-optimal sample complexity in sublinear time. *arXiv preprint arXiv:1604.00845*, 2016.

[23] Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the Fourier spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, 1993.

[24] David Lawlor, Yang Wang, and Andrew Christlieb. Adaptive sub-linear time Fourier algorithms. *Advances in Adaptive Data Analysis*, 5(01):1350003, 2013.

[25] Pierre-David Letourneau, Harper Langston, and Richard Lethin. A sparse multi-dimensional fast Fourier transform with stability to noise in the context of image processing and change detection. In *Proceedings of the 2016 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE, to appear.

[26] Juan M Lopez-Sanchez and Joaquim Fortuny-Guasch. 3-D radar imaging using range migration techniques. *Antennas and Propagation, IEEE Transactions on*, 48(5):728–737, 2000.

[27] Stéphane Mallat. *A wavelet tour of signal processing*. Academic press, 1999.

[28] Yishay Mansour. Randomized interpolation and approximation of sparse polynomials. *SIAM Journal on Computing*, 24(2):357–368, 1995.

[29] Louis Joel Mordell. *Diophantine equations*. Elsevier, 1969.

[30] Jacques Morgenstern. Note on a lower bound on the linear complexity of the Fast Fourier Transform. *Journal of the ACM (JACM)*, 20(2):305–306, 1973.

[31] Frank Ong, Sameer Pawar, and Kannan Ramchandran. Fast and efficient sparse 2d discrete fourier transform using sparse-graph codes. *arXiv preprint arXiv:1509.05849*, 2015.

[32] Sanjay Pawar and Kannan Ramchandran. Computing a k-sparse n-length discrete Fourier Transform using at most 4k samples and o (k log k) complexity. In *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on*, pages 464–468. IEEE, 2013.

[33] Gerlind Plonka and Katrin Wannenwetsch. Deterministic sparse FFT algorithms. *PAMM*, 15(1):667–668, 2015.

[34] Gerlind Plonka and Katrin Wannenwetsch. A sparse fast Fourier algorithm for real nonnegative vectors. *arXiv preprint arXiv:1602.05444*, 2016.

[35] Paulo Ribenboim. *The book of prime number records*. Springer Science & Business Media, 2012.

[36] Charles Van Loan. *Computational frameworks for the Fast Fourier Transform*, volume 10. Siam, 1992.