Sampling Theory, Signal Processing, and Data Analysis (2022) 20:1
https://doi.org/10.1007/s43670-021-00018-y

ORIGINAL ARTICLE

Sparse Fourier transforms on rank-1 lattices for the rapid and low-memory approximation of functions of many variables

Craig Gross[1] · Mark Iwen[2] · Lutz Kämmerer[3] · Toni Volkmer[3]

Received: 7 February 2021 / Accepted: 16 November 2021

Abstract

This paper considers fast and provably accurate algorithms for approximating smooth functions on the d-dimensional torus, $f : \mathbb{T}^d \to \mathbb{C}$, that are sparse (or compressible) in the multidimensional Fourier basis. In particular, suppose that the Fourier series coefficients of f, $\{c_{\mathbf{k}}(f)\}_{\mathbf{k}\in\mathbb{Z}^d}$, are concentrated in a given arbitrary finite set $\mathscr{I} \subset \mathbb{Z}^d$ so that

$$\min_{\Omega\subset\mathscr{I} \ s.t.\ |\Omega|=s} \left\| f - \sum_{\mathbf{k}\in\Omega} c_{\mathbf{k}}(f)\, \mathbb{e}^{-2\pi \mathbb{i}\mathbf{k}\cdot\circ} \right\|_2 < \epsilon \|f\|_2$$

holds for $s \ll |\mathscr{I}|$ and $\epsilon \in (0, 1)$ small. In such cases we aim to both identify a near-minimizing subset $\Omega \subset \mathscr{I}$ and accurately approximate its associated Fourier coefficients $\{c_{\mathbf{k}}(f)\}_{\mathbf{k}\in\Omega}$ as rapidly as possible. In this paper we present both deter-

Communicated by Simon Foucart.

C. Gross and M. Iwen were supported in part by the National Science Foundation DMS 1912706. T. Volkmer was supported in part by Sächsische Aufbaubank-Förderbank-(SAB) 100378180.

✉ Craig Gross
grosscra@msu.edu

Mark Iwen
markiwen@math.msu.edu

Lutz Kämmerer
lutz.kaemmerer@mathematik.tu-chemnitz.de

Toni Volkmer
toni.volkmer@math.tu-chemnitz.de

[1] Department of Mathematics, Michigan State University, East Lansing, USA

[2] Department of Mathematics and Department of Computational Mathematics, Science and Engineering (CMSE), Michigan State University, East Lansing, USA

[3] Faculty of Mathematics, Chemnitz University of Technology, Chemnitz, Germany

Published online: 13 December 2021

Birkhäuser

ministic and explicit as well as randomized algorithms for solving this problem using $\mathscr{O}(s^2 d \log^c(|\mathscr{I}|))$-time/memory and $\mathscr{O}(sd \log^c(|\mathscr{I}|))$-time/memory, respectively. Most crucially, all of the methods proposed herein achieve these runtimes while simultaneously satisfying theoretical best s-term approximation guarantees which guarantee their numerical accuracy and robustness to noise for general functions. These results are achieved by modifying several different one-dimensional Sparse Fourier Transform (SFT) methods to subsample a function along a reconstructing rank-1 lattice for the given frequency set $\mathscr{I} \subset \mathbb{Z}^d$ in order to rapidly identify a near-minimizing subset $\Omega \subset \mathscr{I}$ as above without having to use anything about the lattice beyond its generating vector. This requires the development of new fast and low-memory frequency identification techniques capable of rapidly recovering vector-valued frequencies in $\mathbb{Z}^d$ as opposed to recovering simple integer frequencies as required in the univariate setting. Two different multivariate frequency identification strategies are proposed, analyzed, and shown to lead to their own best s-term approximation methods herein, each with different accuracy versus computational speed and memory tradeoffs.

Keywords Multivariate Fourier approximation · Approximation algorithms · Fast Fourier transforms · Sparse Fourier transforms · Rank-1 lattices · Fast algorithms

Mathematics Subject Classification 65T40 · 65D15 · 42B05 · 65Y20 · 65T50

1 Introduction

This paper considers methods for efficiently computing sparse Fourier transforms of multivariate periodic functions using rank-1 lattices. In particular, for a function $f : \mathbb{T}^d \to \mathbb{C}$ (where $\mathbb{T} := [0, 1]$ with the endpoints identified), our goal is to compute the Fourier coefficients of f,

$$c_{\mathbf{k}}(f) := \int_{\mathbb{T}^d} f(\mathbf{x})\, \mathrm{e}^{-2\pi \mathrm{i} \mathbf{k} \cdot \mathbf{x}}\, \mathrm{d}\mathbf{x},$$

via samples of f at points in $\mathbb{T}^d$. Here, we assume that f is from the Wiener algebra $\mathscr{W}(\mathbb{T}^d) := \{f \in L^1(\mathbb{T}^d) \colon \|f\|_{\mathscr{W}(\mathbb{T}^d)} := \sum_{\mathbf{k} \in \mathbb{Z}^d} |c_{\mathbf{k}}(f)| < \infty\}$, and that f is well approximated by just a few of the dominant terms in its Fourier expansion (i.e., has an accurate sparse approximation in the Fourier basis).

One quasi-Monte Carlo approach which is especially popular in the context of Fourier approximations is sampling along rank-1 lattices adapted to frequency spaces of interest [17,20,23,24,26,29,30,35,36]. In the standard rank-1 lattice approach, a one-dimensional, length-M discrete Fourier transform (DFT) is applied to samples of f along a rank-1 lattice $\Lambda(\mathbf{z}, M)$ with generating vector $\mathbf{z} \in \mathbb{Z}^d$ over $\mathbb{T}^d$ defined by

$$\Lambda(\mathbf{z}, M) := \left\{ \frac{j}{M} \mathbf{z} \bmod 1 \mid j \in [M] := \{0, 1, \ldots, M-1\} \right\}.$$

If we write these lattice samples as $\mathbf{f} := (f(j\mathbf{z}/M \bmod 1))_{j\in[M]}$ this DFT then has the form

$$(\mathbf{F}_M\,\mathbf{f})_\omega = \frac{1}{M}\sum_{j\in[M]} f\left(\frac{j}{M}\mathbf{z} \bmod 1\right) \mathrm{e}^{-2\pi \mathrm{i}\omega j/M}. \tag{1}$$

Writing any function $f \in \mathcal{W}(\mathbb{T}^d)$ as a Fourier series, $f = \sum_{\mathbf{k}\in\mathbb{Z}^d} c_{\mathbf{k}}(f)\,\mathrm{e}^{2\pi\mathrm{i}\mathbf{k}\cdot\circ}$, the lattice DFT, (1) is exactly equivalent to the DFT of the univariate function

$$a(t) := \sum_{\mathbf{k}\in\mathbb{Z}^d} c_{\mathbf{k}}(f)\,\mathrm{e}^{2\pi\mathrm{i}\mathbf{k}\cdot\mathbf{z}t} \tag{2}$$

using the equispaced samples $\big(a(j/M)\big)_{j\in[M]} = \mathbf{f}$. Just as the DFT of equispaced samples of a can be used to approximate its Fourier coefficients, so then can this DFT be used to help approximate the original Fourier coefficients $c_{\mathbf{k}}(f)$ of f. This multivariate to univariate transformation allows us to carry over many standard one-dimensional DFT results in a straightforward manner.

However, in order for our univariate DFT to be properly related to the original multivariate Fourier coefficients $c_{\mathbf{k}}(f)$, any aliasing must not produce collisions which disrupt the multivariate to univariate transformation. Specifically, after applying a length-M DFT to the univariate function a in (2), all one-dimensional frequencies $\mathbf{k}\cdot\mathbf{z}$ are aliased to their residues modulo M. To avoid the disruptive aliasing collisions, we use a special class of rank-1 lattices. For the remainder of the paper, we will restrict our attention to finding energetic Fourier coefficients in some chosen finite multivariate frequency set $\mathcal{I} \subset \mathbb{Z}^d$.

Definition 1 (Reconstructing rank-1 lattice) Given a finite multivariate frequency set $\mathcal{I} \subset \mathbb{Z}^d$, we say that $\Lambda(\mathbf{z}, M)$ is a *reconstructing rank-1 lattice* for $\mathcal{I}$ if the mapping $m_{\mathbf{z},M} : \mathcal{I} \to [M]$ given by $\mathbf{k} \mapsto \mathbf{k}\cdot\mathbf{z} \bmod M$ is injective.

When sampling along a reconstructing rank-1 lattice, each coefficient produced by the DFT of a can be uniquely mapped back to the corresponding multivariate frequency $\mathbf{k}$ of f by inverting $m_{\mathbf{z},M}$.

With this setup in hand, we can now clarify the main two bottlenecks that this paper addresses. First, in general, reconstructing rank-1 lattices require lengths M satisfying $|\mathcal{I}| \le M \le |\mathcal{I}|^2$ (see e.g. [30, Theorem 8.16]). For large search spaces of multivariate frequencies $\mathcal{I}$ such as the full cube $\mathcal{I} = \big((-\lceil\frac{N}{2}\rceil, \lfloor\frac{N}{2}\rfloor] \cap \mathbb{Z}\big)^d$, a length-$M$ DFT suffers from the curse of dimensionality. Even for frequency spaces that depend moderately on the dimension such as a hyperbolic cross (see Sect. 5), M may be significantly inflated in comparison to $|\mathcal{I}|$. For example, the well-chosen rank-1 lattice for the hyperbolic cross in Sect. 5.1.1 is approximately 45 times the cardinality of $\mathcal{I}$. See, e.g., [18, Section 5] for more information about oversampling related to rank-1 lattices.

Our solution to intractable or over-inflated DFTs makes use of our assumption that f is Fourier sparse or compressible. Since the rank-1 lattice mapping allows us to make use of one-dimensional results, one-dimensional sparse Fourier transform (SFT) techniques [1,2,7,10–12,14–16,22,25,27,31,32,34] become particularly appealing. In general, they are able to sidestep runtimes which depend polynomially on the bandwidth, in this case M, and instead run sublinearly in the magnitude of the underlying frequency space under consideration. Additionally, these techniques often furnish recovery guarantees for Fourier compressible functions in terms of best s-term approximations in the same vein as compressed sensing results [8,9].

Our second bottleneck is the fact that after we find the significant Fourier coefficients of the one-dimensional function a, we must invert the frequency mapping $m_{\mathbf{z},M}$ in Definition 1 to relate these to the coefficients of f. In order to know or store this inverse map we require the calculation of $m_{\mathbf{z},M}(\mathscr{I})$. When we consider a function with a sparse Fourier series however, any benefit in using an SFT to calculate the DFT of samples along the lattice is lost in comparison to the $\mathscr{O}(d\,|\mathscr{I}|)$ size and operation count of the inverse computation.

The methods given in this paper instead use samples along possibly larger lattices to produce a sparse approximation of the Fourier transform of f without directly inverting $m_{\mathbf{z},M}$. The two algorithms considered below are able to operate on SFTs of manipulations of a in order to relate the univariate coefficients to their multivariate counterparts in $o(|\mathscr{I}|)$-time. This allows the methods developed herein to run faster and with less memory than it takes to simply enumerate the frequency set $\mathscr{I}$ and/or store $m_{\mathbf{z},M}(\mathscr{I})$ whenever f has a sufficiently accurate sparse approximation.

1.1 Prior work

Much recent work has considered the problem of quickly recovering both exactly sparse multivariate trigonometric polynomials as well as approximating more general functions by sparse trigonometric polynomials using dimension-incremental approaches [5,6,33,37]. These methods recover multivariate frequencies adaptively by searching lower-dimensional projections of $\mathscr{I} \subset \left(\left(-\left\lceil \frac{N}{2} \right\rceil, \left\lfloor \frac{N}{2} \right\rfloor\right] \cap \mathbb{Z}\right)^d$ for energetic frequencies. These lower dimensional candidate sets are then paired together to build up a fully d-dimensional search space smaller than the original one, which is expected to support the most energetic frequencies (see e.g., [19, Section 3] and the references within for a general overview).

In the context of Fourier methods, lattice-based techniques do a good job of support identification on the intermediary, lower-dimensional candidate sets, and especially recently, techniques based on multiple rank-1 lattices have shown success [19,21]. Though the total complexity in each of these steps is manageable and can be kept linear in the sparsity, these steps must be repeated in general to ensure that no potential frequencies have been left out. In particular, this results in at least $\mathscr{O}(ds^2N)$ operations (up to logarithmic factors) for functions supported on arbitrary frequency sets in order to obtain approximations that are guaranteed to be accurate with high probability. Though from an implementational perspective, this runtime can be mitigated by completing many of the repetitions and initial one-dimensional searches in parallel, once pairing begins, the results of previous iterations must be synchronized and communicated to future steps, necessitating serial interruptions.

Other earlier works include [16] in which previously existing univariate SFT results [15,34] are refined and adapted to the multivariate setting. Though the resulting complexity on the dimension is well above the dimension-incremental approaches, deterministic guarantees are given for multivariate Fourier approximation in $\mathcal{O}(d^4s^2)$ (up to logarithmic factors) time and memory, as well as a random variant which drops to linear scaling in s, leading to a runtime on the order of $\mathcal{O}(d^4s)$ with respect to s and d. Additionally, the compressed sensing type guarantees in terms of Fourier compressibility of the function under consideration carry over from the univariate SFT analysis. The scheme essentially makes use of a reconstructing rank-1 lattice on a superset of the full integer cube $\mathcal{I} = \left((-\lceil\frac{dN}{2}\rceil, \lfloor\frac{dN}{2}\rfloor] \cap \mathbb{Z}\right)^d$ with certain number theoretic properties that allow for fast inversion of the resulting one-dimensional coefficients by the Chinese Remainder Theorem. We note that this necessarily inflated frequency domain accounts for the suboptimal scaling in d above.

In [28], another fully deterministic sampling strategy and reconstruction algorithm is given. Like [16] though, the method can only be applied to Fourier approximations over an ambient frequency space $\mathcal{I}$ which is a full d-dimensional cube. Moreover, the vector space structure exploited to construct the sampling sets necessitates that the side length N of this cube is the power of a prime. However, the benefits to this construction are among the best considered so far: the method is entirely deterministic, has noise-robust recovery guarantees in terms of best s-term estimates, the sampling sets used are on the order of $\mathcal{O}(d^3s^2N)$, and the reconstruction algorithm's runtime complexity is on the order of $\mathcal{O}(d^3s^2N^2)$ both up to logarithmic factors. On the other hand, this algorithm still does not scale linearly in s.

Finally, we discuss [3,4], a pair of papers detailing high-dimensional Fourier recovery algorithms which offer a simplified (and therefore faster) approach to lattice transforms and dimension-incremental methods. These algorithms make heavy use of a one-dimensional SFT [7,25] based on a phase modulation approach to discover energetic frequencies in a fashion similar to our Algorithm 1 below. The main idea is to recover entries of multivariate frequencies by using equispaced evaluations of the function along a coordinate axis as well as samples of the function at the same points slightly shifted (the remaining dimensions are generally ignored). This shift in space produces a modulation in frequency from which frequency data can be recovered (cf. (3) and Algorithm 1 below). By supplementing this approach with simple reconstructing rank-1 lattice analysis for repetitions of the full integer cube, the runtime and number of samples are given on average as $\mathcal{O}(ds)$ up to logarithmic factors.

However, due to the possibility of collisions of multivariate frequencies under the hashing algorithms employed, these results hold only for random signal models. In particular, theoretical results are only stated for functions with randomly generated Fourier coefficients on the unit circle with randomly chosen frequencies from a given frequency set. Additionally, the analysis of these techniques assumes that the algorithm applied to the randomly generated signal does not encounter certain low probability (with respect to the random signal model considered therein) energetic frequency configurations. Furthermore, the method is restricted in stability, allowing for spatial shifts in sampling bounded by at most the reciprocal of the side length of the multivariate frequency cube under consideration, and only exact recovery is considered (or recovery up to factors related to sample corruption by gaussian noise in [3]). In addition, no results given are proven concerning the approximation of more general periodic functions, e.g., compressible functions.

1.2 Main contributions

We begin with a brief summary of the benefits provided by our approach in comparison to the methods discussed above. Below, we ignore logarithmic factors in our summary of the runtime/sampling complexities.

– All variants, deterministic and random, of both algorithms presented in this paper have runtime and sampling complexities **linear in** d with best s-term estimates for **arbitrary signals**. This is in contrast to the complexities of dimension-incremental approaches [5,6,19,21] and the number theoretic approaches [16,28] while still achieving similarly strong best s-term guarantees.
– Both algorithms proposed herein have randomized variants with runtime and sampling complexities **linear in** s with best s-term estimates on **arbitrary signals** that hold **with high probability**. Thus, the randomized methods proposed in this paper achieve the efficient runtime complexities of [3,4] while simultaneously exhibiting best s-term approximation guarantees for general periodic functions thereby improving on the non-deterministic dimension incremental approaches [5,6,19,21].
– Both algorithms proposed herein have a deterministic variant with runtime and sampling complexities **quadratic in** s with best s-term estimates on **arbitrary signals** that also hold **deterministically**. This is in contrast to all previously discussed methods without deterministic guarantees, [3–6,19,21], as well as improving on prior deterministic results [16,28] for functions whose energetic frequency support sets $\mathscr{I}$ are smaller than the full cube.

Overview of the methods and related theory

We will build on the fast and potentially deterministic one-dimensional SFT from [16] and its discrete variant from [27] by applying those techniques along rank-1 lattices. As previously discussed, the primary difficulty in doing so is matching energetic one-dimensional Fourier coefficients with their d-dimensional counterparts. We are especially interested in doing this in an efficient and provably accurate way. We propose and analyze two different methods for solving this problem herein.

The first frequency identification approach, Algorithm 1, involves modifications of the phase shifting technique from [3,4,7,25]. We make use of the translation to modulation property of the Fourier transform (cf. (3) below) observed in these works to extract out frequency data. Combining this with SFTs on rank-1 lattices gives a new class of fast methods with several benefits. Notably, we are able to maintain error guarantees for any function (not just random signals) in terms of best Fourier s-term approximations. Additionally, we factor the instability and potential for collisions from [3,4] into these best s-term approximations. The only downside in our estimates is an additional linear factor of N multiplying the terms commonly seen in standard error bounds (cf. Corollaries 1, 2). However, we are able to maintain deterministic results with runtime and sampling complexities that are quadratic in s, as well as results for random variants with complexities that are linear in s. Additionally, the dependence on the dimension d is reduced from $\mathscr{O}(d^4)$ in [16] to only $\mathscr{O}(d)$.

Our second technique in Algorithm 2 uses a different approach to applying SFTs to modifications of the multivariate function along a reconstructing rank-1 lattice. Effectively, we reduce f to a two-dimensional function. This is done by mapping all but one dimension, say ℓ, down to one using a rank-1 lattice, and leaving the ℓ dimension free. From here, we take a two-dimensional DFT (taking care to use SFTs where possible). The locations of Fourier coefficients in this two-dimensional DFT can then be used to determine the ℓth coordinate of the frequency data. This is then repeated for each dimension $\ell \in \{1, \dots d\}$.

This process is slower but more stable than Algorithm 1. In particular this produces more accurate best Fourier s-term approximation guarantees without the extraneous factor of N (cf. Corollaries 3, 4). The deterministic results still have a complexity quadratic in s with random extensions that are linear in s. However, we incur an extra quadratic factor of N in the complexity bounds (cf. Lemma 3).

We stress here that by compartmentalizing the translation from multivariate analysis to univariate analysis into the theory of rank-1 lattices, our techniques are suitable for any frequency set of interest $\mathscr{I}$. The only constraint is the necessity for a reconstructing rank-1 lattice for $\mathscr{I}$ (and potentially projections of $\mathscr{I}$ in the case of Algorithm 2). This flexibility improves the results from [16], primarily with respect to the polynomial factor of d in our runtime and sampling complexities. We remark that though the existence of the necessary reconstructing rank-1 lattice is a nontrivial requirement, there exist efficient construction algorithms for arbitrary frequency sets via deterministic component by component methods, see e.g., [18,23,30].

In terms of implementation, we note that the multivariate techniques we employ are entirely modular with respect to the univariate SFT used. As such, the complexity estimates and error bounds for our approaches in Sect. 4 are directly derived from the chosen SFT. Finally, the methods we present are trivially parallelizable so that in particular, a large majority of these univariate SFTs in Algorithm 1 or Algorithm 2 can occur in parallel.

1.3 Organization

The remainder of this paper is presented as follows: in Sect. 2, we set the notation, the notions of the Fourier transform, and the various types of manipulations we will be using in the sequel. Section 3 reviews and further refines the univariate SFTs from [16,27] which we will use in our multivariate techniques. Section 4 presents our main multivariate approximation algorithms and their analysis. In particular, Sect. 4.1 discusses the phase-shifting approach, while Sect. 4.2 discusses the two-dimensional SFT/DFT combination approach. Finally, we implement these two algorithms numerically and present the empirical results in Sect. 5.

2 Notation and assumptions

2.1 Multivariate

We begin by defining a one-dimensional frequency band of length N as $\mathscr{B}_N := \left(-\left\lceil\frac{N}{2}\right\rceil, \left\lfloor\frac{N}{2}\right\rfloor\right] \cap \mathbb{Z}$. For a potentially large but finite multivariate frequency set $\mathscr{I}$, which we think of as containing the most significant frequencies of the function under consideration, we choose $N = \max_{\ell\in[d]}(\max_{\mathbf{k}\in\mathscr{I}} k_\ell - \min_{\tilde{\mathbf{k}}\in\mathscr{I}} \tilde{k}_\ell) + 1$ as the minimal width such that $\mathscr{I} \subset \mathbf{h} + \mathscr{B}_N^d$ for some $\mathbf{h} \in \mathbb{Z}^d$. By appropriately modulating any multivariate function $f : \mathbb{T}^d \to \mathbb{C}$ under consideration, i.e., considering $\mathrm{e}^{-2\pi \mathrm{i}\mathbf{h}\cdot\circ} f$, we shift the frequencies of Fourier coefficients of f originally in $\mathscr{I}$ to $\mathscr{I} - \mathbf{h} \subset \mathscr{B}_N^d$. Thus, we assume without loss of generality that $\mathscr{I} \subset \mathscr{B}_N^d$ with N as above. Without loss of generality, we will also assume that for a reconstructing rank-1 lattice $\Lambda(\mathbf{z}, M)$, the generating vector satisfies $\mathbf{z} \in [M]^d$.

To avoid confusion with the hat notation which will be reserved for univariate functions below, we denote the sequence of all Fourier coefficients (i.e., the Fourier transform) of a periodic function $f : \mathbb{T}^d \to \mathbb{C}$ as $c(f) = (c_{\mathbf{k}}(f))_{\mathbf{k}\in\mathbb{Z}^d}$, also writing this as simply c when the function is clear from context. Its restriction to $\mathscr{I}$ is denoted $c(f)|_{\mathscr{I}} = (c_{\mathbf{k}}(f))_{\mathbf{k}\in\mathscr{I}}$, and the best s-term approximation, that is, its restriction to the support of the s-largest magnitude entries, is denoted c_s^{opt} or $(c|_{\mathscr{I}})_s^{\mathrm{opt}}$ on $\mathbb{Z}^d$ or $\mathscr{I}$, respectively. We denote multiindexed vectors only defined on finite index sets (which are not restrictions of infinitely indexed sequences) in boldface, e.g., $\mathbf{b} = (b_{\mathbf{k}})_{\mathbf{k}\in\mathscr{I}}$, as well as identify this multivariate vector as a one-dimensional vector $\mathbf{b} \in \mathbb{C}^{|\mathscr{I}|}$ via lexicographic ordering when dictated by context. Again dictated by context, we also extend these multiindexed vectors to larger index sets by setting them to zero outside of their original domain. For example, if $\mathbf{b} = (b_{\mathbf{k}})_{\mathbf{k}\in\mathscr{I}}$ and $c = (c_{\mathbf{k}})_{\mathbf{k}\in\mathbb{Z}^d}$,

$$\|\mathbf{b} - c\|_{\ell^1(\mathbb{Z}^d)} = \sum_{\mathbf{k}\in\mathscr{I}} |b_{\mathbf{k}} - c_{\mathbf{k}}| + \sum_{\mathbf{k}\in\mathbb{Z}^d\setminus\mathscr{I}} |c_{\mathbf{k}}|.$$

In the multivariate approaches which follow, we will also make use of the shift operator $S_{\ell,\alpha}$ in the ℓth coordinate with shift $\alpha \in \mathbb{R}$ defined by its action on the multivariate periodic function $f : \mathbb{T}^d \to \mathbb{C}$ as

$$S_{\ell,\alpha}(f)(x_1, \ldots, x_d) := f(x_1, \ldots, x_{\ell-1}, (x_\ell + \alpha) \bmod 1, x_{\ell+1}, \ldots, x_d).$$

When necessary, we will separate out coordinate ℓ of a multivariate point $\mathbf{x} \in \mathbb{T}^d$ or frequency $\mathbf{k} \in \mathbb{Z}^d$, denoting the remaining coordinates as $\mathbf{x}'_\ell \in \mathbb{T}^{d-1}$ or $\mathbf{k}'_\ell \in \mathbb{Z}^{d-1}$. With a slight abuse of notation, we can rewrite the original point or frequency as $\mathbf{x} = (x_\ell, \mathbf{x}'_\ell)$ or $\mathbf{k} = (k_\ell, \mathbf{k}'_\ell)$.

2.2 Univariate

For any univariate periodic function $a : \mathbb{T} \to \mathbb{C}$, we define the vector $\mathbf{a} \in \mathbb{C}^M$ as the vector of M equispaced samples of a on $\mathbb{T}$, that is, $\mathbf{a} = (a(j/M))_{j\in[M]}$. As in the multivariate case, we define the Fourier transform of $a : \mathbb{T} \to \mathbb{C}$ as the sequence $\hat{a} = (\hat{a}_\omega)_{\omega \in \mathbb{Z}}$ with

$$\hat{a}_\omega := \int_{\mathbb{T}} a(t)\, \mathbb{e}^{-2\pi \mathbb{i} \omega x}\, \mathrm{d}x \text{ for all } \omega \in \mathbb{Z}.$$

Additionally, we define the vector $\hat{\mathbf{a}} \in \mathbb{C}^M$ as the restriction of $\hat{a}$ to $\mathscr{B}_M$. If not explicitly stated, the length of the discretized function $\mathbf{a}$ and Fourier transform $\hat{\mathbf{a}}$ will be clear from context. Note that $\hat{\mathbf{a}}$ is not necessarily the discrete Fourier transform of $\mathbf{a}$, which we define as

$$(\mathbf{F}_M\, \mathbf{a})_\omega := \frac{1}{M} \sum_{j \in [M]} a_j\, \mathbb{e}^{-2\pi \mathbb{i} \omega j / M} = \frac{1}{M} \sum_{j \in [M]} a\left(\frac{j}{M}\right) \mathbb{e}^{-2\pi \mathbb{i} \omega j / M}, \text{ where}$$
$$\mathbf{F}_M := \left(\mathbb{e}^{-2\pi \mathbb{i} \omega j / M} / M\right)_{j \in [M],\, \omega \in \mathscr{B}_M}$$

is the discrete Fourier matrix. Our convention here and in the remainder of the paper is to use zero-based indexing which is always taken implicitly modulo the length of the dimension, e.g., $(\mathbf{F}_M)_{0,-1} = (\mathbf{F}_M)_{0,M-1}$.

For any vector $\mathbf{b} \in \mathbb{C}^M$, we denote its best s-term approximation $\mathbf{b}_s^{\mathrm{opt}}$, where as above, $\mathbf{b}_s^{\mathrm{opt}}$ is the restriction of $\mathbf{b}$ to its s largest magnitude entries. In the sequel, we always assume that our sparsity parameters s are at most half the size of the vectors under consideration so that, e.g., $\mathbf{b}_{2s}^{\mathrm{opt}}$ is well-defined. Additionally as above, vectors can also be compared with other vectors on larger index sets than they are defined by simply setting the smaller vectors to zero outside of their original domain.

As for one-dimensional approximations, we will be considering SFT algorithms which, given sparsity parameter s and bandwidth M, produce an s-sparse approximation of the Fourier transform of a function $a \in \mathscr{W}(\mathbb{T}) \cap C(\mathbb{T})$ restricted to $\mathscr{B}_M$, where $C(\mathbb{T})$ is the set of continuous functions on $\mathbb{T}$. Note that these are not necessarily discrete algorithms which take in $\mathbf{a}$ as input. In particular, in addition to our assumption that any function we wish to approximate is in the Wiener algebra $\mathscr{W}(\mathbb{T})$, we also require continuity so that the SFT can make use of point samples of the function at arbitrary locations in $\mathbb{T}$. We denote these algorithms $\mathscr{A}_{s,M} : \mathscr{W}(\mathbb{T}) \cap C(\mathbb{T}) \to \mathbb{C}^M$, which produce $\mathscr{A}_{s,M} a =: \mathbf{v} \in \mathbb{C}^M$ as approximations to $\hat{\mathbf{a}} \in \mathbb{C}^M$ using some fixed number of samples of a.

Finally, the argument of a complex number will be used in the implementation and analysis of Algorithm 1. We use the principal branch of the argument function so that for $w \in \mathbb{C}$, $\arg(w) \in (-\pi, \pi]$.

3 One-dimensional sparse Fourier transform results

Below, we summarize some of the previous work on one-dimensional sparse Fourier transforms which will be used in our multivariate algorithms. Rather than focus on the inner workings of these SFTs, we highlight four main properties concerning their recovery guarantees and computational complexity. This compartmentalization allows for any SFT satisfying these properties to be easily extended for multivariate Fourier recovery simply by plugging into Algorithms 1 and 2.

We first review the sublinear-time algorithm from [16] which uses fewer than M nonequispaced samples of a function. We refer the reader interested in its implementation and mathematical explanation to [16] as well as [15,34]. Below, we will use slightly improved error bounds over those in its original presentation. The proof of these improvements can be found in Appendix A.

Theorem 1 (Robust sublinear-time, nonequispaced SFT: [16], Theorem 7/[27], Lemma 4) *For a signal $a \in \mathcal{W}(\mathbb{T}) \cap C(\mathbb{T})$ corrupted by some arbitrary noise $\mu : \mathbb{T} \to \mathbb{C}$, Algorithm 3 of* [16], *denoted $\mathscr{A}^{\text{sub}}_{2s,M}$, will output a $2s$-sparse coefficient vector $\mathbf{v} \in \mathbb{C}^M$ which*

1. *reconstructs every frequency of $\hat{\mathbf{a}} \in \mathbb{C}^M$, $\omega \in \mathscr{B}_M$, with corresponding Fourier coefficients meeting the tolerance*

$$|\hat{a}_\omega| > (4 + 2\sqrt{2}) \left(\frac{\|\hat{\mathbf{a}} - \hat{\mathbf{a}}_s^{\text{opt}}\|_1}{s} + \|\hat{a} - \hat{\mathbf{a}}\|_1 + \|\mu\|_\infty \right),$$

2. *satisfies the ℓ^∞ error estimate for recovered coefficients*

$$\|(\hat{\mathbf{a}} - \mathbf{v})|_{\operatorname{supp}(\mathbf{v})}\|_\infty \leq \sqrt{2} \left(\frac{\|\hat{\mathbf{a}} - \hat{\mathbf{a}}_s^{\text{opt}}\|_1}{s} + \|\hat{a} - \hat{\mathbf{a}}\|_1 + \|\mu\|_\infty \right),$$

3. *satisfies the ℓ^2 error estimate*

$$\|\hat{\mathbf{a}} - \mathbf{v}\|_2 \leq \|\hat{\mathbf{a}} - \hat{\mathbf{a}}_{2s}^{\text{opt}}\|_2 + \frac{(8\sqrt{2} + 6)\|\hat{\mathbf{a}} - \hat{\mathbf{a}}_s^{\text{opt}}\|_1}{\sqrt{s}} \\ + (8\sqrt{2} + 6)\sqrt{s}(\|\hat{a} - \hat{\mathbf{a}}\|_1 + \|\mu\|_\infty),$$

4. *and the number of required samples of a and the operation count for $\mathscr{A}^{\text{sub}}_{2s,M}$ are*

$$\mathscr{O}\left(\frac{s^2 \log^4 M}{\log s} \right).$$

The Monte Carlo variant of $\mathscr{A}^{\text{sub}}_{2s,M}$, denoted $\mathscr{A}^{\text{sub,MC}}_{2s,M}$, referred to by Corollary 4 of [16] *satisfies all of the conditions* (1)–(3) *simultaneously with probability* $(1-\sigma) \in [2/3, 1)$ *and has number of required samples and operation count*

$$\mathscr{O}\left(s \log^3(M) \log\left(\frac{M}{\sigma} \right) \right).$$

The samples required by $\mathscr{A}^{\text{sub,MC}}_{2s,M}$ are a subset of those required by $\mathscr{A}^{\text{sub}}_{2s,M}$.

Remark 1 In the noiseless case, if the univariate function a is Fourier s-sparse, i.e., is a trigonometric polynomial and M is large enough such that $\operatorname{supp}(\hat{a}) \subset \mathscr{B}_M$, both $\mathscr{A}_{2s,M}^{\text{sub}}$ and $\mathscr{A}_{2s,M}^{\text{sub,MC}}$ will exactly recover $\hat{\mathbf{a}}$ (the latter with probability $1-\sigma$), and therefore $\hat{a}$. In particular, note that the output of either algorithm will then actually be s-sparse.

Using the above SFT algorithm with the discretization process outlined in [27] leads to a fully *discrete* sparse Fourier transform, requiring only equispaced samples of a. However, rather than separately accounting for the truncation to the frequency band $\mathscr{B}_M$ as above, the equispaced samples allow us to take advantage of aliasing, which is particularly important when we apply the algorithm along reconstructing rank-1 lattices. Thus, instead of approximating $\hat{\mathbf{a}} \in \mathbb{C}^M$, the restriction of $\hat{a}$ to $\mathscr{B}_M$, as above, we prefer to approximate the discrete Fourier transform of $\mathbf{a}$.

Eventually, we will consider techniques for approximation of arbitrary periodic functions rather than simply polynomials. For this reason, we require noise-robust recovery results for the method in [27]. The necessary modifications to account for this robustness as well as the improved guarantees carried over from the previous algorithm are given in Appendix B. The upshot is that we are able to state four properties of this SFT analogous to those in Theorem 1 which allow for modular proofs of the multivariate results later on.

Theorem 2 (Robust discrete sublinear-time SFT: see [27], Theorem 5) *For a signal $a \in \mathscr{W}(\mathbb{T}) \cap C(\mathbb{T})$ corrupted by some arbitrary noise $\mu : \mathbb{T} \to \mathbb{C}$, and $1 \leq r \leq \frac{M}{36}$, Algorithm 1 of [27], denoted $\mathscr{A}_{2s,M}^{\text{disc}}$, will output a $2s$-sparse coefficient vector $\mathbf{v} \in \mathbb{C}^M$ which*

1. *reconstructs every frequency of $\mathbf{F}_M\,\mathbf{a} \in \mathbb{C}^M$, $\omega \in \mathscr{B}_M$, with corresponding aliased Fourier coefficient meeting the tolerance*

$$|(\mathbf{F}_M\,\mathbf{a})_\omega| > 12(1+\sqrt{2})\left(\frac{\|\mathbf{F}_M\,\mathbf{a} - (\mathbf{F}_M\,\mathbf{a})_s^{\text{opt}}\|_1}{2s} + 2(\|\mathbf{a}\|_\infty M^{-r} + \|\boldsymbol{\mu}\|_\infty)\right),$$

2. *satisfies the ℓ^∞ error estimate for recovered coefficients*

$$\|(\mathbf{F}_M\,\mathbf{a} - \mathbf{v})|_{\operatorname{supp}(\mathbf{v})}\|_\infty \leq 3\sqrt{2}\left(\frac{\|\mathbf{F}_M\,\mathbf{a} - (\mathbf{F}_M\,\mathbf{a})_s^{\text{opt}}\|_1}{2s} + 2(\|\mathbf{a}\|_\infty M^{-r} + \|\boldsymbol{\mu}\|_\infty)\right),$$

3. *satisfies the ℓ^2 error estimate*

$$\|\mathbf{F}_M\,\mathbf{a} - \mathbf{v}\|_2 \leq \|\mathbf{F}_M\,\mathbf{a} - (\mathbf{F}_M\,\mathbf{a})_{2s}^{\text{opt}}\|_2 + 38\frac{\|\mathbf{F}_M\,\mathbf{a} - (\mathbf{F}_M\,\mathbf{a})_s^{\text{opt}}\|_1}{\sqrt{s}} + 152\sqrt{s}(\|\mathbf{a}\|_\infty M^{-r} + \|\boldsymbol{\mu}\|_\infty),$$

4. *and the number of required samples of $\mathbf{a}$ and the operation count for $\mathscr{A}_{2s,M}^{\text{disc}}$ are*

$$\mathscr{O}\left(\frac{s^2 r^{3/2} \log^{11/2} M}{\log s}\right).$$

The Monte Carlo variant of $\mathscr{A}^{\text{disc}}_{2s,M}$*, denoted* $\mathscr{A}^{\text{disc,MC}}_{2s,M}$*, satisfies all of the conditions (1)–(3) simultaneously with probability* $(1-\sigma) \in [2/3, 1)$ *and has number of required samples and operation count*

$$\mathscr{O}\left(sr^{3/2}\log^{9/2}(M)\log\left(\frac{M}{\sigma}\right)\right).$$

4 Fast multivariate sparse Fourier transforms

Having detailed two sublinear-time, one-dimensional SFT algorithms, we are now prepared to extend these to the multivariate setting. The general approach will be to apply the one-dimensional methods to transformations of our multivariate function of interest with samples taken along rank-1 lattices. The particular approaches for transforming our multivariate function will then allow for the efficient extraction of multidimensional frequency information for the most energetic coefficients identified by univarate SFTs. In particular, our first approach considered in Sect. 4.1 successively shifts the function in each dimension, whereas our second approach considered in Sect. 4.2 successively collapses all but one dimension along a rank-1 lattice and samples the resulting two-dimensional function.

Since the two approaches in Algorithms 1 and 2 below can make use of any univariate SFT algorithm, their analysis will be presented in a modular fashion below. Each algorithm is followed by a lemma (Lemmas 1 and 3 respectively) which provides associated error guarantees when any sufficiently accurate univariate SFT $\mathscr{A}_{s,M}$ is employed. The lemmas are then each followed by two corollaries (Corollaries 1, 2, 3 and 4 respectively) where we apply the lemma to the two example univariate SFTs reviewed in Sect. 3 specified by Theorems 2 and 1.

4.1 Phase encoding

We begin by noting that this section makes significant use of the property of the Fourier transform that translation of a function modulates its Fourier coefficients. In our notation, we have that for any dimension $\ell \in [d]$ and shift $\alpha \in \mathbb{R}$, applying the shift operator $S_{\ell,\alpha}$ to $f : \mathbb{T}^d \to \mathbb{C}$ will modulate the Fourier coefficients as

$$c_{\mathbf{k}}(S_{\ell,\alpha} f) = \mathrm{e}^{2\pi \mathrm{i} k_\ell \alpha} c_{\mathbf{k}}(f). \tag{3}$$

The main idea of our phase encoding approach in Algorithm 1 is that by exploiting this spatial translation property, we can separate out the components of recovered frequencies in modulations of the function's Fourier coefficients. Before stating the algorithm in detail, we begin with a simple example.

Example 1 (Phase encoding on a trigonometric monomial) Let $d = 2$. Suppose that $f(\mathbf{x}) = \mathrm{e}^{2\pi \mathrm{i}\mathbf{k}\cdot\mathbf{x}}$ is a trigonometric monomial with single frequency $\mathbf{k} \in \mathscr{I} \subset \mathbb{Z}^2$ for some known, potentially large $\mathscr{I}$. Given $\Lambda(\mathbf{z}, M)$, a reconstructing rank-1 lattice for $\mathscr{I}$, we consider the one-dimensional restriction of f to the lattice $a(t) := f(t\mathbf{z})$ as

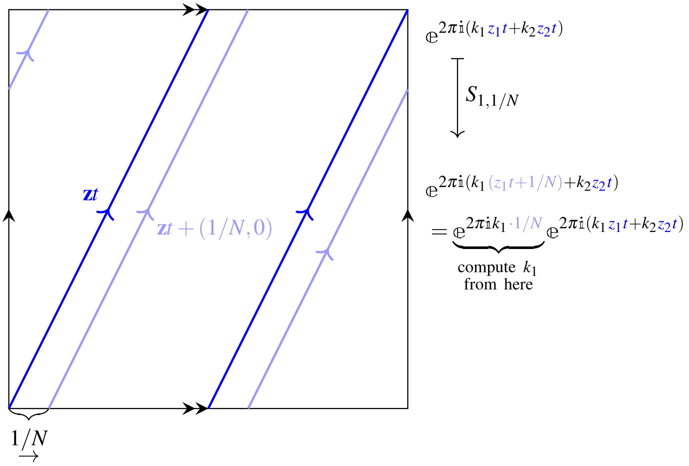

$\mathbb{e}^{2\pi\mathbb{i}(k_1 z_1 t+k_2 z_2 t)}$
$S_{1,1/N}$
$\mathbb{e}^{2\pi\mathbb{i}(k_1 (z_1 t+1/N)+k_2 z_2 t)}$
$= \underbrace{\mathbb{e}^{2\pi\mathbb{i}k_1 \cdot 1/N}}_{\text{compute } k_1 \text{ from here}} \mathbb{e}^{2\pi\mathbb{i}(k_1 z_1 t+k_2 z_2 t)}$
$\mathbf{z}t$
$\mathbf{z}t + (1/N, 0)$
$1/N$
$\rightarrow$

Fig. 1 The basic procedure for the phase encoding algorithm applied to the trigonometric monomial $f(\mathbf{x}) = \mathbb{e}^{2\pi\mathbb{i}\mathbf{k}\cdot\mathbf{x}}$

in (2). Since f is Fourier-sparse, a lattice DFT on a is unnecessarily expensive. Thus, applying a much faster SFT to a returns $\hat{a}_{\mathbf{k}\cdot\mathbf{z} \bmod M} = 1$. Our goal is to match this coefficient of a to the correct Fourier coefficient of f without having to search all of $\mathscr{I}$.

Figure 1 depicts the phase encoding method we use in Algorithm 1 below. In order to compute a, we restrict f to the blue line in this figure, $\mathbf{z}t$. However, to get extra information about $\mathbf{k}$, we restrict $S_{1,1/N} f$, a shift of f in the first coordinate by $1/N$, to the lattice. The shifted lattice that we effectively restrict f to, $\mathbf{z}t + (1/N, 0)$, is depicted in light blue.

The resulting modulation of f induced by this spatial shift (as described by (3)) is detailed in the remainder of Fig. 1. Thus, defining $a^1(t) := S_{1,1/N} f(\mathbf{z}t)$, an SFT would discover $\hat{a}^1_{\mathbf{k}\cdot\mathbf{z} \bmod M} = \mathbb{e}^{2\pi\mathbb{i}k_1/N}$. We then can extract k_1 from this modulation.

Repeating this process in the $\ell = 2$ coordinate will recover k_2, and therefore, the entirety of $\mathbf{k}$ is recovered by using $d = 2$ SFTs. From here, we can then match $\hat{a}_{\mathbf{k}\cdot\mathbf{z} \bmod M} = 1$ to $c_{\mathbf{k}}(f)$ in faster than $\mathscr{O}(|\mathscr{I}|)$ time and memory as desired.

In the language of Algorithm 1, the original SFT of a occurs on Line 1. The SFTs of the shifts of f, $a^1, \ldots, a^d$, occur on Line 3. In this example, we considered a function with only one significant Fourier mode, however, we will generally recover s significant Fourier modes from the SFT algorithm. Thus, the for loop from Line 6 to Line 13 considers each of these recovered one-dimensional frequencies separately. Line 8 computes the modulation induced by each of the d shifts, then extracts each coordinate of the d-dimensional frequency. The remaining check on Line 10 is useful for the theoretical analysis to ensure that only correctly recovered frequencies are considered.

Algorithm 1 Simple Frequency Index Recovery by Phase Encoding

```
Input: A multivariate periodic function $f \in \mathcal{W}(\mathbb{T}^d) \cap C(\mathbb{T}^d)$ (from which we are able to obtain potentially
    noisy samples), a multivariate frequency set $\mathcal{I} \subset \mathcal{B}_N^d$, a reconstructing rank-1 lattice $\Lambda(\mathbf{z}, M)$ for $\mathcal{I}$,
    and an SFT algorithm $\mathcal{A}_{s,M}$.
Output: Sparse coefficient vector $\mathbf{b} = (b_{\mathbf{k}})_{\mathbf{k} \in \mathcal{B}_N^d}$ (optionally supported on $\mathcal{I}$, see Line 10), an approxi-
    mation to $\left(c|_{\mathcal{I}}\right)_s^{\mathrm{opt}}$.
1: Apply $\mathcal{A}_{s,M}$ to the univariate restriction of $f$ to the lattice, $a(t) = f(t\mathbf{z})$, to produce $\mathbf{v} = \mathcal{A}_{s,M} a$, a
    sparse approximation of $\mathbf{F}_M\, \mathbf{a} \in \mathbb{C}^M$.
2: for all $\ell \in [d]$ do
3:     Apply $\mathcal{A}_{s,M}$ to $a^\ell(t) = S_{\ell,1/N} f(t\mathbf{z})$ to produce $\mathbf{v}^\ell = \mathcal{A}_{s,M} a^\ell$, a sparse approximation of $\mathbf{F}_M\, \mathbf{a}^\ell \in$
       $\mathbb{C}^M$.
4: end for
5: $\mathbf{b} \leftarrow \mathbf{0}$
6: for all $\omega \in \mathrm{supp}(\mathbf{v}) \subset \mathcal{B}_M$ do
7:     for all $\ell \in [d]$ do
8:         $(k_\omega)_\ell \leftarrow \mathrm{round}(N \arg(v_\omega^\ell / v_\omega)/2\pi)$
9:     end for
10:    if $\mathbf{k}_\omega \cdot \mathbf{z} \equiv \omega \pmod{M}$ (and optionally $\mathbf{k}_\omega \in \mathcal{I}$; see Remark 2) then
11:        $b_{\mathbf{k}_\omega} \leftarrow b_{\mathbf{k}_\omega} + v_\omega$
12:    end if
13: end for
```

4.1.1 Analysis of Algorithm 1

Having seen the phase encoding approach of Algorithm 1 in action, we now provide an error guarantee for its output. Notice that the assumptions on the SFT necessary for this theoretical analysis are exactly those provided by Theorems 1 and 2 .

Lemma 1 (General recovery result for Algorithm 1) *Let $\mathcal{A}_{s,M}$ in the input to Algorithm 1 be a noise-robust SFT algorithm which, for a function $a \in \mathcal{W}(\mathbb{T}) \cap C(\mathbb{T})$ corrupted by some arbitrary noise $\mu : \mathbb{T} \to \mathbb{C}$, constructs an s-sparse Fourier approximation $\mathcal{A}_{s,M}(a + \mu) =: \mathbf{v} \in \mathbb{C}^M$ which*

1. *reconstructs every frequency (up to s many) of $\mathbf{F}_M\, \mathbf{a} \in \mathbb{C}^M$, $\omega \in \mathcal{B}_M$, with corresponding Fourier coefficient meeting the tolerance $|(\mathbf{F}_M\, \mathbf{a})_\omega| > \tau$,*
2. *satisfies the ℓ^∞ error estimate for recovered coefficients*

$$\left\| (\mathbf{F}_M\, \mathbf{a} - \mathbf{v})|_{\mathrm{supp}(\mathbf{v})} \right\|_\infty \le \eta_\infty < \tau,$$

3. *satisfies the ℓ^2 error estimate*

$$\|\mathbf{F}_M\, \mathbf{a} - \mathbf{v}\|_2 \le \eta_2,$$

4. *and requires $\mathcal{O}(P(s, M))$ total evaluations of a, operating with computational complexity $\mathcal{O}(R(s, M))$.*

Additionally, assume that the parameters τ and η_∞ hold uniformly for each SFT performed in Algorithm 1.

Let f, $\mathscr{I}$, and $\Lambda(\mathbf{z}, M)$ be as specified in the input to Algorithm 1. Collecting the τ-significant frequencies of f into the set $\mathscr{S}_\tau := \{\mathbf{k} \in \mathscr{I} \mid |c_{\mathbf{k}}(f)| > \tau\}$, assume that $|\mathscr{S}_\tau| \leq s$, and set

$$\beta = \max\left(\tau, \eta_\infty\left(1 + \frac{2}{\sin\left(\frac{\pi}{N}\right)}\right)\right).$$

Then Algorithm 1 (ignoring the optional check on Line 10) will produce an s-sparse approximation $\mathbf{b}$ of the Fourier coefficients of f satisfying the error estimate

$$\|\mathbf{b} - c(f)\|_{\ell^2(\mathbb{Z}^d)} \leq \eta_2 + (\beta + \eta_\infty)\sqrt{\max(s - |\mathscr{S}_\beta|, 0)} + \|c(f)|_{\mathscr{I}} - c(f)|_{\mathscr{S}_\beta}\|_{\ell^2(\mathbb{Z}^d)} + \|c(f) - c(f)|_{\mathscr{I}}\|_{\ell^2(\mathbb{Z}^d)}$$

requiring $\mathscr{O}\left(d \cdot P(s, M)\right)$ total evaluations of f, in $\mathscr{O}\left(d \cdot (R(s, M) + s)\right)$ total operations.

Proof We begin by assuming that f is a trigonometric polynomial with $\operatorname{supp}(c(f)) \subset \mathscr{I}$.

Since $\Lambda(\mathbf{z}, M)$ is a reconstructing rank-1 lattice for $\mathscr{I}$, there are no collisions among the one-dimensional frequencies $\{\mathbf{k} \cdot \mathbf{z} \mid \mathbf{k} \in \mathscr{I}\}$ modulo M. Setting $f(t\mathbf{z}) = a(t)$ as in (2) then ensures that for each $\mathbf{k} \in \mathscr{I}$, $c_{\mathbf{k}}(f) = \hat{a}_{\mathbf{k}\cdot\mathbf{z}}$. Since there are no frequency collisions in the lattice DFT, we also have $c_{\mathbf{k}}(f) = (\mathbf{F}_M\, \mathbf{a})_{\mathbf{k}\cdot\mathbf{z} \bmod M}$. Thus, by assumption 1 on the SFT algorithm $\mathscr{A}_{s,M}$, Lines 1 and 3 of Algorithm 1 will produce coefficient estimates of $c_{\mathbf{k}}(f)$ for every $\mathbf{k} \in \mathscr{S}_\tau$. We then write these SFT approximations as $v_{\mathbf{k}\cdot\mathbf{z} \bmod M} = c_{\mathbf{k}}(f) + \eta_{\mathbf{k}}$ and $v^\ell_{\mathbf{k}\cdot\mathbf{z} \bmod M} = \mathbb{e}^{2\pi \mathbb{i} k_\ell/N}(c_{\mathbf{k}}(f) + \eta^\ell_{\mathbf{k}})$ respectively, where we have made use of (3). Note that $|\eta_{\mathbf{k}}|, |\eta^\ell_{\mathbf{k}}| \leq \eta_\infty$. Now, considering the estimate for k_ℓ, we have

$$\begin{aligned} \frac{N}{2\pi}\arg\left(\frac{v^\ell_{\mathbf{k}\cdot\mathbf{z} \bmod M}}{v_{\mathbf{k}\cdot\mathbf{z} \bmod M}}\right) &= \frac{N}{2\pi}\arg\left(\mathbb{e}^{2\pi \mathbb{i} k_\ell/N}\frac{c_{\mathbf{k}}(f) + \eta^\ell_{\mathbf{k}}}{v_{\mathbf{k}\cdot\mathbf{z} \bmod M}}\right) \\ &= k_\ell + \frac{N}{2\pi}\arg\left(\frac{c_{\mathbf{k}}(f) + \eta^\ell_{\mathbf{k}}}{v_{\mathbf{k}\cdot\mathbf{z} \bmod M}}\right) \\ &= k_\ell + \frac{N}{2\pi}\arg\left(1 + \frac{\eta^\ell_{\mathbf{k}} - \eta_{\mathbf{k}}}{v_{\mathbf{k}\cdot\mathbf{z} \bmod M}}\right). \end{aligned}$$

We now only consider $|c_{\mathbf{k}}| > \beta \geq \max(\tau, 3\eta_\infty)$, that is $\mathbf{k} \in \mathscr{S}_\beta \subset \mathscr{S}_\tau$, and therefore, the corresponding approximate coefficient satisfies $|v_{\mathbf{k}\cdot\mathbf{z} \bmod M}| > \beta - \eta_\infty$. Thus, the magnitude of the fraction in the argument must be strictly less than $\frac{2\eta_\infty}{\beta - \eta_\infty} \leq 1$. Therefore, we consider the argument of a point lying in the right half of the complex plane, in the open disc of radius $\frac{2\eta_\infty}{\beta - \eta_\infty}$ centered at 1. The maximal absolute argument of a point in this disc will be that of a point lying on a tangent line passing through the origin. This point, the origin, and 1 then form a right triangle from which we deduce that

$$\left|\arg\left(1+\frac{\eta_{\mathbf{k}}^{\ell}-\eta_{\mathbf{k}}}{v_{\mathbf{k}\cdot z \bmod M}}\right)\right| < \arcsin\left(\frac{2\eta_\infty}{\beta-\eta_\infty}\right),$$

and our choice of $\beta \geq \eta_\infty(1+2/\sin(\pi/N))$ then implies that

$$\left|\arg\left(1+\frac{\eta_{\mathbf{k}}^{\ell}-\eta_{\mathbf{k}}}{v_{\mathbf{k}\cdot z \bmod M}}\right)\right| < \frac{\pi}{N}.$$

Thus,

$$\left|\frac{N}{2\pi}\arg\left(\frac{v_{\mathbf{k}\cdot\mathbf{z} \bmod M}^{\ell}}{v_{\mathbf{k}\cdot\mathbf{z} \bmod M}}\right)-k_\ell\right| < \frac{1}{2},$$

and so after rounding to the nearest integer, Algorithm 1 will recover k_ℓ for all $\ell \in [d]$ and $\mathbf{k} \in \mathscr{S}_\beta$.

We now know that the final loop of Algorithm 1 will properly map the one-dimensional frequency $\omega = \mathbf{k}\cdot\mathbf{z} \bmod M$ to $\mathbf{k}$ for all $\mathbf{k} \in \mathscr{S}_\beta$. Thus, for these same $\mathbf{k} \in \mathscr{S}_\beta$, Line 11 ensures that we set $b_{\mathbf{k}} := v_{\mathbf{k}\cdot\mathbf{z} \bmod M}$. Additionally, the $\max(s-|\mathscr{S}_\beta|, 0)$ many coefficients v_ω for which $\omega \neq \mathbf{k}\cdot\mathbf{z} \bmod M$ for any $\mathbf{k} \in \mathscr{S}_\beta$ are still available for potential assignment. If any multivariate frequency $\mathbf{k}_\omega \in \mathscr{I}$ is reconstructed and passes the mandatory check in Line 10 then the approximate Fourier coefficient v_ω properly corresponds to $(\mathbf{F}_M\,\mathbf{a})_{\mathbf{k}_\omega\cdot\mathbf{z} \bmod M} = c_{\mathbf{k}_\omega}(f)$.

On the other hand, if some error introduced in the SFTs reconstructs a multivariate frequency $\mathbf{k}_\omega \notin \mathscr{I}$, the reconstructing property does not allow us to conclude anything about a (k_ω, ω) pair passing the check in Line 10. Thus, it is possible that v_ω will contribute to some component of $\mathbf{b}$ not corresponding to any frequency in $\mathscr{I}$. At the least however, since we know that all entries of $\mathbf{v}$ corresponding to frequencies in $\mathscr{S}_\beta$ are correctly assigned, the remaining ones satisfy $|v_\omega| \leq \beta + \eta_\infty$. Using these facts allows us to estimate the error as

$$\begin{aligned}\|\mathbf{b}-c\|_{\ell^2(\mathbb{Z}^d)} &\leq \|\mathbf{b}|_{\mathbb{Z}^d\setminus\mathscr{I}}\|_{\ell^2(\mathbb{Z}^d)} + \|\mathbf{b}|_{\mathscr{I}} - c|_{\mathrm{supp}(\mathbf{b})\cap\mathscr{I}}\|_{\ell^2(\mathbb{Z}^d)} + \|c - c|_{\mathscr{S}_\beta}\|_{\ell^2(\mathbb{Z}^d)} \\ &\leq (\beta+\eta_\infty)\sqrt{\max(s-|\mathscr{S}_\beta|, 0)} + \eta_2 + \|c - c|_{\mathscr{S}_\beta}\|_{\ell^2(\mathscr{I})} \qquad (4)\end{aligned}$$

where we have additionally used the accuracy of the initial one-dimensional SFT and the assumption that c is supported on $\mathscr{I}$.

We now handle the case when f is not necessarily a polynomial with Fourier support contained in $\mathscr{I}$. Rather than aiming to approximate $c_{\mathbf{k}}(f)$ for every $\mathbf{k} \in \mathbb{Z}^d$, we restrict attention to only frequencies in $\mathscr{I}$, instead attempting to approximate the Fourier coefficients of $f_{\mathscr{I}} = \sum_{\mathbf{k}\in\mathscr{I}} c_{\mathbf{k}}(f)\mathrm{e}^{2\pi\mathrm{i}\mathbf{k}\cdot\circ}$. We then have that $f =: f_{\mathscr{I}} + f_{\mathbb{Z}^d\setminus\mathscr{I}}$ and view potentially noisy input $f + \mu$ to our algorithm as

$$f + \mu = f_{\mathscr{I}} + \underbrace{f_{\mathbb{Z}^d \setminus \mathscr{I}} + \mu}_{\mu'}.$$

Algorithm 1 applied to $f + \mu$ is then equivalent to applying it to $f_{\mathscr{I}} + \mu'$, where now τ, η_∞, and η_2 depend on μ', and the output is an approximation of $c|_{\mathscr{I}}$. Since μ' represents noise on the input to $\mathscr{A}_{s,M}$ in its applications to $f_{\mathscr{I}}(t\mathbf{z})$ and $S_{\ell,1/N} f_{\mathscr{I}}(t\mathbf{z})$ we remark here that

$$\|\mu'\|_\infty \leq \|f_{\mathbb{Z}^d \setminus \mathscr{I}}\|_\infty + \|\mu\|_\infty \leq \|c(f) - c(f)|_{\mathscr{I}}\|_{\ell^1(\mathbb{Z}^d)} + \|\mu\|_\infty \tag{5}$$

so as to help us estimate τ, η_∞, and η_2 in future applications of the lemma. Accounting for the truncation to $\mathscr{I}$ in the ℓ^2 error bound and using (4) applied to $c|_{\mathscr{I}}$, we estimate

$$\begin{aligned}\|\mathbf{b} - c\|_{\ell^2(\mathbb{Z}^d)} &\leq \|\mathbf{b} - c|_{\mathscr{I}}\|_{\ell^2(\mathbb{Z}^d)} + \|c - c|_{\mathscr{I}}\|_{\ell^2(\mathbb{Z}^d)} \\ &\leq (\beta + \eta_\infty)\sqrt{\max(s - |\mathscr{S}_\beta|, 0)} + \eta_2 + \|c|_{\mathscr{I}} - c|_{\mathscr{S}_\beta}\|_{\ell^2(\mathbb{Z}^d)} \\ &\quad + \|c - c|_{\mathscr{I}}\|_{\ell^2(\mathbb{Z}^d)}.\end{aligned}$$

Recalling that $P(s, M)$ and $R(s, M)$ are the sampling and runtime complexity of $\mathscr{A}_{s,M}$ respectively, since $1 + d$ SFTs are required, the number of f evaluations is $\mathscr{O}(d \cdot P(s, M))$ and the associated computational complexity is $\mathscr{O}(d \cdot R(s, M))$. The complexity of Lines 6–13 is $\mathscr{O}(sd)$. □

Remark 2 Since the only possible misassigned values of v_ω contribute to coefficients in $\mathbf{b}$ outside the chosen frequency set $\mathscr{I}$ for which $\Lambda(\mathbf{z}, M)$ is reconstructing, if it is possible to quickly (e.g., in $\mathscr{O}(d)$ time) check a multivariate frequency's inclusion in $\mathscr{I}$ (e.g., a hyperbolic cross), entries outside of $\mathscr{I}$ in $\mathbf{b}$ can be identified in the optional check on Line 10 and remain (correctly) unassigned. This has the effect of removing the $(\beta + \eta_\infty)\sqrt{\max(s - |\mathscr{S}_\beta|, 0)}$ term in the error bound while not increasing the computational complexity. Additionally, this outputs an approximation to $(c|_{\mathscr{I}})_s^{\text{opt}}$ which is supported only on our supplied frequency set $\mathscr{I}$ as we may expect or prefer.

We now apply Lemma 1 with the discrete sublinear-time SFT from Theorem 2 to give specific error bounds in terms of best s-term approximation errors as well as detailed runtime and sampling complexities.

Corollary 1 (Algorithm 1 with discrete sublinear-time SFT) *Let $N \geq 9$. For $\mathscr{I} \subset \mathscr{B}_N^d$ with reconstructing rank-1 lattice $\Lambda(\mathbf{z}, M)$ and the function $f \in \mathscr{W}(\mathbb{T}^d) \cap C(\mathbb{T}^d)$, we consider applying Algorithm 1 where each function sample may be corrupted by noise at most $e_\infty \geq 0$ in absolute magnitude. Using the discrete sublinear-time SFT algorithm $\mathscr{A}_{2s,M}^{\text{disc}}$ or $\mathscr{A}_{2s,M}^{\text{disc,MC}}$ with parameter $1 \leq r \leq \frac{M}{36}$, Algorithm 1 will produce*

$\mathbf{b} = (b_{\mathbf{k}})_{\mathbf{k}\in\mathscr{B}_N^d}$ *a* $2s$*-sparse approximation of* c *satisfying the error estimate*

$$\|\mathbf{b} - c\|_2 \leq \|c - c|_{\mathscr{I}}\|_2 + (48 + 4N)\,\frac{\|c|_{\mathscr{I}} - (c|_{\mathscr{I}})_s^{\mathrm{opt}}\|_1}{\sqrt{s}} + (188 + 16N)\,\sqrt{s}(\|f\|_\infty M^{-r} + \|c - c|_{\mathscr{I}}\|_1 + e_\infty)$$

albeit with probability $1 - \sigma \in [0, 1)$ *for the Monte Carlo version. The total number of evaluations of* f *and computational complexity will be*

$$\mathscr{O}\left(\frac{ds^2r^{3/2}\log^{11/2} M}{\log s}\right) \text{ or } \mathscr{O}\left(dsr^{3/2}\log^{9/2} M \log\left(\frac{dM}{\sigma}\right)\right)$$

for $\mathscr{A}_{2s,M}^{\mathrm{disc}}$ *or* $\mathscr{A}_{2s,M}^{\mathrm{disc,MC}}$ *respectively.*

Proof For the definitions of τ and β in Lemma 1 with associated values given by Theorem 2, Lemma 4 applied with $\mathbf{x} = c|_{\mathscr{I}}$ implies that $\mathscr{S}_\beta$ can contain at most $2s$ elements and the bound

$$\begin{aligned}\|c|_{\mathscr{I}} - c|_{\mathscr{S}_\beta}\|_{\ell^2(\mathbb{Z}^d)} &\leq \|c|_{\mathscr{I}} - (c|_{\mathscr{I}})_{2s}^{\mathrm{opt}}\|_{\ell^2(\mathbb{Z}^d)} + \beta\sqrt{2s}\\ &\leq \frac{\|c|_{\mathscr{I}} - (c|_{\mathscr{I}})_s^{\mathrm{opt}}\|_{\ell^1(\mathbb{Z}^d)}}{2\sqrt{s}} + \beta\sqrt{2s}.\end{aligned} \tag{6}$$

Note that the last inequality follows from [9, Theorem 2.5] applied to $c|_{\mathscr{I}} - (c|_{\mathscr{I}})_s^{\mathrm{opt}}$. Lemma 1 then holds with s replaced by $2s$ for the $2s$-sparse approximations given by $\mathscr{A}_{2s,M}^{\mathrm{disc}}$ or $\mathscr{A}_{2s,M}^{\mathrm{disc,MC}}$ in Algorithm 1.

After treating the truncation error as measurement noise as well as accounting for any noise in the input bounded by e_∞, Theorem 2 gives the values

$$\eta_\infty = 3\sqrt{2}\left(\frac{\|c|_{\mathscr{I}} - (c|_{\mathscr{I}})_s^{\mathrm{opt}}\|_1}{2s} + 2(\|f\|_\infty M^{-r} + \|c - c|_{\mathscr{I}}\|_1 + e_\infty)\right),$$
$$\tau = \frac{12(1+\sqrt{2})}{3\sqrt{2}}\eta_\infty.$$

Assuming $N \geq 9$,

$$\beta = \max\left(\tau, \eta_\infty\left(1 + \frac{2}{\sin\left(\frac{\pi}{N}\right)}\right)\right) = \eta_\infty\left(1 + \frac{2}{\sin\left(\frac{\pi}{N}\right)}\right) \leq \eta_\infty\left(1 + \frac{2}{9\sin\left(\frac{\pi}{9}\right)}N\right).$$

Inserting the estimate for $\|c|_{\mathscr{I}} - c|_{\mathscr{S}_\beta}\|_2$ from (6), this bound for β, and the value for η_2 from Theorem 2 (where again we use [9, Theorem 2.5])

$$\eta_2 \leq \frac{77\|c|_{\mathscr{I}} - (c|_{\mathscr{I}})_s^{\mathrm{opt}}\|_1}{2\sqrt{s}} + 152\sqrt{s}(\|f\|_\infty M^{-r} + \|c - c|_{\mathscr{I}}\|_1 + e_\infty)$$

into the recovery bound in Lemma 1 gives the final error estimate.

The change to the complexity of the randomized algorithm arises from distributing the probability of failure σ over the $d+1$ SFTs in a union bound. □

Though the nonequispaced SFTs discussed in Theorem 1 do not approximate the discrete Fourier transform and therefore do not alias the one-dimensional frequencies $\mathbf{k}\cdot\mathbf{z}$ into frequencies in $\mathscr{B}_M$, slightly modifying Algorithm 1 to use SFTs with a larger bandwidth allows for the following recovery result.

Corollary 2 (Algorithm 1 with nonequispaced sublinear-time SFT) *For $\mathscr{I} \subset \mathscr{B}_N^d$ with $N \geq 6$, fix the new bandwidth parameter $\tilde{M} := 2\max_{\mathbf{k}\in\mathscr{I}} |\mathbf{k}\cdot\mathbf{z}| + 1$. For $\Lambda(\mathbf{z}, M)$, a reconstructing rank-1 lattice for $\mathscr{I}$ with $M \leq \tilde{M}$, and the function $f \in \mathscr{W}(\mathbb{T}^d) \cap C(\mathbb{T}^d)$, we consider applying Algorithm 1 where each function sample may be corrupted by noise at most $e_\infty \geq 0$ in absolute magnitude with the following modifications:*

1. *use the sublinear-time SFT algorithm $\mathscr{A}^{\mathrm{sub}}_{2s,\tilde{M}}$ or $\mathscr{A}^{\mathrm{sub,MC}}_{2s,\tilde{M}}$*
2. *and only check equality against ω in Line 10 (rather than equivalence modulo M),*

to produce $\mathbf{b} = (b_{\mathbf{k}})_{\mathbf{k}\in\mathscr{B}_N^d}$ a $2s$-sparse approximation of c satisfying the error estimate

$$\|\mathbf{b} - c\|_{\ell^2(\mathbb{Z}^d)} \leq (24+3N)\left[\frac{\|c|_{\mathscr{I}} - (c|_{\mathscr{I}})_s^{\mathrm{opt}}\|_1}{\sqrt{s}} + \sqrt{s}\|c - c|_{\mathscr{I}}\|_1 + \sqrt{s}e_\infty\right] + \|c - c|_{\mathscr{I}}\|_2.$$

albeit with probability $1-\sigma \in [0,1)$ for the Monte Carlo version. For $\mathscr{A}^{\mathrm{sub}}_{2s,\tilde{M}}$ and $\mathscr{A}^{\mathrm{sub,MC}}_{2s,\tilde{M}}$ respectively, the total number of evaluations of f and computational complexity will be

$$\mathscr{O}\left(\frac{ds^2\log^4\tilde{M}}{\log s}\right) \text{ or } \mathscr{O}\left(ds\log^3(\tilde{M})\log\left(\frac{d\tilde{M}}{\sigma}\right)\right).$$

Proof The bandwidth specified ensures that $\mathscr{B}_{\tilde{M}} \supset \{\mathbf{k}\cdot\mathbf{z} \mid \mathbf{k}\in\mathscr{I}\}$. In the case where f is a trigonometric polynomial with $\operatorname{supp}(c(f)) \subset \mathscr{I}$, so long as there exists some $M \leq \tilde{M}$ such that $\Lambda(\mathbf{z}, M)$ is reconstructing for $\mathscr{I}$, we are guaranteed that a length-$\tilde{M}$ DFT on a polynomial supported on $\{\mathbf{k}\cdot\mathbf{z} \mid \mathbf{k}\in\mathscr{I}\}$ will not suffer from aliasing collisions. Thus, the one-dimensional Fourier transforms truncated to $\mathscr{B}_{\tilde{M}}$ coincide with length $\tilde{M}$ DFTs. Thus, we can view an approximation from the algorithm in Theorem 1 as one of a length $\tilde{M}$ DFT. The reasoning in the proofs of Lemma 1 and Corollary 1 then holds with the SFT algorithms, parameters, numbers of samples, and complexities of Theorem 1. □

Remark 3 As in [13], we can estimate $\tilde{M}$ above with two different techniques:

$$\tilde{M} = 1 + 2 \max_{\mathbf{k} \in \mathscr{I}} \left| \sum_{\ell \in [d]} k_\ell z_\ell \right| \leq 1 + 2 \sum_{\ell \in [d]} |z_\ell| \max_{\mathbf{k} \in \mathscr{I}} |k_\ell| = \mathscr{O}(dNM),$$

$$\tilde{M} = 1 + 2 \max_{\mathbf{k} \in \mathscr{I}} \left| \sum_{\ell \in [d]} k_\ell z_\ell \right| \leq 1 + 2\|\mathbf{z}\|_\infty \max_{\mathbf{k} \in \mathscr{I}} \|\mathbf{k}\|_1 = \mathscr{O}\left(M \max_{\mathbf{k} \in \mathscr{I}} \|\mathbf{k}\|_1 \right).$$

The latter case is especially useful when $\mathscr{I}$ is a subset of a known ℓ^1 ball as it will provide a dimension independent upper bound on $\tilde{M}$. Either of these upper bounds may then be used in practice to avoid having to estimate $\tilde{M}$.

That being said however, if one is willing to perform the one-time search through the frequency set $\mathscr{I}$ to more accurately calculate $\tilde{M}$, one can go even further to use the minimal bandwidth $\tilde{M}' = \max_{\mathbf{k} \in \mathscr{I}}(\mathbf{k} \cdot \mathbf{z}) - \min_{\mathbf{k} \in \mathscr{I}}(\mathbf{k} \cdot \mathbf{z}) + 1$ so long as the function samples are properly modulated to shift the one-dimensional frequencies into $\mathscr{B}_{\tilde{M}'}$. For example, running $\mathscr{A}^{\text{sub}}_{2s,\tilde{M}'}$ or $\mathscr{A}^{\text{sub,MC}}_{2s,\tilde{M}'}$ on $a(t) = \mathbb{e}^{2\pi \mathbb{i} \phi t} f(t\mathbf{z})$ and $a^\ell(t) = \mathbb{e}^{2\pi \mathbb{i} \phi t} S_{\ell,1/N} f(t\mathbf{z})$ with $\phi = \left\lfloor \frac{\tilde{M}'}{2} \right\rfloor - \max_{\mathbf{k} \in \mathscr{I}}(\mathbf{k} \cdot \mathbf{z})$ is acceptable so long as this shift is accounted for in the frequency check on Line 10. Of course, these improvements will only have the effect of reducing the logarithmic factors in the computational complexity.

4.2 Two-dimensional DFT technique

Below, we will consider a method for recovering frequencies which, rather than shifting one dimension of the multivariate periodic function f at a time, leaves one dimension of f out at a time. We will fix one dimension $\ell \in [d]$ of f at equispaced nodes over $\mathbb{T}$ and apply a lattice SFT to the other $d - 1$ components. Applying a standard FFT to the results will produce a two-dimensional DFT. The indices corresponding to the standard FFT will represent frequency components in dimension ℓ while the indices corresponding to the lattice SFT will be used to synchronize with known one-dimensional frequencies $\mathbf{k} \cdot \mathbf{z} \bmod M$.

Again, before stating Algorithm 2 in detail, we present an example.

Example 2 (Two-dimensional DFT technique on a trigonometric monomial)

As in Example 1, we let f be the trigonometric monomial $f(\mathbf{x}) := \mathbb{e}^{2\pi \mathbb{i} \mathbf{k} \cdot \mathbf{x}}$. However, in this example, we let $d = 3$, so $\mathbf{k} \in \mathscr{I} \subset \mathbb{Z}^3$ and the domain of f is $\mathbb{T}^3$ depicted in Fig. 2. We will consider the procedure to compute the $\ell = 1$ component of $\mathbf{k}$.

First, we take a reconstructing rank-1 lattice $\Lambda(\mathbf{z}, M)$ for $\mathscr{I}$ and restrict all but the first component of f to the lattice. This produces a two-dimensional function of the form

$$(x_1, t) \mapsto \mathbb{e}^{2\pi \mathbb{i}(k_1 x_1 + k_2 z_2 t + k_3 z_3 t)}.$$

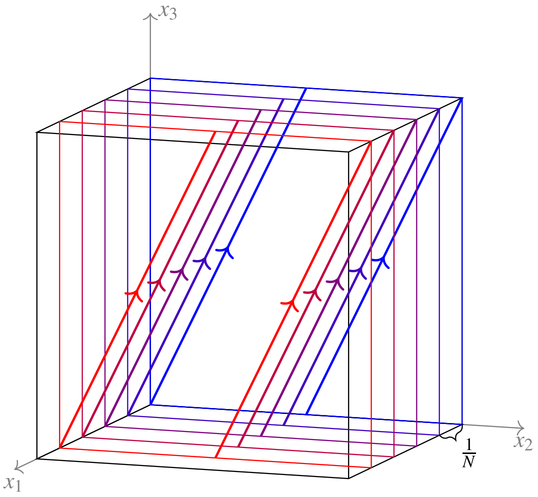


Fig. 2 An example of $\mathbb{T}^3$ depicting the N projected rank-1 lattices on which $f(\mathbf{x})$ is sampled to compute the $\ell = 1$ component of each d-dimensional frequency

We then sample this function at N equispaced points over $\mathbb{T}$ in the x_1 variable. This produces N projected lattices spaced $1/N$ apart in the x_1 direction on which we sample f, depicted in Fig. 2. Fixing x_1 at each equispaced point produces the N univariate functions which are organized into the top array of Fig. 3. Notice that colors of the entries in this array correspond to the lattices in Fig. 2 over which we sample f to produce that entry.

The next step is to apply an SFT to each of the univariate functions in this array. Each function has exactly one active frequency, $k_2 z_2 + k_3 z_3$, with corresponding Fourier coefficient $\mathrm{e}^{2\pi \mathrm{i} k_1 j/N}$. Thus, collecting the results into a matrix, produces the left-most matrix in Fig. 3 with only the $k_2 z_2 + k_3 z_3 \bmod M$ column filled. This column contains N equispaced samples of the function $\mathrm{e}^{2\pi \mathrm{i} k_1 x_1}$, and so finally applying a DFT to the matrix will produce the right-most matrix in Fig. 3. We find only one entry in row $k_1 \bmod M$ corresponding to the only active frequency of $\mathrm{e}^{2\pi \mathrm{i} k_1 x_1}$. Thus, we can read off the $\ell = 1$ entry of $\mathbf{k}$ by determining which row contains the Fourier coefficient of f of interest. Repeating this process for all $\ell = 1, \dots, d$ we will be able to recover $\mathbf{k}$.

We now generalize the procedure demonstrated in Example 2 in a lemma. In particular, we must account for functions which have more than one significant frequency. For theoretical simplicity, we use a length M-DFT in the first step rather than an SFT.

Lemma 2 *Fix some finite multivariate frequency set* $\mathscr{I} \subset \mathscr{B}_N^d$, *let* $\Lambda(\mathbf{z}, M)$ *be a reconstructing rank-1 lattice for* $\{\mathbf{k} - k_\ell \mathbf{e}_\ell \mid \mathbf{k} \in \mathscr{I}\}$ *for all* $\ell \in [d]$, *and assume that* f *has Fourier support* $\operatorname{supp}(c) \subset \mathscr{I}$. *Fixing one dimension* $\ell \in [d]$, *and writing the generating vector as* $\mathbf{z} = (z_\ell, \mathbf{z}'_\ell) \in \mathbb{Z}^d$, *define the polynomials*

$$a_j^\ell(t) := f\left(\frac{j}{N}, t\mathbf{z}'_\ell\right) \textit{ for all } j \in [N],$$

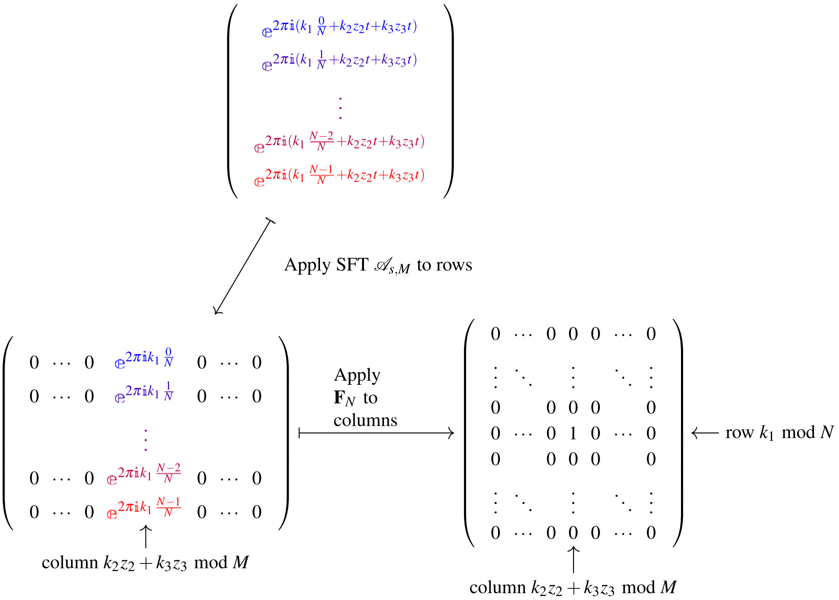

Apply SFT $\mathscr{A}_{s,M}$ to rows
Apply
$\mathbf{F}_N$ to
columns
row k_1 mod N
column $k_2z_2 + k_3z_3$ mod M
column $k_2z_2 + k_3z_3$ mod M

Fig. 3 One round of the basic procedure for the two dimensional DFT algorithm applied to the trigonometric monomial $f(\mathbf{x}) = \mathbb{e}^{2\pi \mathbb{i} \mathbf{k}\cdot\mathbf{x}}$ sampled over the sets depicted in Fig. 2. Notice that each row corresponds to samples of $f(\mathbf{x})$ on the shifted lattice of the corresponding color

that is, fix coordinate ℓ at j/N and restrict the remaining coordinates to dimensions $[d] \setminus \{\ell\}$ of the rank-1 lattice. Then for all one-dimensional frequencies $\omega \in [M]$,

$$\left(\mathbf{F}_M\, \mathbf{a}_j^\ell\right)_\omega = \begin{cases} \sum\limits_{\substack{h_\ell \in \mathscr{B}_N \text{s.t.} \\ (h_\ell, \mathbf{k}'_\ell) \in \mathscr{I}}} \mathbb{e}^{2\pi \mathbb{i} j h_\ell / N}\, c_{(h_\ell, \mathbf{k}'_\ell)}(f) & \text{if } \exists \mathbf{k} \in \mathscr{I} \text{ with } \omega \equiv \mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \pmod{M}, \\ 0 & \text{otherwise.} \end{cases}$$

Moreover, defining the matrix $\mathbf{A}^\ell = \left(\left(\mathbf{F}_M\, \mathbf{a}_j^\ell\right)_\omega\right)_{j\in[N],\omega\in[M]}$, we have

$$\left(\mathbf{F}_N\, \mathbf{A}^\ell\right)_{k_\ell \bmod N, \mathbf{k}'_\ell\cdot\mathbf{z}'_\ell \bmod M} = c_{\mathbf{k}}(f) \textit{ for all } \mathbf{k} \in \mathscr{I},$$

and the remaining entries of the matrix $\mathbf{F}_N\, \mathbf{A}^\ell \in \mathbb{C}^{N\times M}$ are zero.

Proof Using the Fourier series representation of f, we have

$$a_j^\ell(t) := \sum_{\mathbf{k}\in\mathscr{I}} c_{\mathbf{k}}(f)\, \mathbb{e}^{2\pi \mathbb{i}\left(\frac{jk_\ell}{N} + \mathbf{k}'_\ell\cdot\mathbf{z}'_\ell t\right)}.$$

We calculate for $\omega \in [M]$

$$\begin{aligned}\left(\mathbf{F}_M\,\mathbf{a}_j^\ell\right)_\omega &= \frac{1}{M}\sum_{i\in[M]}\sum_{\mathbf{h}\in\mathscr{I}} \mathbb{e}^{\frac{2\pi\mathbb{i}\,jh_\ell}{N}}\, c_{\mathbf{h}}(f)\, \mathbb{e}^{\frac{2\pi\mathbb{i}(\mathbf{h}'_\ell\cdot\mathbf{z}'_\ell-\omega)i}{M}}\\ &= \sum_{\mathbf{h}\in\mathscr{I}} \mathbb{e}^{\frac{2\pi\mathbb{i}\,jh_\ell}{N}}\, c_{\mathbf{h}}(f)\,\delta_{0,(\mathbf{h}'_\ell\cdot\mathbf{z}'_\ell-\omega \bmod M)}.\end{aligned}$$

When there exists some $\mathbf{k} \in \mathscr{I}$ such that $\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \equiv \omega \pmod{M}$, the fact that $\Lambda(\mathbf{z}, M)$ is a reconstructing rank-1 lattice for $\{\mathbf{k} - k_\ell\mathbf{e}_\ell \mid \mathbf{k} \in \mathscr{I}\}$ ensures that such $\mathbf{k}'_\ell$ satisfying this equivalence is unique. Then, we can simplify this sum to

$$\left(\mathbf{F}_M\,\mathbf{a}_j^\ell\right)_\omega = \sum_{\substack{h_\ell\in\mathscr{B}_N \text{ s.t.}\\ (h_\ell,\mathbf{k}'_\ell)\in\mathscr{I}}} \mathbb{e}^{\frac{2\pi\mathbb{i}\,jh_\ell}{N}}\, c_{(h_\ell,\mathbf{k}'_\ell)}(f).$$

When no $\mathbf{k} \in \mathscr{I}$ exists such that $\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \equiv \omega \pmod{M}$, this sum is instead zero as desired. Applying $\mathbf{F}_N$ to $\mathbf{A}^\ell$ then allows us to compute

$$\left(\mathbf{F}_N\,\mathbf{A}^\ell\right)_{k_\ell \bmod N,\, \mathbf{k}'_\ell\cdot\mathbf{z}'_\ell \bmod M} = \frac{1}{N}\sum_{j\in[N]}\sum_{h_\ell\in\mathscr{B}_N} c_{(h_\ell,\mathbf{k}'_\ell)}(f)\,\mathbb{e}^{\frac{2\pi\mathbb{i}(h_\ell-k_\ell \bmod N)j}{N}} = c_{\mathbf{k}}(f).$$

□

Example 2 and Lemma 2 explain the procedure in Lines 1 through 10 of Algorithm 2. However, some care must be taken when we assign rows of nonzero entries in the resulting matrix to coordinates of significant frequencies. The solution is the minimization problem in Line 14. This step uses column information as well as the values of the entries in the matrix to ensure that we are properly matching frequency components with the correct Fourier coefficient v_ω.

The remainder of the algorithm is the same as Algorithm 1. Line 16 consists of the same check to ensure that recovered frequencies are correct, and if this check passes, the one-dimensional Fourier coefficient is assigned to its matched d-dimensional frequency.

Remark 4 We bring special attention to the fact that Algorithm 2 requires as input a rank-1 lattice $\Lambda(\mathbf{z}, M)$ which is reconstructing for not only $\mathscr{I}$, but also the projections of $\mathscr{I}$ of the form $\{\mathbf{k} - k_\ell\mathbf{e}_\ell \mid \mathbf{k} \in \mathscr{I}\}$ for any $\ell \in [d]$. For frequency sets $\mathscr{I}$ which are downward closed, that is, if $\mathscr{I}$ is such that for any $\mathbf{k} \in \mathscr{I}$ and $\mathbf{h} \in \mathbb{Z}^d$, $|\mathbf{h}| \leq |\mathbf{k}|$ component-wise implies that $\mathbf{h} \in \mathscr{I}$, any reconstructing rank-1 lattice for $\mathscr{I}$ is necessarily one for the considered projections as well. Thus, for many frequency spaces of interest, e.g., hyperbolic crosses (cf. Remarks 2 and 3 as well as Sect. 5 below), any reconstructing rank-1 lattice for $\mathscr{I}$ will suffice as input to Algorithm 2.

Algorithm 2 Frequency Index Recovery by Two Dimensional DFT

Input: A multivariate periodic function $f \in \mathscr{W}(\mathbb{T}^d) \cap C(\mathbb{T}^d)$ (from which we are able to obtain potentially noisy samples), a multivariate frequency set $\mathscr{I} \subset \mathscr{B}_N^d$, a rank-1 lattice $\Lambda(\mathbf{z}, M)$ which is reconstructing for $\mathscr{I}$ and $\{\mathbf{k} - k_\ell \mathbf{e}_\ell \mid \mathbf{k} \in \mathscr{I}\}$ for all $\ell \in [d]$, and an SFT algorithm $\mathscr{A}_{s,M}$.

Output: Sparse coefficient vector $\mathbf{b} = (b_\mathbf{k})_{\mathbf{k} \in \mathscr{B}_N^d}$ (optionally supported on $\mathscr{I}$, see Line 16), an approximation to $\left(c|_\mathscr{I}\right)_s^{\text{opt}}$.

```
1: Apply $\mathscr{A}_{s,M}$ to the univariate restriction of $f$ to the lattice, $a(t) := f(t\mathbf{z})$, to produce $\mathbf{v} := \mathscr{A}_{s,M} a$, a sparse approximation of $\mathbf{F}_M\, \mathbf{a} \in \mathbb{C}^M$.
2: for all $\ell \in [d]$ do
3:     for all $j \in [N]$ do
4:         Apply $\mathscr{A}_{s,M}$ to $a_j^\ell(t) := f(\frac{j}{N}, t\mathbf{z}'_\ell)$ to produce $\mathbf{v}_j^\ell := \mathscr{A}_{s,M} a_j^\ell$, a sparse approximation of $\mathbf{F}_M\, \mathbf{a}_j^\ell$.
5:         Row $j$ of $\mathbf{V}^\ell \leftarrow \mathbf{v}_j^\ell$.
6:     end for
7:     for all nonzero columns $\omega$ of $\mathbf{V}^\ell$ do
8:         Apply $\mathbf{F}_N$ to column $\omega$ of $\mathbf{V}^\ell$ to produce $\mathbf{F}_N\, \mathbf{V}^\ell$.
9:     end for
10: end for
11: $\mathbf{b} \leftarrow \mathbf{0}$
12: for all $\omega \in \operatorname{supp}(\mathbf{v})$ do
13:     for all $\ell \in [d]$ do
14:         $((k_\omega)_\ell, \sim) \leftarrow \arg\min\{|v_\omega - (\mathbf{F}_N\, \mathbf{V}^\ell)_{h,\omega'}| \mid (h, \omega') \in \mathscr{B}_N \times [M]$ with $hz_\ell + \omega' \equiv \omega \pmod{M}\}$
15:     end for
16:     if $\mathbf{k}_\omega \cdot \mathbf{z} \equiv \omega \pmod{M}$ (and optionally $\mathbf{k}_\omega \in \mathscr{I}$) then
17:         $b_{\mathbf{k}_\omega} \leftarrow b_{\mathbf{k}_\omega} + v_\omega$
18:     end if
19: end for
```

4.2.1 Analysis of Algorithm 2

With the conceptual explanation of Algorithm 2 complete, we now provide error guarantees for its output.

Lemma 3 (General recovery result for Algorithm 2) *Let f, $\mathscr{I}$, and $\Lambda(\mathbf{z}, M)$ be as specified in the input to Algorithm 2. Additionally, let $\mathscr{A}_{s,M}$ be a noise-robust SFT algorithm satisfying the same constraints as in Lemma 1 with parameters τ and η_∞ holding uniformly for each SFT performed in Algorithm 2.*

Collect the τ-significant frequencies of f into the set $\mathscr{S}_\tau := \{\mathbf{k} \in \mathscr{I} \mid |c_\mathbf{k}(f)| > \tau\}$ and assume that $|\mathscr{S}_\tau| \leq s$. Then Algorithm 2 (ignoring the optional check on Line 16) will produce an s-sparse approximation of the Fourier coefficients of f satisfying the error estimate

$$\|\mathbf{b} - c\|_{\ell^2(\mathbb{Z}^d)} \leq \eta_2 + (4\tau + \eta_\infty)\sqrt{\max(s - |\mathscr{S}_{4\tau}|, 0)} \\ + \|c|_\mathscr{I} - c|_{\mathscr{S}_{4\tau}}\|_{\ell^2(\mathbb{Z}^d)} + \|c - c|_\mathscr{I}\|_{\ell^2(\mathbb{Z}^d)},$$

requiring $\mathscr{O}(dN \cdot P(s, M))$ total evaluations of f, in $\mathscr{O}(dN(R(s, M) + sN \log N))$ total operations.

Proof We begin by assuming that f is a trigonometric polynomial with $\operatorname{supp}(c(f)) \subset \mathscr{I}$. Since $\Lambda(\mathbf{z}, M)$ is a reconstructing rank-1 lattice for $\mathscr{I}$, the DFT-aliasing ensures that Line 1 of Algorithm 2 will return approximate coefficients uniquely corresponding to all τ-significant frequencies $\mathbf{k} \in \mathscr{S}_\tau$ which we can label $v_{\mathbf{k}\cdot\mathbf{z} \bmod M}$. Additionally, Line 4 recovers approximations to all τ-significant frequencies of $\mathbf{F}_M\, \mathbf{a}_j^\ell$ which have the form given in Lemma 2. In particular, if $\mathbf{k} \in \mathscr{S}_\tau$, we have

$$\begin{aligned}
\tau < |c_{\mathbf{k}}(f)| &= \left| \left(\mathbf{F}_N\, \mathbf{A}^\ell\right)_{k_\ell \bmod N, \mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} \right| \\
&= \left| \frac{1}{N} \sum_{j \in [N]} \left(\mathbf{F}_M\, \mathbf{a}_j^\ell\right)_{\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} \mathrm{e}^{\frac{-2\pi \mathrm{i}\, j k_\ell \bmod N}{N}} \right| \\
&\leq \frac{1}{N} \sum_{j \in [N]} \left| \left(\mathbf{F}_M\, \mathbf{a}_j^\ell\right)_{\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} \right| \\
&\leq \max_{j \in [N]} \left| (\mathbf{F}_M\, \mathbf{a}_j^\ell)_{\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} \right| .
\end{aligned}$$

Thus, there exists at least one $\mathbf{F}_M\, \mathbf{a}_j^\ell$ with $\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M$ recovered as a τ-significant frequency in the SFT of Line 4, and $\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M$ will be a nonzero column in $\mathbf{V}^\ell$ for all $\mathbf{k} \in \mathscr{S}_\tau$.

Analyzing these SFTs in more detail for any $\mathbf{k} \in \mathscr{I}$ such that $\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M$ is a nonzero column of $\mathbf{V}^\ell$, we write

$$\left(\mathbf{v}_j^\ell\right)_{\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} = \left(\mathbf{F}_M\, \mathbf{a}_j^\ell\right)_{\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} + \left(\eta_j^\ell\right)_{\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M}$$

where, by the ℓ^∞ and recovery guarantees for $\mathscr{A}_{s,M}$, the error satisfies

$$\left| \left(\eta_j^\ell\right)_{\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} \right| \leq \begin{cases} \eta_\infty & \text{if } \left(\mathbf{v}_j^\ell\right)_{\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} \neq 0 \\ \tau & \text{if } \left(\mathbf{v}_j^\ell\right)_{\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} = 0 \end{cases} \leq \tau.$$

Thus, in the application of $\mathbf{F}_N$ to column $\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M$ of $\mathbf{V}^\ell$, we have

$$\begin{aligned}
&\left(\mathbf{F}_N\, \mathbf{V}^\ell\right)_{k_\ell \bmod N, \mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} \\
&\quad = \left(\mathbf{F}_N\, \mathbf{A}^\ell\right)_{k_\ell \bmod N, \mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M} + \left(\mathbf{F}_N\, \left(\left(\eta_j^\ell\right)_{\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell \bmod M}\right)_{j \in [N]}\right)_{k_\ell \bmod N} \\
&\quad =: c_{\mathbf{k}}(f) + \eta_{\mathbf{k}}^\ell
\end{aligned}$$

with

$$|\eta^\ell_{\mathbf{k}}| = \left|\frac{1}{N}\sum_{j\in[N]} \left(\eta^\ell_j\right)_{\mathbf{k}'_\ell\cdot\mathbf{z}'_\ell \bmod M} \mathrm{e}^{\frac{-2\pi \mathrm{i}\, jk_\ell \bmod N}{N}}\right| \le \max_{j\in[N]} \left|\left(\eta^\ell_j\right)_{\mathbf{k}'_\ell\cdot\mathbf{z}'_\ell \bmod M}\right| \le \tau.$$

These same calculations apply to the computed columns of $\mathbf{F}_N\,\mathbf{V}^\ell$ which do not correspond to values of $\mathbf{k}'_\ell\cdot\mathbf{z}'_\ell \bmod M$ for some $\mathbf{k}\in\mathscr{I}$ since we assume $\operatorname{supp}(c(f))\subset\mathscr{I}$, and so at worst, these columns are filled with noise bounded in magnitude by τ.

Restricting our attention to $\mathbf{k}\in\mathscr{S}_{4\tau}\subset\mathscr{S}_\tau$, we know that Line 14 will be run with $\omega=\mathbf{k}\cdot\mathbf{z} \bmod M$ and $(k_\ell \bmod N, \mathbf{k}'_\ell\cdot\mathbf{z}'_\ell \bmod M)$ as an admissible index in the minimization. By the reconstructing property of $\Lambda(\mathbf{z}, M)$, no other $\mathbf{h}\in\mathscr{I}$ will correspond to an admissible index $(h_\ell \bmod N, \mathbf{h}'_\ell\cdot\mathbf{z}'_\ell \bmod M)$, and so the only remaining values of $(\mathbf{F}_N\,\mathbf{V}^\ell)_{h,\omega'}$ in the minimization correspond to pure noise η bounded in magnitude by τ. Analyzing the objective at $(k_\ell \bmod N, \mathbf{k}'_\ell\cdot\mathbf{z}'_\ell \bmod M)$, we find

$$|v_{\mathbf{k}\cdot\mathbf{z} \bmod M} - (\mathbf{F}_N\,\mathbf{V}^\ell)_{k_\ell \bmod N, \mathbf{k}'_\ell\cdot\mathbf{z}'_\ell \bmod M}| \le 2\tau < |c_{\mathbf{k}}(f)| - 2\tau \le |v_{\mathbf{k}\cdot\mathbf{z} \bmod M} - \eta|,$$

and so the value for $(k_\omega)_\ell$ will in fact be assigned k_ℓ. Thus, after all d components of $\mathbf{k}_\omega=\mathbf{k}$ have been recovered, $b_{\mathbf{k}}$ will be assigned $v_{\mathbf{k}\cdot\mathbf{z} \bmod M}$.

The remaining $\max(s-|\mathscr{S}_{4\tau}|, 0)$ nonzero entries of $\mathbf{v}$ can be distributed to entries of $\mathbf{b}$ possibly correctly but with no guarantee; at the very least however, these values must be at most $4\tau+\eta_\infty$ in magnitude. We split $\mathbf{b}$ as $\mathbf{b}=\mathbf{b}^{\text{correct}}+\mathbf{b}^{\text{incorrect}}$ to account for the values of $\mathbf{v}$ respectively assigned correctly and incorrectly and note that $\operatorname{supp}(\mathbf{b}^{\text{correct}})\supset\mathscr{S}_{4\tau}$. We then estimate the error as

$$\begin{aligned}\|\mathbf{b}-c\|_{\ell^2(\mathbb{Z}^d)} &\le \|\mathbf{b}^{\text{correct}} - c|_{\operatorname{supp}(\mathbf{b}^{\text{correct}})}\|_{\ell^2(\mathbb{Z}^d)} + \|\mathbf{b}^{\text{incorrect}}\|_{\ell^2(\mathbb{Z}^{\mathrm{d}})}\\ &\quad + \|c - c|_{\operatorname{supp}(\mathbf{b}^{\text{correct}})}\|_{\ell^2(\mathbb{Z}^d)}\\ &\le \eta_2 + (4\tau+\eta_\infty)\sqrt{\max(s-|\mathscr{S}_{4\tau}|, 0)} + \|c - c|_{\mathscr{S}_{4\tau}}\|_{\ell^2(\mathbb{Z}^d)}.\end{aligned}$$

As in the proof of Lemma 1, we note that the mandatory check in Line 16 helps ensure that all misassigned values v_ω which contribute to $\mathbf{b}^{\text{incorrect}}$ correspond to reconstructed $\mathbf{k}_\omega$ *outside* of $\mathscr{I}$, with the optional check in this line (see Remark 2) eliminating $\mathbf{b}^{\text{incorrect}}$ and the corresponding term in the error estimate entirely.

Now, supposing that the Fourier support of f is not limited to only $\mathscr{I}$, just as in the analysis for Algorithm 1, we treat f as a perturbation of $f_{\mathscr{I}}$, and use the robust SFT algorithm and the previous argument to approximate $c|_{\mathscr{I}}$. Note again that in each SFT, the noise added when using measurements of f as proxies for those of $f_{\mathscr{I}}$ is compounded by $\|f_{\mathbb{Z}^d\setminus\mathscr{I}}\|_\infty$ and is bounded by $\|c-c|_{\mathscr{I}}\|_{\ell^1(\mathbb{Z}^d)}$. Applying the guarantees above gives

$$\begin{aligned}\|\mathbf{b}-c\|_{\ell^2(\mathbb{Z}^d)} &\le \|\mathbf{b}-c|_{\mathscr{I}}\|_{\ell^2(\mathbb{Z}^d)} + \|c-c|_{\mathscr{I}}\|_{\ell^2(\mathbb{Z}^d)}\\ &\le \eta_2 + (4\tau+\eta_\infty)\sqrt{\max(s-|\mathscr{S}_{4\tau}|, 0)}\\ &\qquad + \|c|_{\mathscr{I}} - c|_{\mathscr{S}_{4\tau}}\|_{\ell^2(\mathbb{Z}^d)} + \|c-c|_{\mathscr{I}}\|_{\ell^2(\mathbb{Z}^d)}.\end{aligned}$$

Employing fast Fourier transforms for the at most dsN DFTs, the computational complexity of Lines 2–10 is $\mathscr{O}\left(d(N \cdot R(s, M) + sN^2 \log N)\right)$ (which dominates the complexity of the remainder of the algorithm). Since $1 + dN$ SFTs are required, the number of f evaluations is $\mathscr{O}(dN \cdot P(s, M))$. □

Remark 5 Though the number of nonzero columns of $\mathbf{V}^\ell$ can be theoretically at most sN, in practice with a high quality algorithm, each of the N SFTs should recover nearly the same frequencies, meaning that there are actually $\mathscr{O}(s)$ columns. This would remove a power of N in the second term of the runtime estimate.

Note however, that even with near exact SFT algorithms, recovering exactly s total frequencies is not a certainty. There can be cancellations for certain terms in $\mathbf{F}_M\, \mathbf{a}_j^\ell$ depending interactions between the coefficients sharing the same values on their $[d] \setminus \{\ell\}$ entries, which makes it possible that an SFT on $\mathbf{F}_M\, \mathbf{a}_j^\ell$ will miss coefficients. If required to output s-entries, an SFT algorithm could favor some noisy value corresponding to a frequency outside the support.

Remark 6 Though we perform an exact FFT of the nonzero columns of $\mathbf{V}^\ell$ in Line 8 of Algorithm 2, Lemma 2 implies that the resulting matrix will be as sparse as the original function's Fourier transform. Thus, for a truly compressible function, an SFT down the columns of $\mathbf{V}^\ell$ would be feasible as well. However, in especially higher dimensions, even small N can allow for large frequency spaces $\mathscr{I}$. In these large frequency spaces, what is perceived as relatively sparse can therefore quickly surpass N, rendering an s-sparse, length N SFT useless.

As a simple example, consider $\mathscr{I}$ to be the cube of side length $N = s$, $\mathscr{B}_s^d$. For d large enough, any frequency support of size s will be small in comparison to $|\mathscr{I}| = s^d$. However, using an s-sparse SFT instead of a length-s DFT in Algorithm 2 will actually be more expensive.

Applying the discrete sublinear-time SFT from Theorem 2 to Lemma 3 analogously to the derivation of Corollary 1 from Lemma 1 allows for the following recovery bound for Algorithm 2. In particular, we observe asymptotically improved error guarantees over Corollary 1 at the cost of a slight increase in runtime.

Corollary 3 (Algorithm 2 with discrete sublinear-time SFT) *For $\mathscr{I} \subset \mathbb{Z}^d$ with reconstructing rank-1 lattice $\Lambda(\mathbf{z}, M)$ and the function $f \in \mathscr{W}(\mathbb{T}^d) \cap C(\mathbb{T}^d)$, we consider applying Algorithm 2 where each function sample may be corrupted by noise at most $e_\infty \geq 0$ in absolute magnitude. Using the discrete sublinear-time SFT algorithm $\mathscr{A}_{2s,M}^{\mathrm{disc}}$ or $\mathscr{A}_{2s,M}^{\mathrm{disc,MC}}$ with parameter $1 \leq r \leq \frac{M}{36}$ will produce $\mathbf{b} = (b_\mathbf{k})_{\mathbf{k} \in \mathscr{B}_N^d}$ a $2s$-sparse approximation of c satisfying the error estimate*

$$\|\mathbf{b} - c\|_{\ell^2(\mathbb{Z}^d)} \leq 206 \frac{\|c|_{\mathscr{I}} - (c|_{\mathscr{I}})_s^{\mathrm{opt}}\|_1}{\sqrt{s}} + 820\sqrt{s}(\|f\|_\infty M^{-r} + \|c - c|_{\mathscr{I}}\|_1 + e_\infty) + \|c - c|_{\mathscr{I}}\|_2$$

albeit with probability $1 - \sigma \in [0, 1)$ for the Monte Carlo version.

The total number of evaluations of f and the computational complexity will be

$$\mathscr{O}\left(dsN\left(\frac{sr^{3/2}\log^{11/2}M}{\log s}+N\log N\right)\right)$$
$$\textit{or}\quad \mathscr{O}\left(dsN\left(r^{3/2}\log^{9/2}(M)\log\left(\frac{dNM}{\sigma}\right)+N\log N\right)\right)$$

for $\mathscr{A}^{\mathrm{disc}}_{2s,M}$ or $\mathscr{A}^{\mathrm{disc,MC}}_{2s,M}$ respectively.

Again, the same strategy from Corollary 2 of widening the frequency band and shifting the one-dimensional transforms accordingly allows us to use the nonequispaced SFT algorithm from Theorem 1 in Algorithm 2. Note here that the widening and shifting occurs on a dimension by dimension basis so as to account for the differing one-dimensional frequencies of the form $\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell$ for $\mathbf{k} \in \mathscr{I}$.

Corollary 4 (Algorithm 2 with nonequispaced sublinear-time SFT) *For $\mathscr{I} \subset \mathscr{B}^d_N$, let $\tilde{M}$ be the larger one-dimensional bandwidth parameter from Corollary 2, and additionally define $\tilde{M}^\ell := 2\max_{\mathbf{k}\in\mathscr{I}} |\mathbf{k}'_\ell \cdot \mathbf{z}'_\ell|+1$. For $\Lambda(\mathbf{z}, M)$, a reconstructing rank-1 lattice for $\mathscr{I}$ with $M \leq \min\{\tilde{M}, \min_{\ell\in[d]} \tilde{M}^\ell\}$, and the function $f \in \mathscr{W}(\mathbb{T}^d)\cap C(\mathbb{T}^d)$, we consider applying Algorithm 2 where each function sample may be corrupted by noise at most $e_\infty \geq 0$ in absolute magnitude with the following modifications:*

1. *use the sublinear-time SFT algorithm $\mathscr{A}^{\mathrm{sub}}_{2s,\tilde{M}}$ or $\mathscr{A}^{\mathrm{sub,MC}}_{2s,\tilde{M}}$ in Line 1 and $\mathscr{A}^{\mathrm{sub}}_{2s,\tilde{M}^\ell}$ or $\mathscr{A}^{\mathrm{sub,MC}}_{2s,\tilde{M}^\ell}$ in Line* 4
2. *and only check equality against ω in Line* 14 *(rather than equivalence modulo M),*

to produce $\mathbf{b} = (b_{\mathbf{k}})_{\mathbf{k}\in\mathscr{B}^d_N}$ a $2s$-sparse approximation of c satisfying the error estimate

$$\|\mathbf{b}-c\|_{\ell^2(\mathbb{Z}^d)} = 98\left(\frac{\|c|_{\mathscr{I}}-(c|_{\mathscr{I}})^{\mathrm{opt}}_s\|_1}{\sqrt{s}}+\sqrt{s}\|c-c|_{\mathscr{I}}\|_1+\sqrt{s}e_\infty\right)+\|c-c|_{\mathscr{I}}\|_2$$

albeit with probability $1-\sigma \in [0, 1)$ for the Monte Carlo version.

Letting $\bar{M} = \max(\tilde{M}, \max_{\ell\in[d]} \tilde{M}^\ell)$, the total number of evaluations of f will be

$$\mathscr{O}\left(\frac{dNs^2\log^4\bar{M}}{\log s}\right)\ \textit{or}\ \mathscr{O}\left(dNs\log^3\bar{M}\log\left(\frac{dN\bar{M}}{\sigma}\right)\right)$$

with associated computational complexities

$$\mathscr{O}\left(dNs\left(\frac{s\log^4\bar{M}}{\log s}+N\log N\right)\right)\ \textit{or}\ \mathscr{O}\left(dNs\left(\log^3\bar{M}\log\left(\frac{dN\bar{M}}{\sigma}\right)+N\log N\right)\right)$$

for $\mathscr{A}^{\mathrm{sub}}_{2s,\cdot}$ and $\mathscr{A}^{\mathrm{sub,MC}}_{2s,\cdot}$ respectively.

Remark 7 The bounds in Remark 3 will still hold for $\tilde{M}^\ell$ as well; thus one of these upper bounds can be used as the effective bandwidth parameter for every SFT without having to calculate the $d+1$ bandwidths by scanning $\mathscr{I}$. Again however, if this scan is tolerable, one can reduce the overall complexity by using analogous minimal bandwidths discussed in Remark 3 along with corresponding frequency shifts.

5 Numerics

We now demonstrate the effectiveness of our phase encoding and two-dimensional DFT algorithms for computing Fourier coefficients of multivariate functions in a series of empirical tests. The two techniques are implemented in MATLAB, with the code for the algorithms and tests in this section publicly available.[1] The results below use a MATLAB implementation[2] of the randomized univariate sublinear-time nonequispaced algorithm $\mathscr{A}^{\text{sub,MC}}_{2s,M}$ (cf. Theorem 1) as the underlying SFT for both multivariate approaches as this allows for the fastest runtime and most sample efficient implementations.

In the univariate code, all parameters but one are qualitatively tuned below theoretical upper bounds to increase efficiency while maintaining accuracy and are kept constant between tests below. In particular, we fix the values $\texttt{C} := 1$, $\texttt{sigma} := 2/3$, and $\texttt{primeShift} := 0$ (see the documentation and the original paper [16] for more detail). The only parameter we vary is `randomScale` which affects the rate at which the deterministic algorithm $\mathscr{A}^{\text{sub}}_{2s,M}$ is randomly sampled to produce the Monte Carlo version $\mathscr{A}^{\text{sub,MC}}_{2s,M}$. This parameter represents a multiplicative scaling on logarithmic factors of the bandwidth which determines how many prime numbers are randomly selected from those used in the deterministic SFT implementation. Therefore, lower values of `randomScale` will result in using fewer prime numbers, decreasing the number of function samples and overall runtime at the risk of a higher probability of failure. We consider values well below the code default and theoretical upper bound of 21 given in [16].

5.1 Exactly sparse case

In the beginning, we consider the case of multivariate trigonometric polynomials with frequencies supported within hyperbolic cross index sets. We define the d-dimensional hyperbolic cross frequency set

$$\mathscr{H}^d_N := \left\{ \mathbf{k} \in \mathbb{Z}^d : \prod_{\ell=1}^{d} \max(1, |k_\ell|) \leq \frac{N}{2} \quad \text{and} \quad \max_{\ell=1,\ldots,d} k_\ell < \frac{N}{2} \right\} \subset \mathscr{B}^d_N$$

where the second condition ensures that $\mathscr{H}^d_N$ is of expansion $N \in \mathbb{N}$. For a given sparsity s, we choose s many frequencies uniformly at random from $\mathscr{H}^d_N$, and we

[1] Available at https://gitlab.com/grosscra/Rank1LatticeSparseFourier.

[2] Available at https://gitlab.com/grosscra/SublinearSparseFourierMATLAB.

randomly draw corresponding Fourier coefficients $c_{\mathbf{k}}$ from $[-1,1]+\mathrm{i}\,[-1,1]$, $|c_{\mathbf{k}}| \geq 10^{-3}$. For each parameter setting, we perform the tests 100 times.

Over these tests, we will determine the success rate as the percentage of times that all frequencies were correctly identified in the output. We focus on frequency identification since this is the core issue that Algorithms 1 and 2 solve, with the coefficient estimates carrying over directly from the SFT algorithm. Moreover, with the s most significant frequencies identified, any alternative method for quickly computing the corresponding Fourier coefficients (if those from $\mathscr{A}_{s,M}$ are not tolerable) can be performed. Nevertheless, see the experiments following those in Sect. 5.1.1 for examples where we compute ℓ^2 errors in the coefficient vectors rather than just comparing frequencies.

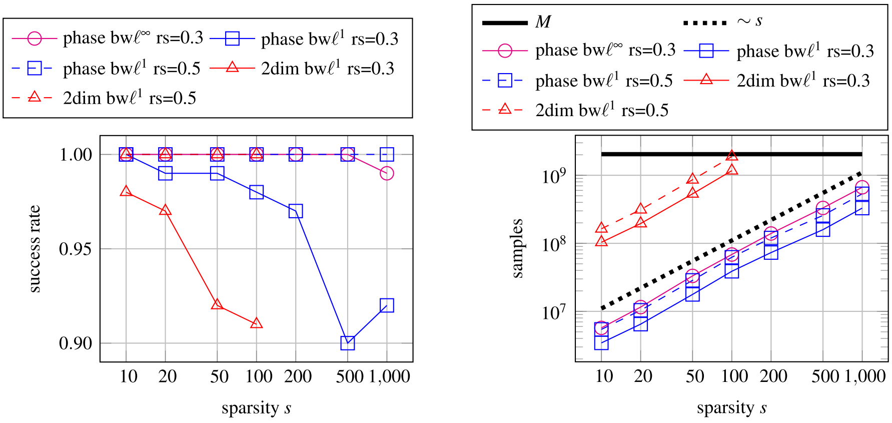
5.1.1 Random frequency sets within 10-dimensional hyperbolic cross and high-dimensional full cuboids

We set the spatial dimension $d := 10$, the expansion $N := 33$, and use $\mathscr{I} := \mathscr{H}_{33}^{10}$ as set of possible frequencies with cardinality $|\mathscr{I}| = 45\,548\,649$. Then, the rank-1 lattice with generating vector

$$\begin{aligned}\mathbf{z} :=&(1,\ 33,\ 579,\ 3\,628,\ 21\,944,\ 169\,230,\\ &\quad 1\,105\,193,\ 7\,798\,320,\ 49\,768\,670,\ 320\,144\,128)^{\top}\end{aligned} \tag{7}$$

and lattice size $M := 2\,040\,484\,044$ is a reconstructing one. We apply Algorithms 1 and 2 with the SFT algorithm $\mathscr{A}_{2s,\tilde{M}}^{\text{sub,MC}}$.

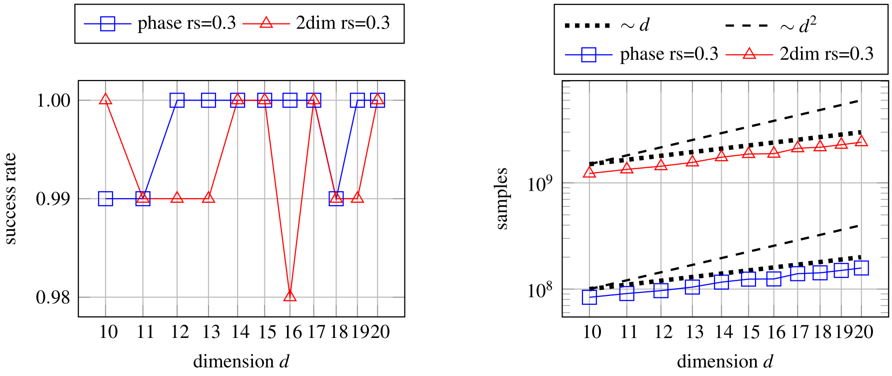
In Fig. 4a, the success rate over 100 test runs is plotted against the sparsities $s \in \{10, 20, 50, 100, 200, 500, 1000\}$ for Algorithm 1 and $s \in \{10, 20, 50, 100\}$ for Algorithm 2. In Fig. 4b, the average numbers of samples over 100 tests are reported. The magenta line with circles corresponds to Algorithm 1 with bandwidth parameter $\tilde{M} = dNM \approx 6.7 \cdot 10^{11}$ and `randomScale` $:= 0.3$. We observe that the number of samples grow nearly linearly with respect to the sparsity s. Moreover, the success rate is at least 0.99 (99 out of 100 test runs), where we define success such that the support of output (sparse coefficient vector) contains the true frequencies. Next, we reduce the bandwidth $\tilde{M}$ to $1 + 2\|\mathbf{z}\|_\infty \max_{\mathbf{k}\in\mathscr{I}} \|\mathbf{k}\|_1 \approx 1.6 \cdot 10^{10}$, see also Remark 3, and visualize this as solid blue line with squares. This smaller bandwidth causes a decrease in the number of samples of up to 50 percent while only mildly decreasing the success rates to values not below 0.90. Increasing the `randomScale` parameter to 0.5, denoted by dashed blue line with squares, raises the success rate to 1.00 while achieving still fewer samples than bandwidth parameter $\tilde{M} = dNM \approx 6.7 \cdot 10^{11}$ and `randomScale` $= 0.3$ (solid magenta line with circles). The numbers of samples for Algorithm 2 are plotted as solid and dashed red lines with triangles for `randomScale` $:= 0.3$ and 0.5, respectively, choosing the bandwidth $\tilde{M} := 1 + 2\|\mathbf{z}\|_\infty \max_{\mathbf{k}\in\mathscr{I}} \|\mathbf{k}\|_1 \approx 1.6 \cdot 10^{10}$. We observe that Algorithm 2 requires much more samples, more than one order of magnitude, compared to Algorithm 1, while achieving similar success rates. For comparison, in case of sparsity $s = 100$ and `randomScale` $= 0.5$, Algorithm 2 takes almost $M = 2\,040\,484\,044$ samples,



(a) Success rates vs. sparsity s.

(b) Samples vs. sparsity s.

Fig. 4 Success rates and average number of samples over 100 test runs for Algorithm 1 with $\mathscr{A}^{\text{sub,MC}}_{2s,\tilde{M}}$, denoted by "phase", and Algorithm 2 with $\mathscr{A}^{\text{sub,MC}}_{2s,\tilde{M}}$, denoted by "2dim", on random multivariate trigonometric polynomials, setting `randomScale` := rs. Random frequencies are chosen from hyperbolic cross $\mathscr{I} := \mathscr{H}^{10}_{33}$. "bw$\ell^\infty$" and "bw$\ell^1$" correspond to bandwidth parameters $\tilde{M} = dNM \approx 6.7 \cdot 10^{11}$ and $\tilde{M} = 1 + 2\|\mathbf{z}\|_\infty \max_{\mathbf{k}\in\mathscr{I}} \|\mathbf{k}\|_1 \approx 1.6 \cdot 10^{10}$, respectively



(a) Success rate vs. spatial dimension d.

(b) Samples vs. spatial dimension d.

Fig. 5 Average number of samples over 100 test runs for Algorithm 1 with $\mathscr{A}^{\text{sub,MC}}_{2s,\tilde{M}}$, denoted by "phase", and Algorithm 2 with $\mathscr{A}^{\text{sub,MC}}_{2s,\tilde{M}}$, denoted by "2dim", on random multivariate trigonometric polynomials, setting `randomScale` := rs. Random frequencies are chosen from **full cuboid** $\mathscr{I}$ of cardinality $|\mathscr{I}| \approx 10^{12}$ with corresponding lattice of size $M = |\mathscr{I}|$ and bandwidth parameter $\tilde{M} = M$

where the latter would be required by a non-SFT approach which uses all rank-1 lattice nodes.

In Fig. 5b, we investigate the dependence of the required number of samples of Algorithms 1 and 2 on the spatial dimension d, where we consider the values $d \in \{10, 11, \dots, 20\}$. As before, the success rates are reported in Fig. 5a. For this, we use a slightly different setting, where we choose $s = 100$ random frequencies from a full cuboid of cardinality $\approx 10^{12}$. Note that a cuboid with edge lengths $N_1, N_2, \ldots, N_d$ has the rank-1 lattice construction

$$\mathbf{z} = (1, N_1, N_1 \cdot N_2, \ldots, N_1 \cdot N_2 \cdots N_{d-1}) = \left(\prod_{j=1}^{\ell-1} N_j \right)_{\ell=1}^{d}$$

with lattice size $M = \prod_{\ell=1}^{d} N_\ell = |\mathscr{I}|$. The main benefit of this construction is that the map $\mathbf{k} \mapsto \mathbf{k} \cdot \mathbf{z}$ is a bijection between $\mathscr{I}$ and $\mathscr{B}_M$. Thus, the one-dimensional bandwidth parameter $\tilde{M} = 2 \max_{\mathbf{k} \in \mathscr{I}} |\mathbf{k} \cdot \mathbf{z}| + 1$ (which is usually larger than M) in this case coincides with $M = |\mathscr{I}|$. By choosing cuboids in this experiment which have approximately the same cardinality, we remove any dependence on $\tilde{M}$ in our experiments, allowing us to focus on the dependence on d.

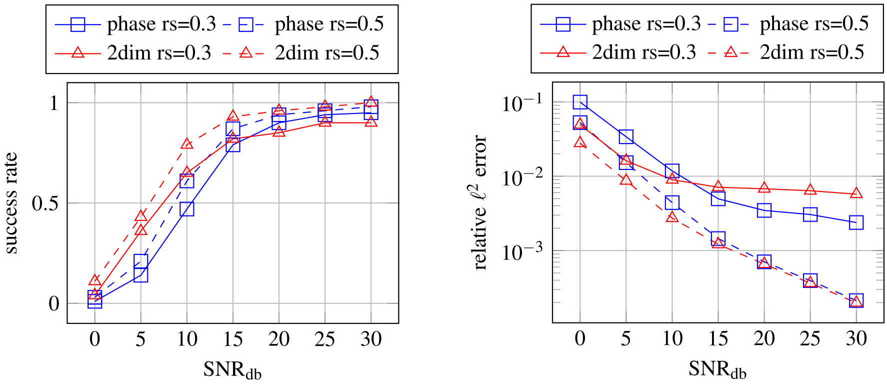
In our examples, the cuboids are constructed by manually tuning the edge lengths for each dimension so that the total cardinality is $\approx 10^{12}$. One way to start this procedure is by computing $(10^{12})^{1/d}$ and then choosing d edge lengths that approximately average to this value. From here, the edge lengths can be qualitatively tweaked to arrive at a cuboid of the desired size. For instance, we utilize the cuboid $\mathscr{I} := \{-8, -7, \ldots, 7\}^9 \times \{-7, -6, \ldots, 7\}$, $|\mathscr{I}| \approx 1.03 \cdot 10^{12}$, in the case $d = 10$ and $\mathscr{I} := \{-2, -1, \ldots, 2\} \times \{-2, -1, 0, 1\}^{18} \times \{-1, 0, 1\}$, $|\mathscr{I}| \approx 1.03 \cdot 10^{12}$, for $d = 20$. Since the expansion N is a factor in the number of samples of Algorithm 2, cf. Corollary 4, and we want to concentrate on the dependence on the spatial dimension d, we now fix this parameter to $N := 16$ independent of d. Moreover, the `randomScale` parameter is set to 0.3. The plots indicate that the numbers of samples grow approximately linearly with respect to the dimension d as stated by Corollaries 2 and 4 for Algorithms 1 and 2, respectively. The success rates are slightly better compared to the tests from Fig. 4a.

5.1.2 Random frequency sets within 10-dimensional hyperbolic cross and noisy samples

In this section, we again consider random multivariate trigonometric polynomials with frequencies supported within the hyperbolic cross index set $\mathscr{H}_{33}^{10}$ of expansion $N = 33$ and use the reconstructing rank-1 lattice with generating vector $\mathbf{z}$ as stated in (7) and size $M := 2\,040\,484\,044$. Similarly as in [21, Section 5.2], we perturb the samples of the trigonometric polynomial by additive complex (white) Gaussian noise $\varepsilon_j \in \mathbb{C}$ with zero mean and standard deviation σ. The noise is generated by $\varepsilon_j := \sigma/\sqrt{2}\left(\varepsilon_{1,j} + \mathrm{i}\varepsilon_{2,j}\right)$ where $\varepsilon_{1,j}, \varepsilon_{2,j}$ are independent standard normal distributed. Since the signal-to-noise ratio (SNR) can be approximately computed by

$$\mathrm{SNR} \approx \frac{\sum_{j=0}^{M-1} |f(\mathbf{x}_j)|^2/M}{\sum_{j=0}^{M-1} |\varepsilon_j|^2/M} \approx \frac{\sum_{\mathbf{k} \in \mathrm{supp}(c)} |c_{\mathbf{k}}(f)|^2}{\sigma^2},$$

this leads to the choice $\sigma := \sqrt{\sum_{\mathbf{k} \in \mathrm{supp}(c)} |c_{\mathbf{k}}(f)|^2} / \sqrt{\mathrm{SNR}}$ for a targeted SNR value. The SNR is often expressed in the logarithmic decibel scale (dB), with $\mathrm{SNR}_{\mathrm{dB}} =$

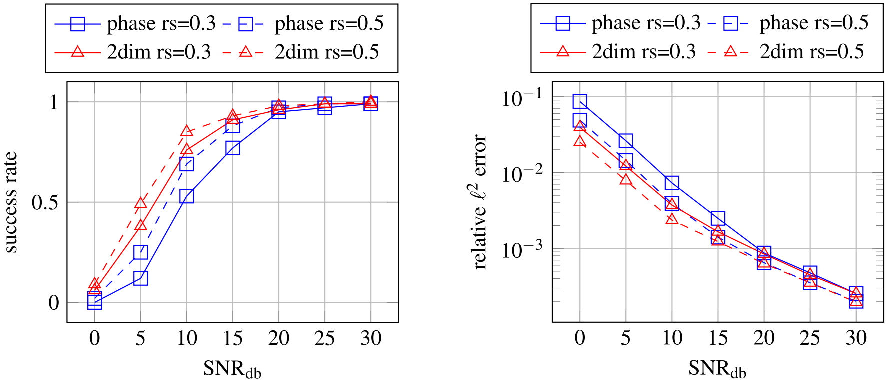


(a) success rates vs. noise level for $s = 100$ (b) relative ℓ^2 errors vs. noise level for $s = 100$

Fig. 6 Average success rates (all frequencies detected) and relative ℓ^2 errors over 100 test runs for Algorithm 1 with $\mathscr{A}^{\mathrm{sub,MC}}_{2s,\tilde{M}}$, denoted by "phase", and Algorithm 2 with $\mathscr{A}^{\mathrm{sub,MC}}_{2s,\tilde{M}}$, denoted by "2dim", on random multivariate trigonometric polynomials within hyperbolic cross $\mathscr{I} := \mathscr{H}^{10}_{33}$, setting `randomScale` := rs $\in \{0.3, 0.5\}$ and bandwidth parameter $\tilde{M} = 1 + 2\|\mathbf{z}\|_\infty \max_{\mathbf{k}\in\mathscr{I}} \|\mathbf{k}\|_1 \approx 1.6 \cdot 10^{10}$

$10 \log_{10}$ SNR and SNR $= 10^{\mathrm{SNR_{dB}}/10}$, i.e., a linear SNR $= 10^2$ corresponds to a logarithmic SNR$_{\mathrm{dB}} = 20$ and SNR $= 10^3$ corresponds to SNR$_{\mathrm{dB}} = 30$. Here, our tests use sparsity $s = 100$ and signal-to-noise ratios SNR$_{\mathrm{dB}} \in \{0, 5, 10, 15, 20, 25, 30\}$. Moreover, we only use the bandwidth parameter $\tilde{M} = 1 + 2\|\mathbf{z}\|_\infty \max_{\mathbf{k}\in\mathscr{I}} \|\mathbf{k}\|_1 \approx 1.6 \cdot 10^{10}$. Besides that, we choose the algorithm parameters as in Fig. 4.

In Fig. 6a, we visualize the success rates in dependence on the noise level. For `randomScale` $\in \{0.3, 0.5\}$ and both algorithms, the success rates start at less than 0.12 for SNR$_{\mathrm{dB}} = 0$ and grow for increasing signal-to-noise ratios until at least 0.90 for SNR$_{\mathrm{dB}} = 30$. The success rates of Algorithm 2 with $\mathscr{A}^{\mathrm{sub,MC}}_{2s,\tilde{M}}$ ("2dim") are often higher than for Algorithm 1 with $\mathscr{A}^{\mathrm{sub,MC}}_{2s,\tilde{M}}$ ("phase"), which may be caused by the larger numbers of samples for Algorithm 2 and the noise model used. Note that the numbers of samples correspond to those in Fig. 4b for $s = 100$ independent of the noise level. For Algorithm 2 with `randomScale` $= 0.3$, the increase of the success rate seems to stagnate at SNR$_{\mathrm{dB}} = 20$, while this does not seem to be the case for `randomScale` $= 0.5$ or Algorithm 1. In particular, this behavior can also be observed in Fig. 6b, where we plot the average relative ℓ^2 error of the Fourier coefficients against the signal-to-noise ratio. Here, we observe that for `randomScale` $= 0.3$, the decrease of the errors for increasing SNR$_{\mathrm{dB}}$ values almost stops once reaching SNR$_{\mathrm{dB}} = 20$ for both algorithms. Initially, the average error of Algorithm 2 is smaller, but at SNR$_{\mathrm{dB}} = 15$ and higher, the average error of Algorithm 1 is smaller. In case of `randomScale` $= 0.5$, we observe a distinct decrease for growing signal-to-noise ratios for both algorithms.



(a) success rates vs. noise level for $s = 100$ (b) relative ℓ^2 errors vs. noise level for $s = 100$

Fig. 7 Average success rates (all frequencies detected) and relative ℓ^2 errors over 100 test runs for Algorithm 1 with $\mathscr{A}^{\text{sub,MC}}_{2s,\tilde{M}}$, denoted by "phase", and Algorithm 2 with $\mathscr{A}^{\text{sub,MC}}_{2s,\tilde{M}}$, denoted by "2dim", on multivariate trigonometric polynomials with (deterministic) frequencies on weighted hyperbolic cross within hyperbolic cross $\mathscr{I} := \mathscr{H}^{10}_{33}$, setting `randomScale` $:= \text{rs} \in \{0.3, 0.5\}$ and bandwidth parameter $\tilde{M} = 1 + 2\|\mathbf{z}\|_\infty \max_{\mathbf{k}\in\mathscr{I}} \|\mathbf{k}\|_1 \approx 1.6 \cdot 10^{10}$

5.1.3 Deterministic frequency set within 10-dimensional hyperbolic cross and noisy samples

Next, instead of randomly chosen frequencies, we now consider frequencies on a d-dimensional weighted hyperbolic cross

$$\mathscr{H}^{d,\alpha}_N := \left\{ \mathbf{k} \in \mathbb{Z}^d : \prod_{\ell=1}^{d} \max(1, \ell^\alpha\, |k_\ell|) \leq \frac{N}{2} \quad \text{and} \quad \max_{\ell=1,\dots,d} k_\ell < \frac{N}{2} \right\}.$$

Here, we use $d = 10$, $N = 33$, $\mathscr{I} := \mathscr{H}^{10}_{33}$, and $\alpha = 1.7$, which yields $s = |\mathscr{H}^{10,1.7}_{33}| = 101$. Again, the Fourier coefficients $c_{\mathbf{k}}$ are randomly chosen from $[-1, 1] + \mathbb{i}\,[-1, 1]$, $|c_{\mathbf{k}}| \geq 10^{-3}$. We use the same lattice and bandwidth parameter as in the last subsection as well as the same noise model and parameters.

In Fig. 7, we depict the obtained results. In particular, the results in Fig. 7a are very similar to the ones for randomly chosen frequencies in Fig. 6a. For the case of deterministic frequencies in Fig. 7a, the success rates are slightly better. Moreover, we do not observe the "stagnation" of the success rates for Algorithm 2 with `randomScale` $= 0.3$. Correspondingly, the relative ℓ^2 errors, as shown in Fig. 7b, decrease distinctly for growing signal-to-noise ratios. Algorithm 2 performs slightly better than Algorithm 1, but also requires more than one order of magnitude more samples, similar to the results shown in Fig. 4b for $s = 100$.

5.2 Compressible case in 10 dimensions

In this section, we apply the methods on a test function which is not exactly sparse but compressible. In addition, we also consider noisy samples as in Sect. 5.1.2. We use the 10-variate periodic test function $f\colon \mathbb{T}^{10} \to \mathbb{R}$,

$$f(\mathbf{x}) := \prod_{\ell\in\{0,2,7\}} N_2(x_\ell) + \prod_{\ell\in\{1,4,5,9\}} N_4(x_\ell) + \prod_{\ell\in\{3,6,8\}} N_6(x_\ell), \tag{8}$$

from [33, Section 3.3] and [21, Section 5.3] with infinitely many non-zero Fourier coefficients $c_{\mathbf{k}}(f)$, where $N_m : \mathbb{T} \to \mathbb{R}$ is the B-Spline of order $m \in \mathbb{N}$,

$$N_m(x) := C_m \sum_{k\in\mathbb{Z}} \operatorname{sinc}\left(\frac{\pi}{m}k\right)^m (-1)^k \, \mathrm{e}^{2\pi \mathrm{i} kx},$$

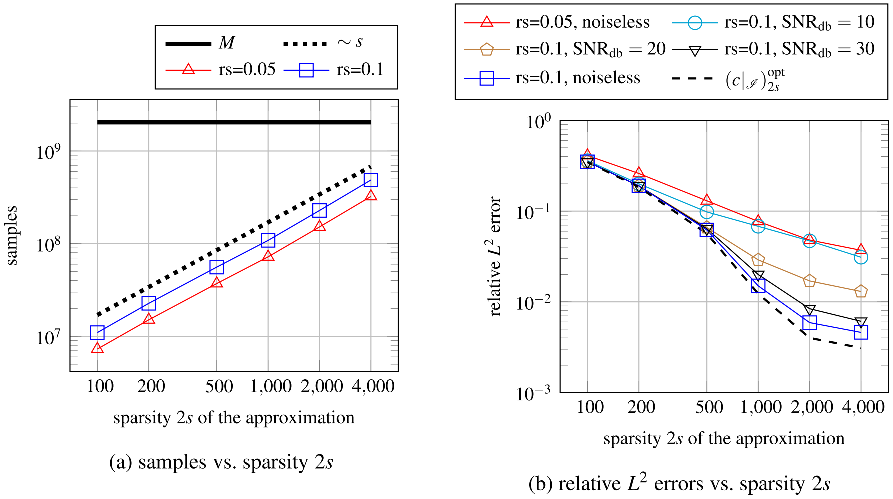
with a constant $C_m > 0$ such that $\|N_m\|_{L^2(\mathbb{T})} = 1$. We remark that each B-Spline N_m of order $m \in \mathbb{N}$ is a piece-wise polynomial of degree $m-1$. We apply Algorithm 1 with $\mathscr{A}^{\text{sub,MC}}_{2s,\tilde{M}}$ and use the sparsity parameters $s \in \{50, 100, 250, 500, 1000, 2000\}$, which corresponds to $2s \in \{100, 200, 500, 1000, 2000, 4000\}$ frequencies and Fourier coefficients for the output of Algorithm 1. We use the frequency set $\mathscr{I} := \mathscr{H}^{10}_{33}$ and `randomScale` := rs $\in \{0.05, 0.1\}$. Moreover, we work with the same rank-1 lattice as in Sect. 5.1.2.

The obtained basis index sets supp($\mathbf{b}$) should "consist of" the union of three lower dimensional manifolds, a three-dimensional hyperbolic cross in the dimensions 1, 3, 8; a four-dimensional hyperbolic cross in the dimensions 2, 5, 6, 10; and a three-dimensional hyperbolic cross in the dimensions 4, 7, 9. All tests are performed 100 times and the relative L^2 approximation error

$$\frac{\|f - \sum_{\mathbf{k}\in\text{supp}(\mathbf{b})} b_{\mathbf{k}} \, \mathrm{e}^{2\pi \mathrm{i}\mathbf{k}\cdot\circ}\|_{L^2}}{\|f\|_{L^2}} = \frac{\sqrt{\|f\|^2_{L^2} - \sum_{\mathbf{k}\in\text{supp}(\mathbf{b})} |c_{\mathbf{k}}(f)|^2 + \sum_{\mathbf{k}\in\text{supp}(\mathbf{b})} |b_{\mathbf{k}} - c_{\mathbf{k}}(f)|^2}}{\|f\|_{L^2}}$$

is computed each time.

In Fig. 8a, we visualize the average number of samples against the sparsity $2s$ of the approximation. We observe an almost linear increase with respect to $2s$. In Fig. 8b, we show the average relative errors for `randomScale` $\in \{0.05, 0.1\}$ in the noiseless case as well as `randomScale` $= 0.1$ for $\text{SNR}_{\text{db}} \in \{10, 20, 30\}$. In general, for increasing sparsity, the errors become smaller. For `randomScale` $= 0.05$ in the noiseless case and `randomScale` $= 0.1$ with $\text{SNR}_{\text{db}} = 10$, the average error are similar and stay above $3 \cdot 10^{-2}$ even for sparsity $2s = 4000$. For higher signal-to-noise ratio, the error decreases further. For $\text{SNR}_{\text{db}} = 30$, the obtained average error is $6.1 \cdot 10^{-3}$ for $2s = 4000$, which is only approximately twice as high as the best possible error when using the $2s$ largest (by magnitude) Fourier coefficients $c_{\mathbf{k}}(f)$



(a) samples vs. sparsity $2s$

(b) relative L^2 errors vs. sparsity $2s$

Fig. 8 Average number of samples and relative L^2 errors over 100 test runs for Algorithm 1 with $\mathscr{A}^{\text{sub,MC}}_{2s,\tilde{M}}$ on 10-dimensional test function (8) consisting of tensor products of B-Splines of different order. Search space is unweighted hyperbolic cross $\mathscr{I} := \mathscr{H}^{10}_{33}$ with SFT parameters `randomScale` := rs ∈ {0.05, 0.1} and $\tilde{M} = 1 + 2\|\mathbf{z}\|_\infty \max_{\mathbf{k}\in\mathscr{I}} \|\mathbf{k}\|_1 \approx 1.6 \cdot 10^{10}$

with the restriction $\mathbf{k} \in \mathscr{I} := \mathscr{H}^{10}_{33}$. The latter is plotted in Fig. 8b as dashed line without markers.

6 Conclusion

In this paper, we have presented two different methods for combining rank-1 lattices with univariate sparse Fourier transforms. By strategically transforming known reconstructing rank-1 lattices $\Lambda(\mathbf{z}, M)$, these techniques quickly and accurately manage the inversion of the frequency mapping $\mathbf{k} \mapsto \mathbf{k} \cdot \mathbf{z} \bmod M$. As such, we are able to transfer the error bounds and fast runtime from any suitably accurate sparse Fourier transform to problems in high-dimensional Fourier approximation.

In particular, we theoretically and numerically observe s-sparse recovery of d dimensional signals with computational complexities linear in s and d. These computational complexities paired with our techniques' flexibility with respect to a frequency set of interest rival the state of the art in high-dimensional Fourier analysis. And indeed, the extensibility of our techniques allow for new, and potentially more efficient results when joined with other univariate sparse Fourier transform algorithms.

We hope to observe the benefits of these algorithms in future applications. One such approach currently being investigated is extending standard Fourier spectral methods for solving partial differential equations. Replacing traditional Fourier techniques on grids with our faster, more flexible approaches allow us to sparsify the spectral method matrix equations to be solved. The size of this sparser matrix equation then has greatly reduced dependence on the dimension, and we expect our methods to be able to approximate solutions to partial differential equations in much higher dimensions than have been feasible by traditional methods.

Funding Craig Gross and Mark Iwen were supported in part by the National Science Foundation (NSF) DMS 1912706. Toni Volkmer was supported in part by Sächsische Aufbaubank-Förderbank-(SAB) 100378180. Lutz Kämmerer was supported by the Deutsche Forschugsgemeinschaft (DFG, German Research Foundation), project number 380648269.

Code availability All code is available at https://gitlab.com/grosscra/Rank1LatticeSparseFourier.

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

A Proof of Theorem 1

We begin with a lemma which will be used to slightly improve the error bounds in Theorems 1 and 2 from their original statements.

Lemma 4 *For $\mathbf{x} \in \mathbb{C}^K$ and $\mathscr{S}_\tau := \{k \in [K] \mid |x_k| \geq \tau\}$, if $\tau \geq \frac{\|\mathbf{x}-\mathbf{x}_s^{\mathrm{opt}}\|_1}{s}$, then $|\mathscr{S}_\tau| \leq 2s$ and*

$$\|\mathbf{x} - \mathbf{x}|_{\mathscr{S}_\tau}\|_2 \leq \|\mathbf{x} - \mathbf{x}_{2s}^{\mathrm{opt}}\|_2 + \tau\sqrt{2s}.$$

Proof Ordering the entries of $\mathbf{x}$ in descending order (with ties broken arbitrarily) as $|x_{k_1}| \geq |x_{k_2}| \geq \ldots$, we first note that

$$\|\mathbf{x} - \mathbf{x}_s^{\mathrm{opt}}\|_1 \geq \sum_{j=s+1}^{2s} |x_{k_j}| \geq s|x_{k_{2s}}|.$$

By assumption then, $\tau \geq |x_{k_{2s}}|$, and since $\mathscr{S}_\tau$ contains the $|\mathscr{S}_\tau|$-many largest entries of $\mathbf{x}$, we must have $\mathscr{S}_\tau \subset \mathrm{supp}(\mathbf{x}_{2s}^{\mathrm{opt}})$. Note then that $|\mathscr{S}_\tau| \leq 2s$. Finally, we calculate

$$\begin{aligned}\|\mathbf{x} - \mathbf{x}|_{\mathscr{S}_\tau}\|_2 &\leq \|\mathbf{x} - \mathbf{x}_{2s}^{\mathrm{opt}}\|_2 + \|\mathbf{x}_{2s}^{\mathrm{opt}} - \mathbf{x}_{\mathscr{S}_\tau}\|_2 \\ &\leq \|\mathbf{x} - \mathbf{x}_{2s}^{\mathrm{opt}}\|_2 + \sqrt{\sum_{k \in \mathrm{supp}(\mathbf{x}_{2s}^{\mathrm{opt}}) \setminus \mathscr{S}_\tau} x_k^2} \\ &\leq \|\mathbf{x} - \mathbf{x}_{2s}^{\mathrm{opt}}\|_2 + \tau\sqrt{2s}\end{aligned}$$

completing the proof. □

We now give the proof of Theorem 1. For convenience we restate the theorem.

Theorem 1 *For a signal $a \in \mathscr{W}(\mathbb{T}) \cap C(\mathbb{T})$ corrupted by some arbitrary noise $\mu : \mathbb{T} \to \mathbb{C}$, Algorithm 3 of* [16]*, denoted $\mathscr{A}^{\text{sub}}_{2s,M}$, will output a $2s$-sparse coefficient vector $\mathbf{v} \in \mathbb{C}^M$ which*

1. *reconstructs every frequency of $\hat{\mathbf{a}} \in \mathbb{C}^M$, $\omega \in \mathscr{B}_M$, with corresponding Fourier coefficients meeting the tolerance*

$$|\hat{a}_\omega| > (4 + 2\sqrt{2}) \left(\frac{\|\hat{\mathbf{a}} - \hat{\mathbf{a}}^{\text{opt}}_s\|_1}{s} + \|\hat{a} - \hat{\mathbf{a}}\|_1 + \|\mu\|_\infty \right),$$

2. *satisfies the ℓ^∞ error estimate for recovered coefficients*

$$\|(\hat{\mathbf{a}} - \mathbf{v})|_{\text{supp}(\mathbf{v})}\|_\infty \leq \sqrt{2} \left(\frac{\|\hat{\mathbf{a}} - \hat{\mathbf{a}}^{\text{opt}}_s\|_1}{s} + \|\hat{a} - \hat{\mathbf{a}}\|_1 + \|\mu\|_\infty \right),$$

3. *satisfies the ℓ^2 error estimate*

$$\|\hat{\mathbf{a}} - \mathbf{v}\|_2 \leq \|\hat{\mathbf{a}} - \hat{\mathbf{a}}^{\text{opt}}_{2s}\|_2 + \frac{(8\sqrt{2} + 6)\|\hat{\mathbf{a}} - \hat{\mathbf{a}}^{\text{opt}}_s\|_1}{\sqrt{s}} + (8\sqrt{2} + 6)\sqrt{s}(\|\hat{a} - \hat{\mathbf{a}}\|_1 + \|\mu\|_\infty),$$

4. *and the number of required samples of a and the operation count for $\mathscr{A}^{\text{sub}}_{2s,M}$ are*

$$\mathscr{O}\left(\frac{s^2 \log^4 M}{\log s} \right).$$

The Monte Carlo variant of $\mathscr{A}^{\text{sub}}_{2s,M}$, denoted $\mathscr{A}^{\text{sub,MC}}_{2s,M}$, referred to by Corollary 4 *of* [16] *satisfies all of the conditions (1)–(3) simultaneously with probability $(1 - \sigma) \in [2/3, 1)$ and has number of required samples and operation count*

$$\mathscr{O}\left(s \log^3(M) \log\left(\frac{M}{\sigma} \right) \right).$$

The samples required by $\mathscr{A}^{\text{sub,MC}}_{2s,M}$ are a subset of those required by $\mathscr{A}^{\text{sub}}_{2s,M}$.

Proof We refer to [16, Theorem 7] and its modification for noise robustness in [27, Lemma 4] for the proofs of properties (2) and (4). As for (1), [16, Lemma 6] and its modification in [27, Lemma 4] imply that any $\omega \in \mathscr{B}_M$ with $|\hat{a}_\omega| > 4(\|\hat{\mathbf{a}} - \hat{\mathbf{a}}^{\text{opt}}_s\|_1/s + \|\hat{a} - \hat{\mathbf{a}}\|_1 + \|\mu\|_\infty) =: 4\delta$ will be identified in [16, Algorithm 3]. An approximate Fourier coefficient for these and any other recovered frequencies is stored in the vector $\mathbf{x}$ which satisfies the same estimate in property (2) by the proof of [16, Theorem 7] and [27, Lemma 4]. However, only the $2s$ largest magnitude values of $\mathbf{x}$ will be returned in $\mathbf{v}$. We therefore analyze what happens when some of the potentially large Fourier coefficients corresponding to frequencies in $\mathscr{S}_{4\delta}$ do not have their approximations assigned to $\mathbf{v}$.

For the definition of $\mathscr{S}_\tau$ in Lemma 4 applied to $\hat{\mathbf{a}}$, we must have $|\mathscr{S}_{4\delta}| \leq 2s = |\operatorname{supp}(\mathbf{v})|$. If $\omega \in \mathscr{S}_{4\delta} \setminus \operatorname{supp}(\mathbf{v})$, there must then exist some other $\omega' \in \operatorname{supp}(\mathbf{v}) \setminus \mathscr{S}_{4\delta}$ which was identified and took the place of ω in $\operatorname{supp}(\mathbf{v})$. For this to happen, $|\hat{a}_{\omega'}| \leq 4\delta$ and $|x_{\omega'}| \geq |x_\omega|$. But by property (2) (extended to all coefficients in $\mathbf{x}$), we know

$$4\delta + \sqrt{2}\delta \geq |\hat{a}_{\omega'}| + \sqrt{2}\delta \geq |x_{\omega'}| \geq |x_\omega| \geq |\hat{a}_\omega| - \sqrt{2}\delta.$$

Thus, any frequency in $\mathscr{S}_{4\delta}$ not chosen satisfies $|\hat{a}_\omega| \leq (4 + 2\sqrt{2})\delta$, and so every frequency in $\mathscr{S}_{(4+2\sqrt{2})\delta}$ is in fact identified in $\mathbf{v}$ verifying property (1).

As for property (3), we estimate the ℓ^2 error using property (2), Lemma 4, and the above argument as

$$\begin{aligned}
\|\hat{\mathbf{a}} - \mathbf{v}\|_2 &\leq \|\hat{\mathbf{a}} - \hat{\mathbf{a}}|_{\operatorname{supp}(\mathbf{v})}\|_2 + \|(\hat{\mathbf{a}} - \mathbf{v})|_{\operatorname{supp}(\mathbf{v})}\|_2 \\
&\leq \|\hat{\mathbf{a}} - \hat{\mathbf{a}}|_{\mathscr{S}_{4\delta} \cap \operatorname{supp}(\mathbf{v})}\|_2 + \sqrt{2}\delta\sqrt{2s} \\
&\leq \|\hat{\mathbf{a}} - \hat{\mathbf{a}}|_{\mathscr{S}_{4\delta}}\|_2 + \|\hat{\mathbf{a}}|_{\mathscr{S}_{4\delta} \setminus \operatorname{supp}(\mathbf{v})}\|_2 + 2\delta\sqrt{s} \\
&\leq \|\hat{\mathbf{a}} - \hat{\mathbf{a}}_{2s}^{\mathrm{opt}}\|_2 + 4\delta\sqrt{2s} + (4 + 2\sqrt{2})\delta\sqrt{2s} + 2\delta\sqrt{s} \\
&= \|\hat{\mathbf{a}} - \hat{\mathbf{a}}_{2s}^{\mathrm{opt}}\|_2 + (8\sqrt{2} + 6)\sqrt{s}\delta
\end{aligned}$$

as desired. □

B Proof of Theorem 2

We again restate the theorem for convenience.

Theorem 2 *For a signal $a \in \mathscr{W}(\mathbb{T}) \cap C(\mathbb{T})$ corrupted by some arbitrary noise $\mu : \mathbb{T} \to \mathbb{C}$, and $1 \leq r \leq \frac{M}{36}$ Algorithm 1 of* [27]*, denoted $\mathscr{A}_{2s,M}^{\mathrm{disc}}$, will output a $2s$-sparse coefficient vector $\mathbf{v} \in \mathbb{C}^M$ which*

1. *reconstructs every frequency of $\mathbf{F}_M\, \mathbf{a} \in \mathbb{C}^M$, $\omega \in \mathscr{B}_M$, with corresponding aliased Fourier coefficient meeting the tolerance*

$$|(\mathbf{F}_M\, \mathbf{a})_\omega| > 12(1 + \sqrt{2})\left(\frac{\|\mathbf{F}_M\, \mathbf{a} - (\mathbf{F}_M\, \mathbf{a})_s^{\mathrm{opt}}\|_1}{2s} + 2(\|\mathbf{a}\|_\infty M^{-r} + \|\boldsymbol{\mu}\|_\infty)\right),$$

2. *satisfies the ℓ^∞ error estimate for recovered coefficients*

$$\|(\mathbf{F}_M\, \mathbf{a} - \mathbf{v})|_{\operatorname{supp}(\mathbf{v})}\|_\infty \leq 3\sqrt{2}\left(\frac{\|\mathbf{F}_M\, \mathbf{a} - (\mathbf{F}_M\, \mathbf{a})_s^{\mathrm{opt}}\|_1}{2s} + 2(\|\mathbf{a}\|_\infty M^{-r} + \|\boldsymbol{\mu}\|_\infty)\right),$$

3. *satisfies the ℓ^2 error estimate*

$$\|\mathbf{F}_M\,\mathbf{a}-\mathbf{v}\|_2 \leq \|\mathbf{F}_M\,\mathbf{a}-(\mathbf{F}_M\,\mathbf{a})_{2s}^{\mathrm{opt}}\|_2 + 38\frac{\|\mathbf{F}_M\,\mathbf{a}-(\mathbf{F}_M\,\mathbf{a})_s^{\mathrm{opt}}\|_1}{\sqrt{s}} + 152\sqrt{s}(\|\mathbf{a}\|_\infty M^{-r}+\|\boldsymbol{\mu}\|_\infty),$$

4. *and the number of required samples of* $\mathbf{a}$ *and the operation count for* $\mathscr{A}_{2s,M}^{\mathrm{disc}}$ *are*

$$\mathscr{O}\left(\frac{s^2 r^{3/2}\log^{11/2} M}{\log s}\right).$$

The Monte Carlo variant of $\mathscr{A}_{2s,M}^{\mathrm{disc}}$, *denoted* $\mathscr{A}_{2s,M}^{\mathrm{disc,MC}}$, *satisfies the all of the conditions* (1)–(3) *simultaneously with probability* $(1-\sigma)\in[2/3,1)$ *and has number of required samples and operation count*

$$\mathscr{O}\left(sr^{3/2}\log^{9/2}(M)\log\left(\frac{M}{\sigma}\right)\right).$$

Proof All notation in this proof matches that in [27] (in particular, we use f to denote the one-dimensional function in place of a in the theorem statement and $N=2M+1$). We begin by substituting the 2π-periodic gaussian filter given in (3) on page 756 with the 1-periodic gaussian and associated Fourier transform

$$g(x)=\frac{1}{c_1}\sum_{n=-\infty}^{\infty}\mathrm{e}^{-\frac{(2\pi)^2(x-n)^2}{2c_1^2}},\quad \hat{g}_\omega=\frac{1}{\sqrt{2\pi}}\mathrm{e}^{-\frac{c_1^2\omega^2}{2}}.$$

Note then that all results regarding the Fourier transform remain unchanged, and since this 1-periodic gaussian is a just a rescaling of the 2π-periodic one used in [27], the bound in [27, Lemma 1] holds with a similarly compressed gaussian, that is, for all $x\in\left[-\frac{1}{2},\frac{1}{2}\right]$

$$g(x)\leq\left(\frac{3}{c_1}+\frac{1}{\sqrt{2\pi}}\right)\mathrm{e}^{-\frac{(2\pi x)^2}{2c_1^2}}. \tag{9}$$

Analogous results up to and including [27, Lemma 10] for 1-periodic functions then hold straightforwardly.

Assuming that our signal measurements $\mathbf{f}=(f(y_j))_{j=0}^{2M}=(f(\frac{j}{N}))_{j=0}^{2M}$ are corrupted by some discrete noise $\boldsymbol{\mu}=(\mu_j)_{j=0}^{2M}$, we consider for any $x\in\mathbb{T}$ a similar bound to [27, Lemma 10]. Here, $j' := \arg\min_j |x - y_j|$ and $\kappa := \lceil \gamma \ln N \rceil + 1$ for some $\gamma \in \mathbb{R}^+$ to be determined. Then,

$$\begin{aligned}
&\left| \frac{1}{N} \sum_{j=0}^{2M} f(y_j) g(x - y_j) - \frac{1}{N} \sum_{j=j'-\kappa}^{j'+\kappa} (f(y_j) + \mu_j) g(x - y_j) \right| \\
&\quad \le \frac{1}{N} \left| \sum_{j=0}^{2M} f(y_j) g(x - y_j) - \sum_{j=j'-\kappa}^{j'+\kappa} f(y_j) g(x - y_j) \right| + \frac{1}{N} \left| \sum_{j=j'-\kappa}^{j'+\kappa} \mu_j g(x - y_j) \right| \\
&\quad \le \frac{1}{N} \left| \sum_{j=0}^{2M} f(y_j) g(x - y_j) - \sum_{j=j'-\kappa}^{j'+\kappa} f(y_j) g(x - y_j) \right| + \frac{\|\boldsymbol{\mu}\|_\infty}{N} \sum_{k=-\kappa}^{\kappa} g(x - y_{j'+k})
\end{aligned}$$

We bound the first term in this sum by a direct application of [27, Lemma 10]; however, we take this opportunity to reduce the constant in the bound given there. In particular, bounding this term by the final expression in the proof of [27, Lemma 10] and using our implicit assumption that $36 \le N$, we have

$$\begin{aligned}
&\left| \frac{1}{N} \sum_{j=0}^{2M} f(y_j) g(x - y_j) - \frac{1}{N} \sum_{j=j'-\kappa}^{j'+\kappa} (f(y_j) + \mu_j) g(x - y_j) \right| \\
&\qquad \le \left(\frac{3}{\sqrt{2\pi}} + \frac{1}{2\pi} \sqrt{\frac{\ln 36}{36}} \right) \|\mathbf{f}\|_\infty N^{-r} + \frac{1}{N} \|\boldsymbol{\mu}\|_\infty \sum_{k=-\kappa}^{\kappa} g(x - y_{j'+k}).
\end{aligned} \tag{10}$$

We now work on bounding the second term. First note that for all $k \in [-\kappa, \kappa] \cap \mathbb{Z}$,

$$g(x - y_{j' \pm k}) = g\left(x - y_{j'} \pm \frac{k}{N} \right).$$

Assuming without loss of generality that $0 \le x - y_{j'}$, we can bound the nonnegatively indexed summands by (9) as

$$g\left(x - y_{j'} + \frac{k}{N} \right) \le \left(\frac{3}{c_1} + \frac{2}{\sqrt{2\pi}} \right) \mathrm{e}^{-\frac{(2\pi)^2 k^2}{2 c_1^2 N^2}}. \tag{11}$$

For the negatively indexed summands, the definition of $j' = \arg\min_j |x - y_j|$ implies that $x - y_{j'} \le \frac{1}{2N}$. In particular,

$$x - y_{j'} - \frac{k}{N} \le \frac{1 - 2k}{2N} < 0$$

implies

$$\left(x - y_{j'} - \frac{k}{N} \right)^2 \ge \frac{1 - 2k}{2N} \left(x - y_{j'} - \frac{k}{N} \right) \ge \frac{2k - 1}{2N} \cdot \frac{k}{N},$$

giving

$$g\left(x - y_{j'} - \frac{k}{N}\right) \le \left(\frac{3}{c_1} + \frac{2}{\sqrt{2\pi}}\right) \mathbb{e}^{-\frac{(2\pi)^2 k^2}{2c_1^2 N^2}} \mathbb{e}^{\frac{(2\pi)^2 k}{4c_1^2 N^2}}. \tag{12}$$

We now bound the final exponential. We first recall from [27] the choices of parameters

$$c_1 = \frac{\beta\sqrt{\ln N}}{N}, \quad \kappa = \lceil \gamma \ln N \rceil + 1, \quad \gamma = \frac{6r}{\sqrt{2\pi}} = \frac{\beta\sqrt{r}}{2\sqrt{\pi}}, \quad \beta = 6\sqrt{r}$$

with $1 \le r \le \frac{N}{36}$. For $k \in [1, \kappa] \cap \mathbb{Z}$ then,

$$\begin{aligned}
\exp\left(\frac{(2\pi)^2 k}{4c_1^2 N^2}\right) &\le \exp\left(\frac{(2\pi)^2 \kappa}{4c_1^2 N^2}\right) \\
&\le \exp\left(\frac{\pi^2\left(\frac{6r\ln N}{\sqrt{2\pi}} + 2\right)}{36 r \ln N}\right) \\
&\le \exp\left(\frac{\pi}{6\sqrt{2}} + \frac{\pi^2}{18 r \ln N}\right) \\
&\le \exp\left(\frac{\pi}{6\sqrt{2}} + \frac{\pi^2}{18 \ln 36}\right) =: A.
\end{aligned}$$

Combining this with our bounds for the nonnegatively indexed summands (11) and the negatively indexed summands (12), we have

$$\frac{1}{N}\sum_{k=-\kappa}^{\kappa} g(x - y_{j'+k}) \le \left(\frac{3}{\beta\sqrt{\ln N}} + \frac{1}{N\sqrt{2\pi}}\right)\left(1 + (1+A)\sum_{k=1}^{\kappa} \mathbb{e}^{-\frac{(2\pi)^2 k^2}{2\beta^2 \ln N}}\right)$$

Expressing the final sum as a truncated lower Riemann sum and applying a change of variables on the resulting integral, we have

$$\sum_{k=1}^{\kappa} \mathbb{e}^{-\frac{(2\pi)^2 k^2}{2\beta^2 \ln N}} \le \frac{\beta\sqrt{\ln N}}{\sqrt{2\pi}} \int_0^\infty e^{-x^2}\, dx = \frac{\beta\sqrt{\ln N}}{2\sqrt{2\pi}}.$$

Making use of our parameter values from [27], and the fact that $1 \le r \le \frac{N}{36}$,

$$\begin{aligned}
\frac{1}{N}\sum_{k=-\kappa}^{\kappa} g(x - y_{j'+k}) &\le \left(\frac{3}{\beta\sqrt{\ln N}} + \frac{1}{N\sqrt{2\pi}}\right)\left(1 + \frac{1+A}{2\sqrt{2\pi}}\beta\sqrt{\ln N}\right) \\
&\le \frac{3}{6\sqrt{\ln 36}} + \frac{3(1+A)}{2\sqrt{2\pi}} + \frac{1}{36\sqrt{2\pi}} + \frac{1+A}{4\pi}\sqrt{\frac{\ln 36}{36}} \\
&< 2.
\end{aligned} \tag{13}$$

With our revised bound for (10) above, we reprove [27, Theorem 4] to estimate $g * f$ by the truncated discrete convolution with noisy samples. In particular, we apply [27, Theorem 3], (10), (9), and finally our same assumption that $1 \leq r \leq \frac{N}{36}$ to obtain

$$\begin{aligned}
&\left| (g * f)(x) - \frac{1}{N} \sum_{j=j'-\left\lceil \frac{6r}{\sqrt{2\pi}} \ln N \right\rceil -1}^{j'+\left\lceil \frac{6r}{\sqrt{2\pi}} \ln N \right\rceil +1} (f(y_j) + \mu_j) g(x - y_j) \right| \\
&\quad \leq \frac{N^{1-r}}{6\sqrt{r}\sqrt{\ln N}} \|\mathbf{f}\|_\infty N^{-r} + \left(\frac{3}{\sqrt{2\pi}} + \frac{1}{2\pi}\sqrt{\frac{\ln 36}{36}} \right) \|\mathbf{f}\|_\infty N^{-r} + 2\|\boldsymbol{\mu}\|_\infty \\
&\quad \leq \left(\frac{1}{6\sqrt{\ln 36}} + \frac{3}{\sqrt{2\pi}} + \frac{1}{2\pi}\sqrt{\frac{\ln 36}{36}} \right) \frac{\|\mathbf{f}\|_\infty}{N^r} + 2\|\boldsymbol{\mu}\|_\infty < 2\left(\frac{\|\mathbf{f}\|_\infty}{N^r} + \|\boldsymbol{\mu}\|_\infty \right).
\end{aligned}$$

Replacing all references of $3\|\mathbf{f}\|_\infty N^{-r}$ by $2(\|\mathbf{f}\|_\infty N^{-r} + \|\boldsymbol{\mu}\|_\infty)$ in the remainder of the steps up to proving [27, Theorem 5] gives the desired noise robustness (with a slightly improved constant).

Using the revised error estimates of the nonequispaced algorithm from Theorem 1 and redefining $\delta = 3(\|\hat{\mathbf{f}} - \hat{\mathbf{f}}_s^{\mathrm{opt}}\|_1/2s + 2(\|\mathbf{f}\|_\infty N^{-r} + \|\boldsymbol{\mu}\|_\infty))$ as in the proof of [27, Theorem 5] (which also contains the proof of property (2)), the discretization algorithm [27, Algorithm 1] will produce candidate Fourier coefficient approximations in lines 9 and 12 corresponding to every $|\hat{f}_\omega| \geq (4 + 2\sqrt{2})\delta$ in place of 4δ in Theorem 1. The exact same argument as in the proof of Theorem 1 then applies to the selection of the $2s$-largest entries of this approximation with the revised threshold values and error bounds to give properties (1) and (3).

In detail, [27, Lemma 13] and the discussion right after its statement gives that property (2) holds for any approximate coefficient with frequency recovered throughout the algorithm (which, for the purposes of the following discussion, we will store in $\mathbf{x}$ rather than $\hat{R}$ defined in [27, Algorithm 1]), not just those in the final output $\mathbf{v} := \mathbf{x}_s^{\mathrm{opt}}$. Additionally, by the same lemma and our revised bounds from Theorem 1, any frequency $\omega \in [N]$ satisfying $|f_\omega| > (4 + 2\sqrt{2})\delta$ will have an associated coefficient estimate in $\mathbf{x}$.

By Lemma 4, $|\mathscr{S}_{(4+2\sqrt{2})\delta}| \leq 2s = |\operatorname{supp}(\mathbf{v})|$, and so if $\omega \in \mathscr{S}_{(4+2\sqrt{2})\delta} \setminus \operatorname{supp}(\mathbf{v})$, there exists some $\omega' \in \operatorname{supp}(\mathbf{v}) \setminus \mathscr{S}_{(4+2\sqrt{2})\delta}$ such that $v_{\omega'}$ took the place of v_ω in $\mathscr{S}$. In particular, this means that $|x_{\omega'}| \geq |x_\omega|$, $|\hat{f}_{\omega'}| \leq (4 + 2\sqrt{2})\delta$, and $|\hat{f}_\omega| > (4 + 2\sqrt{2}\delta)$. Thus,

$$(4 + 2\sqrt{2})\delta + \sqrt{2}\delta > |\hat{f}_{\omega'}| + \sqrt{2}\delta \geq |x_{\omega'}| \geq |x_\omega| \geq |\hat{f}_\omega| - \sqrt{2}\delta,$$

implying that $|\hat{f}_\omega| \leq 4(1 + \sqrt{2})\delta$ and therefore proving (1).

Finally, to prove (3), we use Lemma 4, and consider

$$\begin{aligned}
\|\hat{\mathbf{f}} - \mathbf{v}\|_2 &\leq \|\hat{\mathbf{f}} - \hat{\mathbf{f}}|_{\operatorname{supp}(\mathbf{v})}\|_2 - \|(\hat{\mathbf{f}} - \mathbf{v})|_{\operatorname{supp}(v)}\|_2 \\
&\leq \|\hat{\mathbf{f}} - \hat{\mathbf{f}}|_{\mathscr{S}_{(4+2\sqrt{2})\delta} \cap \operatorname{supp}(\mathbf{v})}\|_2 + \sqrt{2}\delta\sqrt{2s}
\end{aligned}$$

$$
\begin{aligned}
&\leq \|\hat{\mathbf{f}} - \hat{\mathbf{f}}|_{\mathscr{S}_{(4+2\sqrt{2})\delta}}\|_2 + \|\hat{\mathbf{f}}|_{\mathscr{S}_{(4+2\sqrt{2})\delta} \setminus \mathrm{supp}(\mathbf{v})}\|_2 + 2\delta\sqrt{s} \\
&\leq \|\hat{\mathbf{f}} - \hat{\mathbf{f}}_{2s}^{\mathrm{opt}}\|_2 + (4+2\sqrt{2})\delta\sqrt{2s} + 4(1+\sqrt{2})\delta\sqrt{2s} + 2\delta\sqrt{s} \\
&\leq \|\hat{\mathbf{f}} - \hat{\mathbf{f}}_{2s}^{\mathrm{opt}}\|_2 + (14+8\sqrt{2})\delta\sqrt{s}
\end{aligned}
$$

which finishes the proof. □

References

1. Bittens, S., Plonka, G.: Real sparse fast DCT for vectors with short support. Linear Algebra Appl. **582**, 359–390 (2019). https://doi.org/10.1016/j.laa.2019.08.006
2. Bittens, S., Plonka, G.: Sparse fast DCT for vectors with one-block support. Numer. Algorithms **82**(2), 663–697 (2019). https://doi.org/10.1007/s11075-018-0620-1
3. Choi, B., Christlieb, A., Wang, Y.: Multiscale High-Dimensional Sparse Fourier Algorithms for Noisy Data. ArXiv e-prints (2019). arXiv:1907.03692
4. Choi, B., Christlieb, A., Wang, Y.: High-dimensional sparse Fourier algorithms. Numer. Algorithms (2020). https://doi.org/10.1007/s11075-020-00962-1
5. Choi, B., Iwen, M.A., Krahmer, F.: Sparse harmonic transforms: a new class of sublinear-time algorithms for learning functions of many variables. Found. Comput. Math. (2020). https://doi.org/10.1007/s10208-020-09462-z
6. Choi, B., Iwen, M.A., Volkmer, T.: Sparse harmonic transforms II: best s-term approximation guarantees for bounded orthonormal product bases in sublinear-time. ArXiv e-prints (2020). arXiv:1909.09564
7. Christlieb, A., Lawlor, D., Wang, Y.: A multiscale sub-linear time Fourier algorithm for noisy data. Appl. Comput. Harmon. Anal. **40**(3), 553–574 (2016). https://doi.org/10.1016/j.acha.2015.04.002
8. Cohen, A., Dahmen, W., DeVore, R.: Compressed sensing and best k-term approximation. J. Am. Math. Soc. **22**(1), 211–231 (2009). https://doi.org/10.1090/S0894-0347-08-00610-3
9. Foucart, S., Rauhut, H.: A Mathematical Introduction to Compressive Sensing. Springer, Berlin (2013). https://doi.org/10.1007/978-0-8176-4948-7
10. Gilbert, A.C., Indyk, P., Iwen, M., Schmidt, L.: Recent developments in the sparse Fourier transform: a compressed Fourier transform for big data. IEEE Signal Process. Mag. **31**(5), 91–100 (2014). https://doi.org/10.1109/MSP.2014.2329131
11. Gilbert, A.C., Muthukrishnan, S., Strauss, M.: Improved time bounds for near-optimal sparse Fourier representations. In: Papadakis, M., Laine, A.F., Unser, M.A. (eds.) Wavelets XI, vol. 5914, pp. 398 – 412. International Society for Optics and Photonics. SPIE (2005). https://doi.org/10.1117/12.615931
12. Gilbert, A.C., Strauss, M.J., Tropp, J.A.: A tutorial on fast Fourier sampling. IEEE Signal Process. Mag. **25**(2), 57–66 (2008). https://doi.org/10.1109/MSP.2007.915000
13. Gross, C., Iwen, M.A., Kämmerer, L., Volkmer, T.: A deterministic algorithm for constructing multiple rank-1 lattices of near-optimal size. ArXiv e-prints (2020). arXiv:2003.09753
14. Hassanieh, H., Indyk, P., Katabi, D., Price, E.: Simple and practical algorithm for sparse Fourier transform. In: Proceedings of the twenty-third annual ACM-SIAM symposium on discrete algorithms, pp. 1183–1194. ACM, New York (2012). https://doi.org/10.1137/1.9781611973099.93
15. Iwen, M.A.: Combinatorial sublinear-time Fourier algorithms. Found. Comput. Math. **10**(3), 303–338 (2010). https://doi.org/10.1007/s10208-009-9057-1
16. Iwen, M.A.: Improved approximation guarantees for sublinear-time Fourier algorithms. Appl. Comput. Harmon. Anal. **34**, 57–82 (2013). https://doi.org/10.1016/j.acha.2012.03.007
17. Kämmerer, L.: High Dimensional Fast Fourier Transform Based on Rank-1 Lattice Sampling. Dissertation. Universitätsverlag Chemnitz (2014)
18. Kämmerer, L.: Reconstructing multivariate trigonometric polynomials from samples along rank-1 lattices. In: Fasshauer, G.E., Schumaker, L.L. (eds.) Approximation Theory XIV: San Antonio 2013, pp. 255–271. Springer International Publishing, Berlin (2014). https://doi.org/10.1007/978-3-319-06404-8_14
19. Kämmerer, L., Krahmer, F., Volkmer, T.: A sample efficient sparse FFT for arbitrary frequency candidate sets in high dimensions. ArXiv e-prints (2020). arXiv:2006.13053

20. Kämmerer, L., Potts, D., Volkmer, T.: Approximation of multivariate periodic functions by trigonometric polynomials based on rank-1 lattice sampling. J. Complex. **31**(4), 543–576 (2015). https://doi.org/10.1016/j.jco.2015.02.004
21. Kämmerer, L., Potts, D., Volkmer, T.: High-dimensional sparse FFT based on sampling along multiple rank-1 lattices. Appl. Comput. Harmon. Anal. **51**, 225–257 (2021). https://doi.org/10.1016/j.acha.2020.11.002
22. Kapralov, M.: Sparse Fourier transform in any constant dimension with nearly-optimal sample complexity in sublinear time, pp. 264–277. Assoc. Comput. Mach., New York (2016). https://doi.org/10.1145/2897518.2897650
23. Kuo, F.Y., Migliorati, G., Nobile, F., Nuyens, D.: Function integration, reconstruction and approximation using rank-1 lattices. ArXiv e-prints (2020). arXiv:1908.01178
24. Kuo, F.Y., Sloan, I.H., Woźniakowski, H.: Lattice rules for multivariate approximation in the worst case setting. In: Monte Carlo and quasi-Monte Carlo methods 2004, pp. 289–330. Springer, Berlin (2006). https://doi.org/10.1007/3-540-31186-6_18
25. Lawlor, D., Wang, Y., Christlieb, A.: Adaptive sub-linear time Fourier algorithms. Adv. Adapt. Data Anal. **05**(01), 1350003 (2013). https://doi.org/10.1142/S1793536913500039
26. Li, D., Hickernell, F.J.: Trigonometric spectral collocation methods on lattices. In: Cheng, S.Y., Shu, C.-W., Tang T. (eds.) Recent advances in scientific computing and partial differential equations (Hong Kong, 2002), Contemp. Math., vol. 330, pp. 121–132. Amer. Math. Soc., Providence (2003). https://doi.org/10.1090/conm/330/05887
27. Merhi, S., Zhang, R., Iwen, M.A., Christlieb, A.: A new class of fully discrete sparse Fourier transforms: Faster stable implementations with guarantees. J. Fourier Anal. Appl. **25**(3), 751–784 (2019). https://doi.org/10.1007/s00041-018-9616-4
28. Morotti, L.: Explicit universal sampling sets in finite vector spaces. Appl. Comput. Harmon. Anal. **43**(2), 354–369 (2017). https://doi.org/10.1016/j.acha.2016.06.001
29. Munthe-Kaas, H., Sørevik, T.: Multidimensional pseudo-spectral methods on lattice grids. Appl. Numer. Math. **62**(3), 155–165 (2012). https://doi.org/10.1016/j.apnum.2011.11.002
30. Plonka, G., Potts, D., Steidl, G., Tasche, M.: Numerical Fourier Analysis. Springer, Berlin (2018). https://doi.org/10.1007/978-3-030-04306-3
31. Plonka, G., Wannenwetsch, K.: A sparse fast Fourier algorithm for real non-negative vectors. J. Comput. Appl. Math. **321**, 532–539 (2017). https://doi.org/10.1016/j.cam.2017.03.019
32. Plonka, G., Wannenwetsch, K., Cuyt, A., Lee, Ws.: Deterministic sparse FFT for M-sparse vectors. Numer. Algorithms **78**(1), 133–159 (2018). https://doi.org/10.1007/s11075-017-0370-5
33. Potts, D., Volkmer, T.: Sparse high-dimensional FFT based on rank-1 lattice sampling. Appl. Comput. Harmon. Anal. **41**(3), 713–748 (2016). https://doi.org/10.1016/j.acha.2015.05.002
34. Segal, B., Iwen, M.: Improved sparse Fourier approximation results: faster implementations and stronger guarantees. Numer. Algorithms **63**(2), 239–263 (2013). https://doi.org/10.1007/s11075-012-9621-7
35. Temlyakov, V.N.: Reconstruction of periodic functions of several variables from the values at the nodes of number-theoretic nets. Anal. Math. **12**(4), 287–305 (1986). https://doi.org/10.1007/BF01909367
36. Temlyakov, V.N.: Approximation of Periodic Functions. Computational Mathematics Analysis Series. Nova Sci. Publ., Inc., Commack (1993)
37. Volkmer, T.: Multivariate Approximation and High-Dimensional Sparse FFT Based on Rank-1 Lattice Sampling. Dissertation. Universitätsverlag Chemnitz (2017)