ORIGINAL PAPER

A sample efficient sparse FFT for arbitrary frequency candidate sets in high dimensions

Lutz Kämmerer[1] · Felix Krahmer[2] · Toni Volkmer[1]

Abstract

In this paper, a sublinear time algorithm is presented for the reconstruction of functions that can be represented by just few out of a potentially large candidate set of Fourier basis functions in high spatial dimensions, a so-called high-dimensional sparse fast Fourier transform. In contrast to many other such algorithms, our method works for arbitrary candidate sets and does not make additional structural assumptions on the candidate set. Our transform significantly improves upon the other approaches available for such a general framework in terms of the scaling of the sample complexity. Our algorithm is based on sampling the function along multiple rank-1 lattices with random generators. Combined with a dimension-incremental approach, our method yields a sparse Fourier transform whose computational complexity only grows mildly in the dimension and can hence be efficiently computed even in high dimensions. Our theoretical analysis establishes that any Fourier s-sparse function can be accurately reconstructed with high probability. This guarantee is complemented by several numerical tests demonstrating the high efficiency and versatile applicability for the exactly sparse case and also for the compressible case.

Keywords High-dimensional sparse fast Fourier transform · FFT · Multivariate trigonometric polynomials · Lattice rule · Multiple rank-1 lattices · Approximation of multivariate functions

Mathematics Subject Classification (2010) 65T · 65T40 · 42A10

1 Introduction

The problem of reconstructing or approximating a function from a finite number of samples is one of the central objects of study in approximation theory. In finite dimensional

✉ Lutz Kämmerer
   lutz.kaemmerer@math.tu-chemnitz.de

Extended author information available on the last page of the article.

function spaces (or function spaces that allow for good finite-dimensional approximations), this problem can be formulated as a linear system $\mathbf{Ax} = \mathbf{y}$. Here, $\mathbf{y}$ is the vector of observed function values, $\mathbf{x}$ is the (unknown) coefficient vector of the function in a given basis B, and the matrix $\mathbf{A}$ represents the linear map that maps basis coefficients to function evaluations; knowing the basis B and the sampling points, $\mathbf{A}$ can be explicitly determined. The focus of this paper will be on the case where $\mathbf{A}$ is the discrete Fourier transform (DFT) matrix or some submatrix of it.

When $\mathbf{A}$ is left-invertible with left inverse $\mathbf{L}$, as it is the case for the full DFT matrix, one can find $\mathbf{x}$ via the matrix-vector product of $\mathbf{L}$ and $\mathbf{y}$. The computational complexity of such a matrix-vector multiplication can, in general, not be improved beyond a linear scaling in the number of entries of the matrix even if the matrix inverse has been precomputed, as this is what is needed just to read the matrix. In the case of DFT matrices, further acceleration is possible via the Fast Fourier transform (FFT) [11], which improves the complexity of multiplying the (inverse) discrete Fourier matrix of size $n \times n$ with an arbitrary vector of length n from $\mathcal{O}(n^2)$ to $\mathcal{O}(n \log n)$. An extension of this result for irregular sampling nodes with similar advantages is the nonequispaced fast Fourier transform (see, e.g., [31]). Further optimization is possible if the frequencies, i.e., the multi-indices of the basis functions, belong to special structured sets such as hyperbolic crosses, cf. [14].

A natural next step is to consider functions where the frequencies of the nonzero Fourier coefficients form a relatively small set $I \subset \mathbb{Z}^d$ within a possibly huge candidate set $\Gamma \subset \mathbb{Z}^d$, but for which the set I is unknown. Such functions are commonly referred to as Fourier sparse. Here, a function is s-sparse in the given basis if it can be expressed as a linear combination of only at most s basis elements, i.e., $s \geq |I|$. Naturally, this reduced model approach is assumed in order to reduce both the number of samples required and the computation time. The crucial challenge in this approach is the identification of I, which is a much harder task compared to the classical approach where only the Fourier coefficients, i.e., the entries of coefficient vector $\mathbf{x}$, are unknown.

The focus in the area of compressive sensing has been on the sample complexity of this problem. Here, one seeks to recover a sparse signal from as few samples as possible. It has been shown that recovery is possible for a number of random samples that scales linearly in the sparsity s up to logarithmic factors [5, 34], even when the signal is sparse in an arbitrary incoherent basis [4] or a wavelet basis [28]. The resulting methods cannot be considered to be fast transforms per se, as the computation time required for most reconstruction procedures scales at least linearly in the number of candidates $|\Gamma| \gg s$.

For any sublinear time algorithm to recover Fourier sparse signals, significant subsampling compared to $|\Gamma|$ must be an integral part. For the case of $d = 1$ spatial dimension, a randomized approach that achieves runtime scaling quadratically in terms of the sparsity s (up to logarithmic factors in $|\Gamma|$) was presented in [17]. A deterministic, combinatorial algorithm with runtime complexity scaling quadratically in the sparsity s was presented in [21, 22]. Improved randomized algorithms with runtime scaling linearly in the sparsity were introduced in [1, 2, 16, 21, 22] and an improved deterministic algorithm in [29]. The latter one was enhanced once more in [10] to accomodate for samples perturbed by Gaussian noise.

Many of the aforementioned algorithms have been generalized to the multi-dimensional case $d \geq 2$, and new algorithms have been developed. In Table 1, we give an overview on existing multi-dimensional sparse FFT approaches and the ones presented in this paper. We compare sample and computational complexities as well as the general settings and types of sampling strategies.

The deterministic approach in [29] and its variant for noisy samples [10] have been extended to high spatial dimensions d in [7] and [6], respectively, with runtime complexity $\mathcal{O}(d\, s\, \log s)$ and sample complexity $\mathcal{O}(d\, s)$ on average (both with an

Table 1 Overview on multi-dimensional sparse FFT approaches with N_Γ as defined in (6)

Method	Sample complexity	Computational complexity	Comments
Choi et al. [7]	$d\, s$ (w.h.p.)	$d\, s\, \log s$ (w.h.p.)	random signal setting deterministic sampling strategy success w.h.p.
Choi et al. [6]	$d\, s\, \log N_\Gamma$ (w.h.p.)	$d\, s\, \log s\, \log N_\Gamma$ (w.h.p.)	random signal setting deterministic sampling strategy success w.h.p.
Indyk, P. and Kapralov [19]	$d^{\mathcal{O}(d)}\, s\, \log N_\Gamma$	$d^{\mathcal{O}(d)}\, s\, \log N_\Gamma$ $+ d^3\, N_\Gamma^d\, \log^3 N_\Gamma$	arbitrary signal random sampling strategy success w.h.p.
Morotti [30]	$d^3\, s^2\, N_\Gamma\, \log N_\Gamma$	$d^3\, s^2\, N_\Gamma^2\, \log N_\Gamma$	arbitrary signal deterministic sampling strategy guaranteed success
Iwen [22, Thm. 7]	$d^4\, s^2\, \log^4(N_\Gamma)$	$d^4\, s^2\, \log^4(N_\Gamma)$	arbitrary signal deterministic sampling strategy guaranteed success
Iwen [22, Cor. 4]	$d^4\, s\, \log^4(N_\Gamma)$	$d^4\, s\, \log^4(N_\Gamma)$	arbitrary signal random subsampling of deterministic strategy [22, Thm. 7] success w.h.p.
Kämmerer et al. [26]	$d\, s^2\, N_\Gamma\, \log^2 s$ (w.h.p.) $d\, s^3\, N_\Gamma\, \log^3 s$ (w.c.)	$d^2\, s^2\, N_\Gamma\, \log^3 s$ (w.h.p.) $d^2\, s^3\, N_\Gamma\, \log^4 s$ (w.c.)	arbitrary signal random sampling strategy success w.h.p.
Choi et al. [8, 9]	$d^5\, s^3\, \log^4(d\, s\, N_\Gamma)$	$d^6\, s^5\, \log^4(d\, s\, N_\Gamma)$	arbitrary signal random sampling strategy success w.h.p.
Theorem 1	$s\, \log|\Gamma|$ $\lesssim d\, s\, \log N_\Gamma$	$(s\, \log s + d\,|\Gamma|)\log|\Gamma|$ $\lesssim d^2\, N_\Gamma^d\, \log N_\Gamma$ for $s\, \log s \lesssim d\, N_\Gamma^d$	arbitrary signal random sampling strategy success w.h.p.
Section 3	$d\, s^2\, \log^2(d\, s\, N_\Gamma)$ $+ d\, s\, N_\Gamma\, \log(d\, s)$	$d^2\, s^2\, N_\Gamma\, \log^3(d\, s\, N_\Gamma)$ (w.h.p.) $d^2\, s^3\, N_\Gamma\, \log^3(d\, s\, N_\Gamma)$ (w.c.)	arbitrary signal random sampling strategy success w.h.p.

additional factor of $\log N_\Gamma$ for the variant addressing noisy samples), cf. Table 1. The method is highly efficient since the average case sample complexity is best possible, as a frequency set $I \subset \mathbb{Z}^d$ of cardinality $|I| = s$ has $d\,s$ many entries, and correspondingly, the average case runtime complexity is best possible up to a logarithmic factor. However, the approach uses a random signal setting, i.e., the expectations and success probabilities are computed with respect to active frequencies collected in I that are assumed to be distributed uniformly at random in a full (finite) tensor product grid as candidate set Γ. Especially, smaller structured subsets, such as (subsets of) hyperbolic crosses, do not fit into this model assumption. In contrast to this, the other methods mentioned in Table 1 use an "arbitrary signal" model, i.e., the active frequencies can be an arbitrary subset of a suitable candidate set Γ.

In [30], the author presents a deterministic and noise robust high-dimensional sparse FFT approach. The corresponding complexities scale merely quadratic with respect to the sparsity s. However, the computational complexity of the Fourier transform $\mathcal{O}\left(d^3\,s^2\,N_\Gamma^2\,\log N_\Gamma\right)$ and the computational complexity of the construction of the sampling set $\mathcal{O}\left(d^6\,s^2\,N_\Gamma\,\log^3 N_\Gamma\right)$ indicates that the approach has advantages only for moderate expansions N_Γ and moderate dimensions d.

The randomized approach in [1] has also been generalized to Fourier transforms in constant dimension [19], but due to dimensional scaling of $d^{\mathcal{O}(d)}$ in the runtime as well as sample complexity, cf. Table 1, the approach will not be feasible in higher spatial dimensions d.

In contrast, the deterministic approach in [22] has been shown to generalize to high spatial dimensions d with a quadratic scaling of the complexities in the sparsity s and a polynomial scaling d^4 in the spatial dimension, cf. Table 1, but may exhibit limitations with respect to its numerical stability on large candidate sets Γ, in particular in high spatial dimensions d. The reason for this issue is a transformation of the d-dimensional frequency domain Γ, which is assumed to be a tensor product grid, to a one-dimensional frequency domain $[0, \tilde{N})$, where $\tilde{N} \gtrsim |\Gamma|$ is necessary in order to obtain a unique mapping between both frequency grids. Subsequently, one applies a one-dimensional sparse FFT approach to the new one-dimensional problem for typically huge $\tilde{N}$, which suffers from numerical issues due to the fact that highly oscillating basis functions—with then possibly neighboring one-dimensional frequencies—are difficult to distinguish when using only a few samples. Remarkably, the locations of the samples and the algorithm itself are fully deterministic, and the method is guaranteed to successfully detect all active frequencies (on a machine with sufficiently high numerical precision). The related randomized version in [22] uses a random subsampling and achieves a linear scaling of the complexities in s, but also suffers from potential numerical issues for large d.

The aforementioned approaches share the main characteristic that they are specifically designed for full (finite) tensor product grids in frequency domain as candidate sets Γ. In many applications, however, the function space under consideration motivates a significantly reduced set Γ, such as a hyperbolic cross, or possibly an unstructured set Γ. So only those Fourier s-sparse functions need to be considered with active frequencies in the candidate set Γ.

High-dimensional sparse fast Fourier transforms for such scenarios have been designed, for example, using rank-1 lattices [26, 33]. These methods use a

dimension-incremental pairing technique, which can also be found in [32, 35]. We describe the general ideas behind this technique in more detail later in this section. The required runtime complexity of the algorithm in [26] is $\mathcal{O}\left(d^2 \, s^2 \, N_\Gamma \log^3 s\right)$ with high probability, where N_Γ is the expansion of Γ, cf. (6), and the sample complexity is $\mathcal{O}\left(d \, s^2 \, N_\Gamma \log^2 s\right)$, cf. Table 1. Due to the mild dependence on the spatial dimension d and the sparsity s, this approach is well suited for high-dimensional problems.

Furthermore, a much more general approach with runtime complexity $\mathcal{O}(d^6 \, s^5 \log^4(d \, s \, N_\Gamma))$ and sample complexity $\mathcal{O}\left(d^5 \, s^3 \, \log^4(d \, s \, N_\Gamma)\right)$ was presented in [8, 9], which has also been applied to more general tensor product bases. It is based on a similar dimension-incremental pairing technique as in [26, 33] and internally uses ideas from compressed sensing.

One main contribution of this paper is the design of a sample-efficient sparse fast Fourier transform for Fourier sparse functions in high spatial dimensions d with frequencies in a given arbitrary candidate set $\Gamma \subset \mathbb{Z}^d$, $|\Gamma| < \infty$, where in particular Γ may be much smaller than a full tensor product grid. The runtime complexity exhibits an additive joint dependence on s and $|\Gamma|$ up to logarithmic factors and— most importantly—a sample complexity that is linear in s times a logarithmic factor in $|\Gamma|$. We stress on the fact that the algorithm succeeds with high probability, where the failure probability does not depend on the structure of the candidate set Γ or the active frequency set I of sparsity s. In particular, the presented estimates still hold true even if the frequency set I is arbitrarily clustered. More precisely, our main result reads as follows.

Theorem 1 *(Main Theorem). Let the sparsity $s \in \mathbb{N}$ and a frequency candidate set $\Gamma \subset \mathbb{Z}^d$, $|\Gamma| \geq 8$, fulfilling*

$$\Gamma \subset \mathop{\times}_{j=1}^{d} [a_j, b_j], \qquad \max_{j=1,\ldots,d} (b_j - a_j) \leq 10.33 \, s, \tag{1}$$

and the parameter $\delta \in (0, 1)$ be given. Then, there exists a randomized sampling strategy based on random rank-1 lattices generating a set S of at most

$$37 \, s \, (\ln |\Gamma| - \ln \delta)$$

sampling locations such that the following holds.

Consider an arbitrary unknown set $I := \{k \in \Gamma \colon \hat{p}_k \neq 0\} \subset \Gamma$ of d-dimensional frequencies with $|I| \leq s$ and generate a random set S via the sampling strategy.

Then with probability $1 - \delta$ it holds that

- *all frequencies $k \in I$ that belong to the nonzero Fourier coefficients $\hat{p}_k$ and*
- *the corresponding Fourier coefficients $\hat{p}_k$ of*
- *<u>any</u> multivariate trigonometric polynomial $p(\mathbf{x}) := \sum_{k \in I} \hat{p}_k \, e^{2\pi i k \cdot \mathbf{x}}$*

can be identified from its values at the sampling locations in S.

This identification and the computation of the Fourier coefficients $\hat{p}_k$ can be realized by Algorithm 2 in less than

$$C \, (s \log s + d \, |\Gamma|) \, (\ln |\Gamma| - \ln \delta)$$

arithmetic operations with an absolute constant $C > 0$.

Remark 1 In some sense, the statement of Theorem 1 is in between a uniform and a non-uniform recovery guarantee: A support I is fixed, and the guarantee is then uniform for all functions with that support. It will be an interesting question for future research whether and how our sampling approach allows for recovery guarantees that are also uniform over different support sets.

Remark 2 In Theorem 1, we can relax the subset property (1) to

$$(I \cup \Gamma) \subset \underset{j=1}{\overset{d}{\times}} [a_j, b_j], \qquad \max_{j=1,\dots,d} (b_j - a_j) \leq 10.33 \, |I|,$$

with the consequence that only the frequencies $\mathbf{k} \in \Gamma \cap I$ can be identified. Moreover, the constant 10.33 in (1) can be adapted for specific applications, see also Remark 7.

In addition to the significant improvement for arbitrary candidate sets, Theorem 1 also yields significant acceleration for sparse Fourier transforms on a full tensor grid, the setup studied in many of the works discussed above. Namely, in high spatial dimensions, our result can be combined with a so-called dimension-incremental approach, which is also a key ingredient of many of the currently best known algorithms for sparse fast transforms. The idea of such approaches is to identify the multi-indices corresponding to the frequencies of the non-zero Fourier coefficients component by component, where in each step one works with the candidate set Γ consisting of all indices that agree with the previously identified partial component vectors, see Section 3 for more details.

To identify the components of the active multi-indices, we will use Theorem 1, which is based on random rank-1 lattices. In that sense our work follows a similar strategy as in [26] (and its predecessor [33]), which also uses rank-1 lattices in the identification step of a dimension-incremental approach. The sparse transforms designed in these works, however, exhibit a larger runtime complexity due to a suboptimal scaling of the sample complexity required in each dimension-incremental step. More precisely, they are based on spatial discretizations of full frequency candidate sets without taking advantage of the knowledge that only a few frequencies within these candidate sets are active. With Algorithm 1 below at hand, in contrast, one can explicitly take this additional information into account to obtain a constructive sampling strategy with an improved sample complexity in combination with a comparable runtime complexity as in [26]. This is summarized in the following Theorem.

Theorem 2 *(Dimension-incremental strategy). Let the sparsity $s \in \mathbb{N}$, a frequency candidate set $\Gamma \subset \mathbb{Z}^d$, $|\Gamma| < \infty$, and the parameter $\delta \in (0, 1)$ be given. Moreover, we define N_Γ as in (6), e.g., $\Gamma \subset [\lceil -N_\Gamma/2 \rceil, \lfloor N_\Gamma/2 \rfloor]^d \cap \mathbb{Z}^d$. Then, there exists a randomized sampling strategy based on random rank-1 lattices generating a set S of sampling locations with cardinality*

$$|S| \in \mathcal{O}\left(d \, s \max(s, N_\Gamma) \log^2 \frac{d \, s \, N_\Gamma}{\delta} \right)$$

such that the following holds.

Consider an arbitrary multivariate trigonometric polynomial $p(\mathbf{x}) := \sum_{\mathbf{k} \in I} \hat{p}_{\mathbf{k}} \, e^{2\pi i \mathbf{k} \cdot \mathbf{x}}$, $I \subset \Gamma$, $|I| \leq s$, *where we assume* $\min_{\mathbf{k} \in I} |\hat{p}_{\mathbf{k}}| > 0$, *and generate a random set S via the sampling strategy. Then with probability* $1 - \delta$ *it holds that*

- *all frequencies* $\mathbf{k} \in I$ *as well as*
- *all Fourier coefficients* $\hat{p}_{\mathbf{k}}, \mathbf{k} \in I$,

of the multivariate trigonometric polynomial p can be reconstructed from its values at the sampling locations in S.

This identification of the frequencies and the computation of the Fourier coefficients can be realized by a combination of Algorithms 3 *and* 4. *The suggested method has a computational complexity of*

$$\mathcal{O}\left(d^2 \, s^2 \, N_\Gamma \, \log^3 \frac{d \, s \, N_\Gamma}{\delta} \right)$$

with probability at least $1 - \delta$ *as well as* $\mathcal{O}\left(d^2 \, s^3 \, N_\Gamma \, \log^3 \frac{d \, s \, N_\Gamma}{\delta} \right)$ *in the worst case.*

Remark 3 At this point, we stress on the fact that Theorem 2 is based on a dimension-incremental approach, where it is still an open question whether or not the complexities can scale better than s^2. The crucial reason for this behavior in our result is the computation of projections that may cause cancellations and the resulting necessity of a number of repetitions of these projections, which scale approximately linear in s in theory. An in-depth discussion on at least one point of view of that issue can be found in [26, Section 4.2].

Remark 4 Please note the differences in Theorems 1 and 2. In particular, the statement of Theorem 2 is a non-uniform recovery guarantee, due to the applied dimension-incremental approach in our method. It will be a second interesting question for future research whether and how our dimension-incremental sampling approach proves to allow for recovery guarantees that are also uniform—provided that one uses a method guaranteeing a uniform recovery in each dimension-incremental "pairing step," cf. Section 3.

Remark 5 There are two probabilities of failure in Theorem 2 that correspond to different types of mistakes: While the increased computational complexity is directly linked to a large number of false positives, the reconstruction can only fail if there are false negatives, see also Remark 9.

To compare this runtime and sample complexity to those of alternative dimension-incremental strategies, we again refer to Table 1. In addition to the approach [26] that is based on rank-1 lattices, this includes [8, 9] which use compressed sensing techniques for the identification step. These compressed sensing based approaches use the special structure of the candidate set Γ: When restricting to each entry of

the index set, one just encounters a regular grid in a much lower dimensional space, so one can efficiently apply compressed sensing techniques. The entries are then combined using an appropriate pairing technique, also based on compressed sensing ideas. This approach is much more general in that it allows for much larger classes of bounded orthonormal systems than just Fourier, but for that reason, it also does not take optimal advantage of the structural properties of the Fourier transform and the possible computational speedup by FFT techniques that our approach is able to exploit. This explains the inferior dimensional scaling of the runtime complexity.

Besides these works, dimension-incremental methods have also been studied in combination with Prony's method [32]. Also, there are further Prony-like techniques available in higher dimensions, cf. [13]. There is no explicit analysis of runtime and sample complexity available, however, which is why we cannot include this approach in Table 1. In addition, the feasibility of Prony's method in this context is severely limited to low sparsities due to stability problems arising even for sparsity levels in the order of a few hundreds.

2 Reconstructing multivariate trigonometric polynomials from samples along sets of random rank-1 lattices

2.1 Sparse FFT via rank-1 lattices—background and previous works

The frequency identification method we present in this work is based on a class of well established cubature methods in higher dimensions, so-called rank-1 lattice rules, which are special quasi Monte Carlo methods. Such methods consider rank-1 lattices as sampling sets, that is, sets of the form

$$\Lambda(\mathbf{z}, M) := \{j\mathbf{z}/M \bmod \mathbf{1} \colon j = 0, \ldots, M - 1\} \subset \mathbb{T}^d, \tag{2}$$

where $\mathbf{z} \in \mathbb{Z}^d$ and $M \in \mathbb{N}$ are called generating vector and lattice size of $\Lambda(\mathbf{z}, M)$, respectively. A candidate for an approximate integral of a function over a high-dimensional cube is then computed as the average of its samples along this lattice.

Naturally, it depends on the function class under consideration whether and to which accuracy this candidate approximates the true integral. In this paper, we will apply this approach for the identification of a sparse trigonometric polynomial

$$p(\mathbf{x}) := \sum_{\mathbf{k} \in I} \hat{p}_{\mathbf{k}} \, e^{2\pi i \mathbf{k} \cdot \mathbf{x}}, \tag{3}$$

where $\mathbf{k} \cdot \mathbf{x} := \sum_{j=1}^d k_j x_j$ denotes the usual inner product in $\mathbb{R}^d$. Recall that we are interested in the case that the multivariate trigonometric polynomial is sparse, i.e., the index set $I \subset \mathbb{Z}^d$ – the *frequency set* – not only has finite cardinality, $|I| < \infty$, but is also small.

Note that the *Fourier coefficient* $\hat{p}_{\mathbf{k}}$ with frequency $\mathbf{k}$ can be computed by evaluating the integral

$$\hat{p}_{\mathbf{k}} := \int_{\mathbf{x} \in \mathbb{T}^d} p(\mathbf{x}) \, e^{-2\pi i \mathbf{k} \cdot \mathbf{x}} \mathrm{d}\mathbf{x}. \tag{4}$$

As it turns out, a rank-1 lattice rule applied to the integrand in (4) often returns the correct Fourier coefficient. Indeed, the rank-1 lattice rule with rank-1 lattice $\Lambda(\mathbf{z}, M)$ as in (2) yields

$$
\begin{aligned}
\hat{p}_{\mathbf{k}}^{\Lambda(\mathbf{z},M)} &:= \frac{1}{M} \sum_{j=0}^{M-1} p\left(\frac{j}{M}\mathbf{z}\right) e^{-2\pi i \frac{j}{M} \mathbf{k}\cdot\mathbf{z}} = \sum_{\mathbf{h}\in I} \hat{p}_{\mathbf{h}} \frac{1}{M} \sum_{j=0}^{M-1} e^{2\pi i \frac{j}{M}(\mathbf{h}-\mathbf{k})\cdot\mathbf{z}} \\
&= \sum_{\mathbf{h}\in I} \hat{p}_{\mathbf{h}}\, \delta_0((\mathbf{h}-\mathbf{k})\cdot\mathbf{z} \bmod M) \\
&= \sum_{\mathbf{h}\in(\Lambda^\perp(\mathbf{z},M))\cap(I-\mathbf{k})} \hat{p}_{\mathbf{h}+\mathbf{k}},
\end{aligned}
\tag{5}
$$

where $\delta_0(0) := 1$, $\delta_0(l) := 0$ for $l \neq 0$, and $\Lambda^\perp(\mathbf{z}, M) := \{\mathbf{h} \in \mathbb{Z}^d : \mathbf{h} \cdot \mathbf{z} \equiv 0 \,(\mathrm{mod}\, M)\}$ is the integer dual lattice of the rank-1 lattice $\Lambda(\mathbf{z}, M)$. As we will see, for a random lattice Λ, $\Lambda^\perp(\mathbf{z}, M)$ will typically intersect $I - \mathbf{k}$ only in $\mathbf{h} = \mathbf{0}$ for $\mathbf{k} \in I$ or not at all for $\mathbf{k} \notin I$, so (5) typically consists just of the single summand $\hat{p}_{\mathbf{k}}$ or even none, as desired.

When the intersection consists of more elements than just $\mathbf{0}$, we speak of *aliasing*. For a specific frequency, as observed recently by one of the authors in [25], this happens only with small probability for random lattices even with small lattice sizes $M \asymp |I|$. Consequently, one has unique reconstruction of at least some of the original Fourier coefficients $\hat{p}_{\mathbf{k}}$ with a certain probability. Nevertheless, each realization of such a random lattice will yield aliasing effects for some fraction of its Fourier coefficient. This means that our goal of correctly determining for all candidate frequencies $\mathbf{h} \in \Gamma$ whether they are active or not is typically impossible with a single realization. Even to just identify a superset of the active frequencies (with some false-positives allowed) requires a lattice of larger cardinality. More precisely, the samples along a (single) rank-1 lattice rule do not allow for the reconstruction of all original Fourier coefficients $\{\hat{p}_{\mathbf{k}}\}_{\mathbf{k}\in I}$ unless the rank-1 lattice size M is on the order of $M \gtrsim |I|^2$ in the worst case [23]. In addition to this disadvantage, a corresponding generating vector $\mathbf{z}$ needs to be constructed, e.g. using a component-by-component approach, which can lead to large computational costs, cf. [26, Page 3] for a detailed discussion on this topic.

That is why in this paper, we work with multiple rank-1 lattices. Building upon the work of [25], we develop a strategy to identify the frequencies belonging to the non-zero Fourier coefficients of the polynomial p within a frequency candidate set $\Gamma \supset I$ in Section 2.2. To this end, we present some notation and technical basics from [25] in the following.

As we use a random approach to generate the multiple rank-1 lattices, our method will not be guaranteed to be successful in all cases but with a chosen high probability. Naturally, the success rate will depend on the candidate set Γ to some extent. In particular, a candidate set Γ that is compact and of small size yields superior performance as compared to one which is extremely large or very wide spread. To describe

the nature of the frequency candidate set Γ, we will work with its cardinality $|\Gamma|$ and its expansion

$$N_\Gamma := \max_{j=1,\ldots,d} \left\{ \max_{\mathbf{k} \in \Gamma} k_j - \min_{\mathbf{l} \in \Gamma} l_j \right\}. \tag{6}$$

Here, N_Γ is the smallest edge length of a cube containing the frequency candidate set Γ. That is, there exists some $\mathbf{k} \in \mathbb{Z}^d$ such that $\Gamma \subset \{\mathbf{k} + \mathbf{h} \in \mathbb{Z}^d : \mathbf{h} \in [0, N_\Gamma]^d\}$.

If $\mathbf{k}, \mathbf{k}' \in \Gamma$ agree up to multiples of M, $\mathbf{k} \equiv \mathbf{k}' \bmod M$, sampling on a rank-1 lattice of the type $\Lambda(\mathbf{z}, M)$ cannot distinguish $\mathbf{k}$ and $\mathbf{k}'$. In general, one can at best identify the equivalence class mod M. For that reason, it is often beneficial to represent each of these equivalence classes by an element in $\{0, \ldots, M-1\}$, i.e., to consider

$$\Gamma_{\mathrm{mod}\ M} := \{\mathbf{h_k} := \mathbf{k} - M\lfloor \tfrac{\mathbf{k}}{M} \rfloor : \mathbf{k} \in \Gamma\},$$

where we assume $M \in \mathbb{N}$, and to aim to identify $I_{\mathrm{mod}\ M} \cap \Gamma_{\mathrm{mod}\ M}$ using the techniques introduced in this paper. Under the assumption that $|I_{\mathrm{mod}\ M} \cap \Gamma_{\mathrm{mod}\ M}| = |I \cap \Gamma|$ (that will always hold for M large enough), this also allows for the identification of $|I \cap \Gamma|$. We refer the reader to [25, Lemma 2.3] for further details. Working with the modified candidate set is preferred as for $M \leq N_\Gamma$, dealing with $\Gamma_{\mathrm{mod}\ M}$ instead of Γ leads to a smaller expansion of the candidate set Γ under consideration and allows – in specific situations – for a significantly reduced number of sampling values required for the suggested approach, Algorithm 1 below, to identify the frequency support I of the multivariate trigonometric polynomial p.

Furthermore, for technical reasons, it will be beneficial to use rank-1 lattice sizes M that are prime numbers in addition to ensuring that $|\Gamma_{\mathrm{mod}\ M}| = |\Gamma|$, i.e., we consider

$$M \in P^\Gamma := \{M' \in \mathbb{N} : M' \text{ prime with } |\Gamma_{\mathrm{mod}\ M'}| = |\Gamma|\}.$$

This set contains at least all primes greater than N_Γ.

At this point, we stress on the fact that we will exploit the advantageous aliasing formula (5) in order to construct our algorithm. In particular, the strategy is to randomly choose multiple suitable rank-1 lattices $\Lambda(\mathbf{z}_1, M_1), \ldots, \Lambda(\mathbf{z}_L, M_L)$ and to exploit the fact that for a fixed frequency problematic aliasing effects occur with a fixed probability for each of the rank-1 lattices under consideration. In order to control the probability that these aliasing effects occur in a reasonably small proportion of the L rank-1 lattices, we need to choose L sufficiently large. We call such a collection

$$\Lambda = \Lambda(\mathbf{z}_1, M_1, \ldots, \mathbf{z}_L, M_L) := \bigcup_{\ell=1}^{L} \Lambda(\mathbf{z}_\ell, M_\ell)$$

of rank-1 lattices $\Lambda(\mathbf{z}_1, M_1), \ldots, \Lambda(\mathbf{z}_L, M_L)$ a multiple rank-1 lattice configuration.

The separate consideration of the aliasing effects of each single rank-1 lattice $\Lambda(\mathbf{z}_\ell, M_\ell)$, $\ell = 1, \ldots, L$, allows for the separate computation of possibly active $\hat{p}_\mathbf{k}^{\Lambda(\mathbf{z},M)}$, $\mathbf{k} \in \Gamma$. The structure of a specific rank-1 lattice $\Lambda(\mathbf{z}, M)$ will then provide an efficient algorithm that computes all (aliased) Fourier coefficients $\hat{p}_\mathbf{k}^{\Lambda(\mathbf{z},M)}$, $\mathbf{k} \in \Gamma$, at once requiring a computational complexity of $\mathcal{O}(M \log M + d|\Gamma|)$. This is highly efficient compared to applications of matrix vector products, cf. [23].

2.2 New approach and reconstruction guarantee

Given the role of the candidate set Γ, it should not come as a surprise that one can only exactly identify the frequency support I of a multivariate trigonometric polynomial p provided that $I \subset \Gamma$. In practical scenarios, however, errors cannot always be avoided. For example when applying the methods developed in this section as frequency identification steps in a dimension-incremental approach, cf. Section 3 below, the candidate sets arise from previous estimation steps. So it may happen that this assumption is violated at some point. For that reason we do not explicitly require $I \subset \Gamma$ and show that the method will identify the frequency support $I \cap \Gamma$ of p within Γ with high probability for a suitable choice of parameters – even when other frequencies are present. Note, however, that these parameters depend on $I \cup \Gamma$, so they are not straight forward to determine for $I \not\subset \Gamma$ and unknown I.

The main tools for our analysis are the following generalizations of [25, Lemma 3.1 and Theorem 3.2].

Lemma 1 *Let $I \subset \mathbb{Z}^d$ and $\Gamma \subset \mathbb{Z}^d$ be frequency sets of finite cardinalities, $|I| < \infty$ and $|\Gamma| < \infty$. We fix a frequency $\mathbf{k} \in I \cup \Gamma$ and choose a prime number M such that $|(I \cup \Gamma)_{\bmod M}| = |(I \cup \Gamma)|$. In addition, we choose a generating vector $\mathbf{z} \in [0, M-1]^d \cap \mathbb{Z}^d$ uniformly at random. Then, with probability not greater than $\frac{|I|}{M}$, the frequency $\mathbf{k}$ aliases to at least one frequency from $I \setminus \{\mathbf{k}\}$, i.e.,*

$$\mathbb{P}\left(\hat{p}_{\mathbf{k}}^{\Lambda(\mathbf{z},M)} \neq \hat{p}_{\mathbf{k}}\right) \leq \frac{|I|}{M},$$

where $\hat{p}_{\mathbf{k}} = 0$ for all $\mathbf{k} \in \Gamma \setminus I$, cf. (4).

Proof We start with the case $\mathbf{k} \in \Gamma \setminus I$, and we build the frequency set $\tilde{I} = I \cup \{\mathbf{k}\}$. Since we have $|(I \cup \Gamma)_{\bmod M}| = |I \cup \Gamma|$ and $\tilde{I} \subset (I \cup \Gamma)$, it follows that $|\tilde{I}_{\bmod M}| = |\tilde{I}|$, and we apply [25, Lemma 3.1] with the frequency set $\tilde{I}$, which yields that $\mathbf{k}$ aliases to at least one frequency from $\tilde{I} \setminus \{\mathbf{k}\} = I$ with probability at most $\frac{|\tilde{I}|-1}{M} = \frac{|I|}{M}$.

In the case $\mathbf{k} \in I$, we have $|I_{\bmod M}| = |I|$ since $|(I \cup \Gamma)_{\bmod M}| = |I \cup \Gamma|$. Applying [25, Lemma 3.1] yields that $\mathbf{k}$ aliases to at least one other frequency from $I \setminus \{\mathbf{k}\}$ with probability at most $\frac{|I|-1}{M} < \frac{|I|}{M}$. □

Theorem 3 *Consider a frequency set $I \subset \mathbb{Z}^d$ and a set $\Gamma \subset \mathbb{Z}^d$ of frequency candidates, $|I| < \infty$ and $|\Gamma| < \infty$. In addition, we fix $\delta \in (0, 1)$ and $\nu \in (0, 1/2)$. Moreover, we determine the numbers*

$$\lambda \geq c\,|I| \qquad \text{with } c > \frac{1}{\nu},$$

$$L \geq \left\lceil \frac{c(c-2)}{(c\nu-1)^2 \ln(c-1)} \left(\ln|\Gamma| - \ln\delta\right) \right\rceil.$$

We choose L not necessarily distinct lattice sizes $M_\ell \in P^{I \cup \Gamma}$, $M_\ell > \lambda$. For each M_ℓ, we choose a generating vector $\mathbf{z}_\ell \in [0, M_\ell - 1]^d \cap \mathbb{Z}^d$ uniformly at random. Then,

the probability that the frequency $\boldsymbol{k} \in I \cup \Gamma$ aliases to a frequency from $I \setminus \{\boldsymbol{k}\}$ for
at least $\lceil \nu L \rceil \in [1, L]$ many of the rank-1 lattices $\Lambda(\mathbf{z}_\ell, M_\ell)$ is less than $\delta / |\Gamma|$, i.e.,

$$\mathbb{P}\left(\left|\left\{\ell \in \{1, \ldots, L\} \colon \hat{p}_{\boldsymbol{k}}^{\Lambda(\mathbf{z}_\ell, M_\ell)} \neq \hat{p}_{\boldsymbol{k}}\right\}\right| \geq \nu L\right) < \frac{\delta}{|\Gamma|}.$$

Proof For the fixed frequency $\mathbf{k} \in I \cup \Gamma$, we define the random variables

$$Y_\ell^{\mathbf{k}} := \begin{cases} 0 & : \ \mathbf{k} \text{ does not alias to a frequency from } I \setminus \{\mathbf{k}\} \text{ using } \Lambda(\mathbf{z}_\ell, M_\ell), \\ 1 & : \ \mathbf{k} \text{ aliases to at least one frequency from } I \setminus \{\mathbf{k}\} \text{ using } \Lambda(\mathbf{z}_\ell, M_\ell). \end{cases}$$

$$(7)$$

We distinguish two different cases. First, we consider $\mathbf{k} \in I$ for $|I| = 1$. Then,
we achieve $Y_\ell^{\mathbf{k}} = 0$ for each $\ell \in \{1, \ldots, L\}$ and $\mathbb{P}\left\{\sum_{\ell=1}^{L} Y_\ell^{\mathbf{k}} \geq \nu L\right\} = 0 < \frac{\delta}{|\Gamma|}$.
Second, we consider the cases $\mathbf{k} \in I$ in conjunction with $|I| > 1$ or $\mathbf{k} \in \Gamma \setminus I$. The
random variables $Y_\ell^{\mathbf{k}}, \ell = 1, \ldots, L$, are independent with uniformly bounded mean
$\mu_\ell, 0 < \mu_\ell \leq \frac{|I|}{M_\ell} < \frac{|I|}{\lambda} \leq \frac{1}{c} < \nu$, where the upper bounds hold due to Lemma 1. The
strict inequality $0 < \mu_\ell$ holds since we obtain the equality $Y_\ell^{\mathbf{k}} = 1$ for the admissible
choice $\mathbf{z}_\ell = (0, \ldots, 0)^\top$, which then implies $\mathbb{P}\left\{Y_\ell^{\mathbf{k}} = 1\right\} > 0$.

Using Hoeffding's inequality, cf. [18, Theorem 1], and $\mu := L^{-1} \sum_{\ell=1}^{L} \mu_\ell < \nu \leq$
$1/2$, we obtain

$$\mathbb{P}\left\{\sum_{\ell=1}^{L} Y_\ell^{\mathbf{k}} \geq \lceil \nu L \rceil\right\} = \mathbb{P}\left\{\sum_{\ell=1}^{L} Y_\ell^{\mathbf{k}} \geq \nu L\right\} = \mathbb{P}\left\{L^{-1} \sum_{\ell=1}^{L} Y_\ell^{\mathbf{k}} - \mu \geq \nu - \mu\right\}$$

$$\leq \mathbb{P}\left\{L^{-1} \sum_{\ell=1}^{L} Y_\ell^{\mathbf{k}} - \mu \geq \nu - 1/c\right\} \leq e^{-L(\nu-1/c)^2 \frac{1}{1-2\mu} \ln \frac{1-\mu}{\mu}}.$$

Since $\frac{1}{1-2\mu} \ln \frac{1-\mu}{\mu}$ is strictly decreasing, we continue

$$< e^{-L(c\nu-1)^2 \frac{\ln(c-1)}{c(c-2)}} \leq e^{\ln \delta - \ln |\Gamma|} = \frac{\delta}{|\Gamma|},$$

where Hoeffding's inequality holds since $0 < \nu - 1/c < \nu - \mu \leq 1/2 - \mu <$
$1 - \mu$. $\square$

The remaining part of this section works out and analyzes an algorithm that
classifies the frequencies of a given set of frequency candidates Γ in two disjoint
sets, see Algorithm 1. Its main idea is to apply L one-dimensional FFTs of lengths
$M_1, \ldots, M_L$ in lines 1–3 using the sampling values of the unknown sparse signal
$p(\mathbf{x}) = \sum_{\mathbf{k} \in I} \hat{p}_{\mathbf{k}} e^{2\pi i \mathbf{k} \cdot \mathbf{x}}$. This requires $\mathcal{O}(L M \log M)$ arithmetic operations, where
$M := \max\{M_\ell \colon \ell = 1, \ldots, L\}$. We rearrange the results $\hat{g}_h^{(\ell)}$ from line 2 using
$h = \mathbf{k} \cdot \mathbf{z}_\ell \bmod M_\ell$ in order to determine L (potentially) aliased Fourier coefficients
$\hat{p}_{\mathbf{k}}^{(\ell)} := \hat{g}_{\mathbf{k} \cdot \mathbf{z}_\ell \bmod M_\ell}^{(\ell)}, \ell = 1, \ldots, L$, for each $\mathbf{k} \in \Gamma$, which requires $\mathcal{O}(d L |\Gamma|)$ arith-
metic operations. Subsequently in line 6, one counts for fixed frequency $\mathbf{k} \in \Gamma$ how
many of the coefficients $\hat{p}_{\mathbf{k}}^{(\ell)}, \ell = 1, \ldots, L$, are non-zero. If the fraction of the $\hat{p}_{\mathbf{k}}^{(\ell)}$
that are non-zero is above a certain threshold, the frequency $\mathbf{k}$ is kept, as it is likely

to be part of the frequency support I of the unknown trigonometric polynomial p. So one collects this frequency in a set of detected frequencies $\tilde{I}$. If the fraction of the coefficients $\hat{p}_{\mathbf{k}}^{(\ell)}$ that are non-zero, $\ell = 1, \ldots, L$, is below the threshold, one discards the frequency $\mathbf{k}$. This classification has a computational complexity of $\mathcal{O}(L|\Gamma|)$. Altogether, this yields a total computational complexity of $\mathcal{O}(L(M \log M + d|\Gamma|))$ for Algorithm 1.

Algorithm 1 Detecting the frequency set of a trigonometric polynomial p.

Input:	$\Gamma \subset \mathbb{Z}^d$	set of frequency candidates
	$\Lambda := \bigcup_{\ell=1}^{L} \Lambda(\mathbf{z}_\ell, M_\ell)$	sampling nodes
	$\{p(\mathbf{x}) : \mathbf{x} \in \Lambda\}$	sampling values of $p \in \Pi_I$
	$\nu \in (0, 1/2]$	decision parameter

1: **for** $\ell = 1$ to L **do**
2:     Compute $\hat{g}_h^{(\ell)} := \sum_{j=0}^{M_\ell - 1} p\left(\frac{j}{M_\ell}\mathbf{z}\right) \mathrm{e}^{-2\pi \mathrm{i} j h / M_\ell}$, $h = 0, \ldots, M_\ell - 1$,
       using a 1d FFT.
3: **end for**
4: Set $\tilde{I} := \emptyset$.
5: **for each** $\mathbf{k} \in \Gamma$ **do**
6:     **if** $\sum_{\ell=1}^{L} \left(1 - \delta_0\big(\hat{g}_{\mathbf{k} \cdot \mathbf{z}_\ell \bmod M_\ell}^{(\ell)}\big)\right) \geq \nu L$ **then**
7:         Update $\tilde{I} := \tilde{I} \cup \{\mathbf{k}\}$.
8:     **end if**
9: **end for**

Output: $\tilde{I}$ detected frequency set

Computational complexity: $\mathcal{O}(L(M \log M + d|\Gamma|))$,
                where $M := \max\{M_\ell : \ell = 1, \ldots, L\}$

Remark 6 In Algorithm 1, we use a comparison to zero on line 6 implemented as Kronecker delta function δ_0. Clearly, numerical implementations should take into account numerical inaccuracies and utilize a suitable approximation of this function.

The next two lemmas estimate the failure probabilities of the proposed detection method and, subsequently, Corollary 1 reveals the connection to Theorem 3. In the following, we widely use the random variables $Y_\ell^{\mathbf{k}}$, that have already been defined in (7).

Lemma 2 (Probability estimate of false positive)
  For $\mathbf{k} \in \Gamma \setminus I$, $\nu \in (0, 1/2]$, and the computed Fourier coefficients $\hat{p}_{\mathbf{k}}^{(\ell)} := \hat{g}_{\mathbf{k} \cdot \mathbf{z}_\ell \bmod M_\ell}^{(\ell)}$ in line 2 of Algorithm 1, we observe

$$\mathbb{P}\left(\left\{|\{\ell \in \{1, \ldots, L\} : \hat{p}_{\mathbf{k}}^{(\ell)} \neq 0\}| \geq \nu L\right\}\right) \leq \mathbb{P}\left(\left\{\sum_{\ell=1}^{L} Y_\ell^{\mathbf{k}} \geq \nu L\right\}\right). \tag{8}$$

Proof Since $\mathbf{k} \in \Gamma \setminus I$, $\hat{p}_{\mathbf{k}}^{(\ell)} = 0$ holds in each case where $\mathbf{k}$ does not alias to any frequency $\mathbf{h} \in I$. Therefore, we necessarily need aliasing in order to obtain $\hat{p}_{\mathbf{k}}^{(\ell)} \neq 0$, which yields

$$
\left\{ \{\ell \in \{1, \ldots, L\} \colon \hat{p}_{\mathbf{k}}^{(\ell)} \neq 0\} \geq \nu L \right\} \subset \left\{ \begin{array}{l} \mathbf{k} \text{ aliased for at least } \nu L \text{ of the rank-1} \\ \text{lattices } \Lambda(\mathbf{z}_\ell, M_\ell), \ell = 1, \ldots, L, \text{ to at} \\ \text{least one } \mathbf{h} \in I \end{array} \right\}.
$$

$\square$

Lemma 3 (Probability estimate of false negative)

For $\mathbf{k} \in I \cap \Gamma$, $\nu \in (0, 1/2]$, and the computed Fourier coefficients $\hat{p}_{\boldsymbol{k}}^{(\ell)} := \hat{g}_{\boldsymbol{k} \cdot \mathbf{z}_\ell \bmod M_\ell}^{(\ell)}$ in line 2 of Algorithm 1, we observe

$$
\mathbb{P}\left(\left\{ |\{\ell \in \{1, \ldots, L\} \colon \hat{p}_{\boldsymbol{k}}^{(\ell)} \neq 0\}| < \nu L \right\} \right) \leq \mathbb{P}\left(\left\{ \sum_{\ell=1}^{L} Y_\ell^{\boldsymbol{k}} \geq L - \nu L \right\} \right). \quad (9)
$$

Proof We have

$$
\begin{aligned}
&\mathbb{P}\left(\left\{ |\{\ell \in \{1, \ldots, L\} \colon \hat{p}_{\mathbf{k}}^{(\ell)} \neq 0\}| < \nu L \right\} \right) \\
&= \mathbb{P}\left(\left\{ |\{\ell \in \{1, \ldots, L\} \colon \hat{p}_{\mathbf{k}}^{(\ell)} = 0\}| \geq L - \nu L \right\} \right)
\end{aligned}
$$

and the inclusion

$$
\begin{aligned}
&\left\{ |\{\ell \in \{1, \ldots, L\} \colon \hat{p}_{\mathbf{k}}^{(\ell)} = 0\}| \geq L - \nu L \right\} \\
&\subset \left\{ \begin{array}{l} \mathbf{k} \text{ aliased for at least } L - \nu L \text{ of the rank-1 lattices } \Lambda(\mathbf{z}_\ell, M_\ell), \ell = \\ 1, \ldots, L, \text{ to at least one } \mathbf{h} \in I \setminus \{\mathbf{k}\} \end{array} \right\},
\end{aligned}
$$

which holds since aliasing is necessary in order to even obtain $\hat{p}_{\mathbf{k}}^{(\ell)} \neq \hat{p}_{\mathbf{k}} \neq 0$. Consequently, we achieve

$$
\begin{aligned}
&\mathbb{P}\left(\left\{ |\{\ell \in \{1, \ldots, L\} \colon \hat{p}_{\mathbf{k}}^{(\ell)} \neq 0\}| < \nu L \right\} \right) \\
&\leq \mathbb{P}\left(\left\{ \begin{array}{l} \mathbf{k} \text{ aliased for at least } L - \nu L \text{ of the rank-1 lattices } \Lambda(\mathbf{z}_\ell, M_\ell), \ell = 1, \ldots, L, \\ \text{to at least one } \mathbf{h} \in I \setminus \{\mathbf{k}\} \end{array} \right\} \right) \\
&= \mathbb{P}\left(\left\{ \sum_{\ell=1}^{L} Y_\ell^{\mathbf{k}} \geq L - \nu L \right\} \right).
\end{aligned}
$$

$\square$

As a consequence of the last two lemmas, we need upper bounds on

$$
\mathbb{P}\left(\left\{ \sum_{\ell=1}^{L} Y_\ell^{\mathbf{k}} \geq \nu L \right\} \right) \quad \text{and} \quad \mathbb{P}\left(\left\{ \sum_{\ell=1}^{L} Y_\ell^{\mathbf{k}} \geq (1-\nu)L \right\} \right),
$$

which are themselves upper bounds on the failure probabilities of the classification of $\mathbf{k} \in \Gamma \setminus I$ and $\mathbf{k} \in I$ in line 6 of Algorithm 1, respectively. In order to apply Theorem 3, we need to choose $v \in (0, 1/2]$, observing that

$$\mathbb{P}\left(\left\{\sum_{\ell=1}^{L} Y_\ell^{\mathbf{k}} \geq (1-v)L\right\}\right) \leq \mathbb{P}\left(\left\{\sum_{\ell=1}^{L} Y_\ell^{\mathbf{k}} \geq vL\right\}\right). \tag{10}$$

This restriction on v allows for the application of Theorem 3 which leads to the following statement about Algorithm 1.

Corollary 1 *Let $\Gamma \subset \mathbb{Z}^d$ with $|\Gamma| < \infty$ be given and $I \subset \mathbb{Z}^d$ with $|I| < \infty$ be fixed but unknown. Moreover, we choose the parameters $v \in (0, 1/2]$ and $c > 1/v$. Furthermore, we fix $\delta \in (0, 1)$,*

$$L := \left\lceil \frac{c(c-2)}{(cv-1)^2 \ \ln(c-1)} \left(\ln |\Gamma| - \ln \delta \right) \right\rceil, \ and$$

$$M := \min \left\{ p \in P^{I \cup \Gamma} : p > c|I| \right\}.$$

Subsequently, we randomly choose $\mathbf{z}_1, \ldots, \mathbf{z}_L \in [0, M-1]^d \cap \mathbb{Z}^d$. In addition, we assume that the sampling points $\{(\mathbf{x}, p(\mathbf{x})) \in \mathbb{T}^d \times \mathbb{C} : \mathbf{x} \in \Lambda(\mathbf{z}_\ell, M), \ell = 1, \ldots, L\}$ of the multivariate trigonometric polynomial $p(\mathbf{x}) = \sum_{k \in I} \hat{p}_k \, e^{2\pi i k \cdot \mathbf{x}}, \ \hat{p}_k \neq 0$ for each $k \in I$, are given. Then, the probability that the output $\tilde{I}$ of Algorithm 1 does not equal $\Gamma \cap I$ is less than δ.

Proof Applying Theorem 3, Lemma 2, Lemma 3, and the union bound yields

$$\mathbb{P}\left(\tilde{I} \neq \Gamma \cap I\right)$$

$$= \ \mathbb{P}\left(\bigcup_{\mathbf{k} \in \Gamma \setminus I} \left\{|\{\ell : \hat{p}_{\mathbf{k}}^{(\ell)} \neq 0\}| \geq vL\right\} \cup \bigcup_{\mathbf{k} \in I \cap \Gamma} \left\{|\{\ell : \hat{p}_{\mathbf{k}}^{(\ell)} \neq 0\}| < vL\right\}\right)$$

$$\overset{(8) \ \& \ (9)}{\leq} \ \sum_{\mathbf{k} \in \Gamma \setminus I} \mathbb{P}\left(\left\{\sum_{\ell=1}^{L} Y_\ell^{\mathbf{k}} \geq vL\right\}\right) + \sum_{\mathbf{k} \in I \cap \Gamma} \mathbb{P}\left(\left\{\sum_{\ell=1}^{L} Y_\ell^{\mathbf{k}} \geq (1-v)L\right\}\right)$$

$$\overset{v \leq 1/2 \ \& \ (10)}{\leq} \ \sum_{\mathbf{k} \in \Gamma} \mathbb{P}\left(\left\{\sum_{\ell=1}^{L} Y_\ell^{\mathbf{k}} \geq vL\right\}\right) \overset{\text{Thm. 3}}{<} |\Gamma| \frac{\delta}{|\Gamma|} = \delta. \qquad \square$$

Up to now, we considered only the classification of the non-zero frequencies. Specific parameter choices allow for the additional computation of the unknown Fourier coefficients. To this end, we compute

$$\check{p}_{\mathbf{k}} := \text{median}\left\{\text{Re}\left(\hat{p}_{\mathbf{k}}^{(\ell)}\right) : \ell = 1, \ldots, L\right\} + i \cdot \text{median}\left\{\text{Im}\left(\hat{p}_{\mathbf{k}}^{(\ell)}\right) : \ell = 1, \ldots, L\right\}, \tag{11}$$

where $\hat{p}_{\mathbf{k}}^{(\ell)} := \hat{g}_{\mathbf{k}\cdot\mathbf{z}_\ell \bmod M_\ell}^{(\ell)}$. Choosing $\nu = 1/2$ and L odd, we obtain

$$
\check{p}_{\mathbf{k}} = \begin{cases} 0 & : \mathbf{k} \in \Gamma \setminus I, \\ \hat{p}_{\mathbf{k}} & : \mathbf{k} \in I \cap \Gamma, \end{cases}
$$

with probability at least $\eta := 1 - \mathbb{P}\left(\left\{\sum_{\ell=1}^L Y_\ell^{\mathbf{k}} \geq L/2\right\}\right)$ since there is no aliasing for at least $L/2$, i.e., $\frac{L+1}{2}$, rank-1 lattices on the frequency $\mathbf{k}$ with at least this probability. Accordingly, with probability η, we determine for at least $\frac{L+1}{2}$ different ℓ the correct values $\hat{p}_{\mathbf{k}}^{(\ell)} = \hat{p}_{\mathbf{k}}$, which implies the median is exactly this value. Algorithm 2 presents the resulting strategy for detecting the active frequencies $\mathbf{k} \in \Gamma$ as well as computing all medians of the sets $\{\hat{p}_{\mathbf{k}}^{(\ell)} : \ell = 1, \ldots, L\}$, $\mathbf{k} \in \tilde{I}$, as corresponding Fourier coefficients.

Algorithm 2 Detecting the frequency set and computing the Fourier coefficients of a trigonometric polynomial p; (Algorithm 1 with $\nu := 1/2$ and computation of Fourier coefficients).

Input: $\Gamma \subset \mathbb{Z}^d$ set of frequency candidates

           $\Lambda := \bigcup_{\ell=1}^L \Lambda(\mathbf{z}_\ell, M_\ell)$ sampling nodes, L odd

           $\{p(\mathbf{x}) : \mathbf{x} \in \Lambda\}$ sampling values of $p \in \Pi_I$

1: **for** $\ell = 1$ to L **do**

2:      Compute $\hat{g}_h^{(\ell)} := \frac{1}{M_\ell} \sum_{j=0}^{M_\ell-1} p\left(\frac{j}{M_\ell}\mathbf{z}\right) e^{-2\pi \mathrm{i} jh/M_\ell}$, $h = 0, \ldots, M_\ell - 1$,

         using a 1d FFT.

3: **end for**

4: Set $\tilde{I} := \emptyset$.

5: **for each** $\mathbf{k} \in \Gamma$ **do**

6:      **if** $\sum_{\ell=1}^L \left(1 - \delta_0\left(\hat{g}_{\mathbf{k}\cdot\mathbf{z}_\ell \bmod M_\ell}^{(\ell)}\right)\right) \geq L/2$ **then**

7:         Update $\tilde{I} := \tilde{I} \cup \{\mathbf{k}\}$.

8:         With $\hat{p}_{\mathbf{k}}^{(\ell)} := \hat{g}_{\mathbf{k}\cdot\mathbf{z}_\ell \bmod M_\ell}^{(\ell)}$, compute

$$
\check{p}_{\mathbf{k}} := \mathrm{median}\left\{\mathrm{Re}\left(\hat{p}_{\mathbf{k}}^{(\ell)}\right) : \ell = 1, \ldots, L\right\}
$$
$$
+ \mathrm{i} \cdot \mathrm{median}\left\{\mathrm{Im}\left(\hat{p}_{\mathbf{k}}^{(\ell)}\right) : \ell = 1, \ldots, L\right\}.
$$

9:      **end if**

10: **end for**

Output: $\tilde{I}$ detected frequency set

          $(\check{p}_{\mathbf{k}})_{\mathbf{k}\in\tilde{I}}$ Fourier coefficients belonging to $\tilde{I}$

Computational complexity: $\mathscr{O}\left(L(M \log M + d\,|\Gamma|)\right)$,
                     where $M = \max\{M_\ell : \ell = 1, \ldots, L\}$

The differences to Algorithm 1 are the fixed parameter choice $\nu := 1/2$ and the additional line 8 in Algorithm 2, which computes the Fourier coefficients $\check{p}_{\mathbf{k}}$, $\mathbf{k} \in \tilde{I}$, as medians using the method from [3]. Since $\tilde{I} \subset \Gamma$, Algorithm 2 requires at most $\mathscr{O}(L\,|\Gamma|)$ additional arithmetic operations. Thus, we obtain the same computational

complexities for Algorithms 1 and 2. Furthermore, Algorithm 2 yields the Fourier coefficients of the active frequencies with high probability.

Corollary 2 *Let $\Gamma \subset \mathbb{Z}^d$ with $|\Gamma| < \infty$ be given and $I \subset \mathbb{Z}^d$ with $|I| < \infty$ be fixed but unknown. Moreover, we choose $\nu := 1/2$ and $c > 2$. In addition, we fix $\delta \in (0, 1)$,*

$$L := \min \left\{ n \in 2\mathbb{N} + 1 : n \geq \frac{4c}{(c-2)\ln(c-1)} \left(\ln |\Gamma| - \ln \delta \right) \right\}, \text{ and}$$

$$M := \min \left\{ p \in P^{I \cup \Gamma} : p > c|I| \right\}.$$

Subsequently, we randomly choose $\mathbf{z}_1, \ldots, \mathbf{z}_L \in [0, M-1]^d \cap \mathbb{Z}^d$. In addition, we assume that the sampling points $\{(\mathbf{x}, p(\mathbf{x})) \in \mathbb{T}^d \times \mathbb{C} : \mathbf{x} \in \Lambda(\mathbf{z}_\ell, M), \ell = 1, \ldots, L\}$ of the multivariate trigonometric polynomial $p(\mathbf{x}) = \sum_{\mathbf{k} \in I} \hat{p}_{\mathbf{k}} e^{2\pi i \mathbf{k} \cdot \mathbf{x}}$, $\hat{p}_{\mathbf{k}} \neq 0$ for each $\mathbf{k} \in I$, are given. Then, the probability that the output $(\check{p}_{\mathbf{k}})_{\mathbf{k} \in \tilde{I}}$ of Algorithm 2 does not match the correct Fourier coefficients, i.e., $\tilde{I} \neq I \cap \Gamma$ or $\check{p}_{\mathbf{k}} \neq \hat{p}_{\mathbf{k}}$ for at least one $\mathbf{k} \in \tilde{I}$, is bounded from above by δ.

Proof We follow the proof of Corollary 1 for estimating the probability, and we take into account the argumentations on the medians $\check{p}_{\mathbf{k}}$ immediately above this Corollary. □

Remark 7 The parameters in Corollary 2 need to be chosen carefully depending on the specific application. In particular, it might be useful to increase the parameter c, which leads to possibly larger lattice sizes M but smaller numbers L of used rank-1 lattices.

Remark 8 The approach of using medians or medians of means for improved bounds on the probability of failure in sketching procedures is a well-established technique in the computer science literature. Such an approach was also used in the combinatorial reconstruction algorithm for compressed sensing presented in [12], whose ideas form the basis for a number of sparse fast Fourier transform constructions (the first such construction was provided in [20]). These ideas were then combined with rank-1 lattices in [22] to analyze the multivariate setting. In some sense, Algorithm 2 is analogous to Algorithm 2 in [22] except for the crucial difference that their approach uses a fixed generating vector and multiple lattice sizes, while we vary the generating vectors.

Theorem 1 states the result of Corollary 2 for the specific parameter choice $c := 10.33$ and the restriction on the frequency sets given in (1). The choice of $c := 10.33$ allows for adequate expansions of the candidate sets Γ, cf. (1), and simultaneously yields reasonable constants. In fact, we numerically computed the minimal value of the upper bound

$$L M \overset{|\Gamma| \geq 8}{\leq} |I| \ln \frac{|\Gamma|}{\delta} \underbrace{\left(\max_{|I| \geq 1} \frac{\text{nextprime}(c|I|)}{c|I|} \right) \frac{4c^2}{(c-2)\ln(c-1)} \left(1 + \frac{(c-2)\ln(c-1)}{2c\ln(8)} \right)}_{=:K(c)}$$

on LM out of the candidate set $c \in \{10 + n/100 \colon n \in \mathbb{N}, 0 \le n \le 2700\}$ in order to determine $c := 10.33$ with $K(c) < 37$.

Proof of Theorem 1 We apply Corollary 2 setting the parameter $c := 10.33$. Since the subset property (1) of the frequency sets is assumed, $P^{I \cup \Gamma}$ contains each prime number p larger than $c|I|$.

The ratio $\frac{M}{c|I|}$ of the smallest prime number M larger than $c|I|$ is bounded from above by $\frac{M}{c|I|} \le \frac{127}{11 \cdot 10.33}$, which can be proven using [15, Proposition 5.4] and the calculation of finitely many instances of this ratio. Accordingly, the lattice sizes are bounded from above by $M \le \frac{127}{11}|I|$. In addition, for $|\Gamma| \ge 8 > e^2$, the number of lattices L is bounded from above by

$$
\begin{aligned}
L &\le \frac{4c}{(c-2)\ln(c-1)}\,(\ln|\Gamma| - \ln\delta) + 2 \\
&\le \frac{4c}{(c-2)\ln(c-1)}\,(\ln|\Gamma| - \ln\delta) + \frac{2}{\ln 8}(\ln|\Gamma| - \ln\delta) \\
&\le 3.183\,(\ln|\Gamma| - \ln\delta)
\end{aligned}
$$

where in the second inequality, we used that $0 < \delta < 1$ holds. This yields a total number of samples $ML < 37|I|\,(\ln|\Gamma| - \ln\delta)$. According to Corollary 2, we apply Algorithm 2 and obtain the non-zero ones of the Fourier coefficients $\hat{p}_{\mathbf{k}}$, $\mathbf{k} \in I = I \cap \Gamma$, with probability at least $1 - \delta$. □

Remark 9 Clearly, the reconstruction guarantees in Theorem 1 as well as Corollary 2 hold with probability of at least $1 - \delta$. However, from a practical point of view, it is reasonable to expect that the output $\tilde{I}$ of Algorithms 1 and 2 contains I, i.e., $I \subset \tilde{I}$, with significantly higher probability. On the one hand, the estimate (9) in Lemma 3 is very rough since we estimate the probability of cancellations of aliasing Fourier coefficients just by the probability of aliasing frequencies, which are widely differing events in fact. On the other hand, the probability that $I \not\subset \tilde{I}$ holds, i.e., that we observe false negatives, is bounded from above by $\frac{|I|}{|\Gamma|}\delta$, cf. the proof of Corollary 1. Moreover, the probability that we will observe false positives seems to be significantly higher, cf. Section 4.1, which also fits well to the theoretical estimate on the failure probability in Corollary 1 since the upper bound on the probability of observing false positives contributes much more to the upper bound δ of the total failure probability when one assumes that $|I| \ll |\Gamma|$.

From that point of view, the probability of $I \subset \tilde{I}$ in conjunction with, for instance, $|\tilde{I}| \le 2|I|$ may be considerably larger than the probability we estimated in Theorem 1. Even in these cases, one has a reasonable chance to entirely identify the trigonometric polynomial p from the already known sampling values.

An improvement strategy is to postprocess the frequency set $\tilde{I}$ and Fourier coefficients $\check{p}_{\mathbf{k}}$, $\mathbf{k} \in \tilde{I}$, obtained as output of Algorithm 2. One considers $\tilde{I}$ as the frequency set of the polynomial p and one computes the corresponding Fourier coefficients $\hat{p}_{\mathbf{k}}$, $\mathbf{k} \in \tilde{I}$, from the already available sampling values along the fixed rank-1 lattices $\Lambda(\mathbf{z}_1, M_1), \ldots, \Lambda(\mathbf{z}_L, M_L)$ using [27, Algorithm 2] or, even more general, one applies a least squares method. If this set of rank-1 lattices provides a spatial

discretization of the space $\Pi_{\tilde{I}}$ of trigonometric polynomials, cf. [25] for details, and $\tilde{I} \supset I$ holds, we will entirely identify the trigonometric polynomial p. Otherwise, we can use the following hybrid approach. For each rank-1 lattice $\Lambda(\mathbf{z}_\ell, M_\ell)$, we determine the frequency set $\tilde{I}^{(\ell)} := \{\mathbf{k} \in \tilde{I} : \mathbf{k} \cdot \mathbf{z}_\ell \not\equiv \mathbf{h} \cdot \mathbf{z}_\ell (\mathrm{mod}\ M_\ell) \forall \mathbf{h} \in \tilde{I} \setminus \{\mathbf{k}\}\}$ belonging to reconstructable Fourier coefficients $\hat{p}_{\mathbf{k}}$ for any $p \in \Pi_{\tilde{I}}$ and set $\check{p}_{\mathbf{k}} := \hat{p}_{\mathbf{k}}^{\Lambda(\mathbf{z}_\ell, M_\ell)}$ for $\mathbf{k} \in \tilde{I}^{(\ell)}$. Whenever a frequency $\mathbf{k} \in \tilde{I}$ is contained in more than one $\tilde{I}^{(\ell)}$, we set the Fourier coefficient $\check{p}_{\mathbf{k}}$ to the average of the corresponding coefficients $\hat{p}_{\mathbf{k}}^{\Lambda(\mathbf{z}_\ell, M_\ell)}$. If there should be frequencies $\mathbf{k} \in \tilde{I} \setminus \left(\cup_{\ell=1}^{L} \tilde{I}^{(\ell)}\right)$, then we just use the Fourier coefficients $\check{p}_{\mathbf{k}}$ computed in line 8 of Algorithm 2 as a fall-back.

Numerical tests, cf. Section 4.1, confirm that this postprocessing strategy works very well. However, we cannot directly apply the theoretical results from [25], since the frequency set $\tilde{I}$ already depends on the aliasing effects of the rank-1 lattices $\Lambda(\mathbf{z}_1, M_1), \ldots, \Lambda(\mathbf{z}_L, M_L)$.

3 A sparse FFT for high-dimensional data

3.1 Dimension-incremental sparse FFT – background and previous works

As already mentioned in the introduction, one of the main motivations for our algorithm is to apply it as a key ingredient of a dimension-incremental algorithm for the high-dimensional fast Fourier transform. Such algorithms have received considerable attention in recent years due to their excellent and reliable applicability in high-dimensional settings, cf., e.g., [8, 9, 26, 33].

To precisely formulate the algorithmic framework of these approaches that will also form the basis of our sparse FFT, we recall some notation: We denote the multivariate trigonometric polynomial that we aim to recover by

$$p \colon \mathbb{T}^d \to \mathbb{C}, \qquad p(\mathbf{x}) := \sum_{\mathbf{k} \in I} \hat{p}_{\mathbf{k}}\, \mathrm{e}^{2\pi i \mathbf{k} \cdot \mathbf{x}}, \quad \hat{p}_{\mathbf{k}} \in \mathbb{C} \setminus \{z \in \mathbb{C} : |z| < 3\theta\}, \quad (12)$$

for some frequency set $I = \mathrm{supp}\ \hat{p} \subset \mathbb{Z}^d$, $s := |I| < \infty$. Furthermore, we assume that we know a (possibly very large) candidate set $\Gamma \subset \mathbb{Z}^d$ of finite cardinality for I, i.e., $I \subset \Gamma$ and $|\Gamma| < \infty$.

In this notation our key dimensional-incremental strategy, in analogy to a number of recent approaches in the literature, such as [8, 9], amounts to first identifying $I^{(t)}$ as a superset of the projection

$$\mathscr{P}_t(I) := \{k_t \in \mathbb{Z} : \mathbf{k} = (k_1, \ldots, k_d)^\top \in I\}$$

to the tth component of the active frequencies in I for each coordinate $t \in \{1, \ldots, d\}$, and then incrementally combining ("pairing") the elements of the different $I^{(t)}$ to iteratively obtain a superset $I^{(1,\ldots,t)}$ of

$$\mathscr{P}_{1,\ldots,t}(I) := \{(k_1, \ldots, k_t) \in \mathbb{Z}^t : \mathbf{k} = (k_1, \ldots, k_d)^\top \in I\},$$

which will eventually yield a set of multi-indices $I^{(1,\ldots,d)} \supset \mathscr{P}_{1,\ldots,d}(I) = I$ that contains the active ones.

In each of these pairing steps, one determines the candidates for the indices in $I^{(1,\dots,t)}$ by appropriately combining the (already identified) frequency set $I^{(1,\dots,t-1)}$ for $\mathscr{P}_{1,\dots,t-1}(I)$ with the (already identified) frequency set $I^{(t)}$ for $\mathscr{P}_t(I)$. That is, one aims to find $|I|$ active elements from the larger candidate set $(I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathscr{P}_{1,\dots,t}(\Gamma) =: J_t$. This problem is exactly of the form studied in the previous section, and basically this step is where the various approaches that have been proposed in the literature differ. Treating this step as a black box, say Algorithm A, we can formulate the dimension-incremental approach to high-dimensional fast transforms as a meta-algorithm, which is summarized in Algorithm 3.

To apply our algorithm for a wider class of scenarios, we introduced the two additional parameters θ and s_{local} for enhanced stability and robustness. The threshold parameter θ is meant to account for scenarios where due to noise or model mismatch, some linear combination of non-significant coefficients is not evaluated to exactly zero, but just approximately zero. More precisely, one is not aiming to identify all non-zero coefficients, but rather only those with absolute values of at least θ. Naturally the parameter θ should be chosen smaller than the values to be expected, see also (12).

The parameter s_{local} is related to the observation that allowing for a moderate number of false positives can significantly improve the success rate for recovery in practice. If Step 2c of Algorithm 3 is designed to identify only exactly s coefficients, then a false positive can lead to the situation where an active frequency no longer corresponds to any of the s largest coefficients, and hence is not identified.

For that reason it is useful to collect $s_{\text{local}} > s$ frequencies in the intermediate steps of Algorithm 3 in practice, and we incorporate this feature in Algorithm 3 by using $s_{\text{local}} := 2s$ as the default choice.

In general, a major computational bottleneck of such dimension-incremental approaches is the iterated signal reconstruction from sampling values, (i.e., Step 2 in the formalization of Algorithm 3). In the simplest case, each such instance can be thought of as computing a pseudoinverse of a matrix. Since the matrices to be inverted change in every step of the dimension-incremental algorithm, a procedure for highly efficient computation of such a pseudoinverse is the key to the computational efficiency of the entire method. If one has FFT-like algorithms at one's disposal, the computational complexity will be tremendously reduced as compared to algorithms entirely based on computing full matrix vector products. Such high-dimensional FFT-like algorithms, however, typically need to exploit some structure of the sampling pattern and are hence restricted to specific sampling sets. Even for high-dimensional trigonometric polynomials on a regular full grid, this structure will likely get lost in the dimension-incremental procedure: As soon as previous iterations have identified an unstructured candidate set, this set will no longer have the advantageous structure of the underlying grid that can be exploited for computing the FFT.

Algorithm 2 in contrast, yields a fast transform for arbitrary candidate sets. Consequently, it can be efficiently applied in every iteration no matter what the candidate set looks like, which in turn gives rise to the superior computational complexity of the high dimensional sparse fast Fourier transform resulting from employing Algorithm 2 (together with some subsequent low-cost computations) in the role of Algorithm A in the context of Algorithm 3.

Algorithm 3 Dimension-incremental reconstruction of a multivariate trigonometric polynomial p from sampling values.

Input:	$\Gamma \subset \mathbb{Z}^d$	search space in frequency domain, candidate set for $I = \operatorname{supp} \hat{p}$
	$p(\circ)$	trigonometric polynomial p as black box (function handle)
	$s, s_{\text{local}} \in \mathbb{N}$	sparsity parameter, $s \leq s_{\text{local}}$ ($s_{\text{local}} := 2s$ by default)
	Algorithm A	**efficient algorithm** A that guarantees the identification of the frequency support of each s_{local}-sparse trigonometric polynomial in Π_{J_t} w.h.p., cf. Section 3.2, and computes the Fourier coeffcients
	$\theta \in \mathbb{R}^+$	absolute threshold (e.g. close to machine precision)
	$r \in \mathbb{N}$	number of detection iterations

(Step 1) [Single frequency component identification]

 for $t := 1, \ldots, d$ **do**

  Set $K_t := \max(\mathscr{P}_t(\Gamma)) - \min(\mathscr{P}_t(\Gamma)) + 1$, $I^{(t)} := \emptyset$.

  for $i := 1, \ldots, r$ **do**

   Choose $x'_j \in \mathbb{T}$, $j \in \{1, \ldots, d\} \setminus \{t\}$ uniformly at random.

   Set $\mathbf{x}^{(\ell)} := \left(x_1^{(\ell)}, \ldots, x_d^{(\ell)}\right)^\top$, $x_j^{(\ell)} := \begin{cases} \ell/K_t, & j = t, \\ x'_j, & j \neq t, \end{cases}$ for all $\ell = 0, \ldots, K_t - 1$.

   Compute $\tilde{\hat{p}}_{t,k_t} := \frac{1}{K_t} \sum_{\ell=0}^{K_t-1} p\left(\mathbf{x}^{(\ell)}\right) e^{-2\pi i \ell k_t / K_t}$, $k_t \in \mathscr{P}_t(\Gamma)$, via FFT.

   Set $I^{(t)} := I^{(t)} \cup \{k_t \in \mathscr{P}_t(\Gamma): \tilde{\hat{p}}_{t,k_t}$ is among the largest s_{local} (in absolute value)

        elements of $\{\tilde{\hat{p}}_{t,j}\}_{j \in \mathscr{P}_t(\Gamma)}$ and $|\tilde{\hat{p}}_{t,k_t}| \geq \theta\}$.

  end for i

 end for t

(Step 2) [Coupling frequency components identification]

 for $t := 2, \ldots, d$ **do**

  If $t < d$, set $\tilde{r} := r$ and $\tilde{s} := s_{\text{local}}$, otherwise $\tilde{r} := 1$ and $\tilde{s} := s$. Set $I^{(1,\ldots,t)} := \emptyset$.

  for $i := 1, \ldots, \tilde{r}$ **do**

   Choose components $x'_{t+1}, \ldots, x'_d \in \mathbb{T}$ of sampling nodes uniformly at random.

(Step 2a)

   Generate a sampling set $\mathscr{X} \subset \mathbb{T}^t$ for $J_t := (I^{(1,\ldots,t-1)} \times I^{(t)}) \cap \mathscr{P}_{(1,\ldots,t)}(\Gamma)$ that allows for the application of Algorithm A. Set $\mathscr{X}_{t,i} := \{\mathbf{x} := (\tilde{\mathbf{x}}, x'_{t+1}, \ldots, x'_d): \tilde{\mathbf{x}} \in \mathscr{X}\} \subset \mathbb{T}^d$.

(Step 2b)

   Sample p along the nodes of the sampling set $\mathscr{X}_{t,i}$.

(Step 2c)

   Apply Algorithm A to obtain the support $\tilde{J}_{t,i} \subset J_t$, $|\tilde{J}_{t,i}| \leq \tilde{s}$, of frequencies belonging to the at most $\tilde{s}$ largest Fourier coefficients, each larger than θ in absolute value, using the sampling values $p(\mathbf{x}_j)$, $\mathbf{x}_j \in \mathscr{X}_{t,i}$.

(Step 2d)

   Set $I^{(1,\ldots,t)} := I^{(1,\ldots,t)} \cup \tilde{J}_{t,i}$.

  end for i

 end for t

(Step 3) [Computation of Fourier coefficients]

 Generate a sampling set $\mathscr{X} \subset \mathbb{T}^d$ for $I^{(1,\ldots,d)}$ such that the corresponding Fourier matrix $\mathbf{A}(\mathscr{X}, I^{(1,\ldots,d)})$ is of full column rank and its pseudoinverse can be applied **efficiently**. Compute the corresponding Fourier coefficients $\left(\tilde{\hat{p}}_{(1,\ldots,d),\mathbf{k}}\right)_{\mathbf{k} \in I^{(1,\ldots,d)}}$.

 Set $\tilde{I} := \Big\{\mathbf{k} \in I^{(1,\ldots,d)}: \tilde{\hat{p}}_{(1,\ldots,d),\mathbf{k}}$ is among the largest s (in absolute value) elements of

     $\{\tilde{\hat{p}}_{(1,\ldots,d),\mathbf{j}}\}_{\mathbf{j} \in I^{(1,\ldots,d)}}$ and $|\tilde{\hat{p}}_{(1,\ldots,d),\mathbf{k}}| \geq \theta\Big\}$, $\tilde{\hat{\mathbf{p}}} := \left(\tilde{\hat{p}}_{(1,\ldots,d),\mathbf{k}}\right)_{\mathbf{k} \in \tilde{I}}$.

Output:	$\tilde{I} \subset \Gamma \subset \mathbb{Z}^d$	index set of detected frequencies, $|\tilde{I}| \leq \min\{s, |\Gamma|\}$
	$\tilde{\hat{\mathbf{p}}} \in \mathbb{C}^{|\tilde{I}|}$	corresponding Fourier coefficients, $|\tilde{\hat{p}}_{(1,\ldots,d),\mathbf{k}}| \geq \theta$

3.2 Sample complexity and computational complexity

3.2.1 Algorithm 3 using a general efficient identification Algorithm A

Since we proposed to consider Algorithm 3 as a meta-algorithm, we start by ana-
lyzing its sample complexity as well as its computational complexity in general
terms.

First we consider the number of samples used by Algorithm 3 for each step sep-
arately. The general results are collected in Table 2. In Step 1, we use $r \sum_{t=1}^{d} K_t \le$
$d\, r\, N_\Gamma$ many sampling values. For Step 2 we construct r different sampling sets $\mathscr{X}_{t,i}$
for each $t = 2, \ldots, d-1$ and, in addition, one sampling set $\mathscr{X}_{d,1}$, i.e., $r\,(d-2)+1$
many. Since different choices for Algorithm A require different sampling sets and
hence the sampling strategy must be chosen in conjunction with the algorithm substi-
tuted for this black box, we denote the sampling sets by $\mathscr{X}_{t,i} = \mathscr{X}_{t,i}(A)$. Certainly,
the sampling sets $\mathscr{X}_{t,i}(A)$ may depend on the admissible failure probability γ_A of
Algorithm A. With this notation, the number of sampling values used in Step 2 is
bounded by $|\mathscr{X}_{d,1}(A)| + \sum_{t=2}^{d-1} \sum_{i=1}^{r} |\mathscr{X}_{t,i}(A)|$. For Step 3, we need a number of
sampling nodes on the order of $\max(s, N_\Gamma) \log(s/\gamma)$ for a parameter γ. Namely, this
is the number of sampling nodes required by [26, Algorithm 1] to realize a spatial
discretization of trigonometric polynomials with frequencies supported in $I^{(1,\ldots,d)}$
with probability at least $1 - \gamma$ (see [26] for details).

Second, we consider the computational complexity, again for each step. In Step 1,
we apply r one-dimensional FFTs of maximal size N_Γ, d times each. Moreover, each
of the d different frequency sets $I^{(t)}$, $t = 1, \ldots, d$, is constructed incrementally in r
substeps, e.g., by sorting vectors $(|\tilde{\hat{p}}_{t,j}|)_{j \in \mathscr{P}_t(\Gamma)}$ of length at most N_Γ and handling
a sorted vector of length at most N_Γ. Accordingly, the computational complexity of
Step 1 is in $d\, r\, N_\Gamma \log N_\Gamma$.

The computational complexity of Step 2 intrinsically depends on the choice of
Algorithm A. This dependency is two-fold. On the one hand, in Step 2c, the algorithm
is executed, hence the runtime of this step directly corresponds to the computational
complexity of Algorithm A. On the other hand, different choices for Algorithm A
will require different sampling sets, which can be costly to construct if specific prop-
erties are required. Hence the computational complexity of Step 2a, in which the
sampling set is constructed will also depend on Algorithm A. We will subsume the
A-dependent contribution to the computational complexity arising in these two steps

Table 2 Sample complexities and computational complexities for the different steps of Algorithm 3,
where $\mathscr{C}(A)$ denotes the maximum computational cost of a single invocation of the efficient identifica-
tion Algorithm A and where the multiple rank-1 lattice approach from [26, Algorithms 1 and 2] is used in
Step 3

	Sample complexity	Computational complexity
Step 1	$d\, r\, N_\Gamma$	$d\, r\, N_\Gamma \log N_\Gamma$
Step 2	$\|\mathscr{X}_{d,1}(A)\| + \sum_{t=2}^{d-1} \sum_{i=1}^{r} \|\mathscr{X}_{t,i}(A)\|$	$d\, r\, \mathscr{C}(A) + d\, r\, s\, \log(rs)$
Step 3	$\max(s, N_\Gamma) \log(s/\gamma)$	$\max(s, N_\Gamma) \log(s/\gamma)(d + \log(s\, N_\Gamma))$

in a constant $\mathscr{C}(A)$, chosen to be a universal upper bound for these contributions over all possible choices of t and i. Step 2b also has a mild implicit dependency on Algorithm A, as it scales with the size of the sampling set. However, it should not be seen as a part of the algorithm, as it also depends on the sampling procedure in the underlying application, which is independent of the algorithm design. In this paper, we will follow the assumption made in most other works on the topic that the sampling procedure is dominated by the other steps of the algorithms, i.e., the computational complexity of Step 2b is also of order $\mathscr{O}(\mathscr{C}(A))$. At this point, we would remind the reader once again that we tolerate a specific failure probability γ_A of Algorithm A which may imply a γ_A dependence of $\mathscr{C}(A)$.

Step 2d is about adding at most s_{local} elements to the set $I^{(1,...,t)}$; the largest run-time contribution of this step is to avoid listing an existing element again. In analogy to the considerations of Step 1, we observe a computational complexity in $\mathscr{O}(d\,r\,s\,\log(rs))$ for Step 2d under the assumption $s_{\mathrm{local}} \sim s$.

As Step 3 is based on [26, Algorithms 1 and 2], the results in this paper provide bounds for the computational complexity of constructing the sampling set and computing the Fourier coefficients. We obtain $\mathscr{O}(\max(s, N_\Gamma)\log(s/\gamma)(d + \log(s\,N_\Gamma)))$, with the parameter γ as introduced in the beginning of this subsection.

3.2.2 Algorithm 3 using a modification of Algorithm 2 in the role of Algorithm A

Algorithm 4 Modification of Algorithm 2 limiting the output.

Input:	$\Gamma \subset \mathbb{Z}^d$	set of frequency candidates
	$\Lambda := \bigcup_{\ell=1}^{L} \Lambda(\mathbf{z}_\ell, M_\ell)$	sampling nodes, L odd
	$\{p(\mathbf{x}) : \mathbf{x} \in \Lambda\}$	sampling values of $p \in \Pi_I$
	$\tilde{s}$	sparsity parameter
	θ	Fourier coefficient threshold

1: Compute $\tilde{I}$ and $(\check{p}_{\mathbf{k}})_{\mathbf{k} \in \tilde{I}}$ using Algorithm 2
2: Set $\tilde{I} := \{\mathbf{k} \in \tilde{I} : : |\check{p}_{\mathbf{k}}| \geq \theta\}$ and $\tilde{s} = \min(\tilde{s}, |\tilde{I}|)$.
3: Sort the entries of $(\check{p}_{\mathbf{k}})_{\mathbf{k} \in \tilde{I}}$ by their modulus in descending order to obtain $(\check{p}_{\mathbf{k}_1}, \check{p}_{\mathbf{k}_2}, \dots)$ and set $\tilde{I} := \{\mathbf{k}_1, \dots, \mathbf{k}_{\tilde{s}}\}$.

Output:	$\tilde{I}$	detected frequency set $|\tilde{I}| \leq \tilde{s}$
	$(\check{p}_{\mathbf{k}})_{\mathbf{k} \in \tilde{I}}$	Fourier coefficients belonging to $\tilde{I}$, i.e., $|\check{p}_{\mathbf{k}}| \geq \theta$

Computational complexity: $\mathscr{O}(L(M \log M + d\,|\Gamma|) + |\Gamma| \log |\Gamma|)$,
where $M = \max\{M_\ell : \ell = 1, \dots, L\}$

In this section, we analyze Algorithm 3 when using multiple random rank-1 lattices for constructing the first t components of the sampling sets $\mathscr{X}_{t,i}$ in Step 2a according to Corollary 2 and applying Algorithm 2 as Algorithm A in Step 2c, where we use J_t as the set of frequency candidates Γ for Algorithm 2. Since we assumed Algorithm A as an estimator of the $\tilde{s}$ most significant frequencies of the input signal, we need to apply a slight modification of Algorithm 2 which we summarize

in Algorithm 4. It is just the application of Algorithm 2 with a subsequent additional restriction of the output $\tilde{I}$ guaranteeing that this frequency set then fits to the requirements of Step 2c in Algorithm 3.

As above, we assume $s_{\text{local}} \sim s$, e.g., $s_{\text{local}} := 2s$, in order to avoid at least this parameter. It suffices to discuss Step 2 since the complexity of the other steps is independent of Algorithm A and has already been investigated in the last section. Some parameters related to the probability of failure and the number of iterations will not be specified in this section, the next section discusses suitable choices. In particular, the failure probability γ_{A} of Algorithm A still remains unspecified in this section. In Table 3, we give an overview of the sample complexities as well as computational complexities of the three steps involving these parameters r, γ_{A}, and γ. The corresponding complexities for the suitable parameter choices will again be postponed to the next section.

Again, we start with the sample complexity. Each of the sampling sets $\mathscr{X}_{t,i}$ is the union of $\mathscr{O}\left(\log(|J_t|/\gamma_{\text{A}})\right)$ rank-1 lattices in t spatial dimensions, consisting of $M_{t,i,\ell} = \mathscr{O}\left(\max(s, N_\Gamma)\right)$ lattice nodes each, which are embedded into d dimensions by concatenating all the lattice nodes by $d - t$ fixed components (the same for all nodes in the sampling set $\mathscr{X}_{t,i}$), which are drawn uniformly at random from $\mathbb{T}$. The generating vectors $\mathbf{z}_{t,i,\ell}$ of the t-dimensional rank-1 lattices are drawn independently of these components and of each other, uniformly at random from $[0, M_{t,i,\ell} - 1]^t \cap \mathbb{Z}^t$. As $|J_t| \lesssim r s N_\Gamma$, an upper bound of the size of each sampling set is given by $|\mathscr{X}_{t,i}| \lesssim \max(s, N_\Gamma) \log(r s N_\Gamma/\gamma_{\text{A}})$. As the number of these sampling sets $\mathscr{X}_{t,i}$ is $1 + (d - 2) r$, we obtain that the sampling complexity of Step 2 is $\mathscr{O}\left(d r \max(s, N_\Gamma) \log(r s N_\Gamma/\gamma_{\text{A}})\right)$.

In order to estimate the computational complexity of Step 2 with Algorithm 4 taking the role of Algorithm A, we distinguish two different cases. First, we consider the worst case scenario. For this, we just use the estimate $|J_t| \lesssim r s N_\Gamma$ and apply Theorem 1 for each t and i, where J_t takes the role of the set of frequency candidates Γ in Theorem 1. Accordingly, we observe a computational complexity in

$$\mathscr{O}\left(d r \left(\max(s, N_\Gamma) \log(s N_\Gamma) + d r s N_\Gamma\right) \log \frac{r s N_\Gamma}{\gamma_{\text{A}}}\right). \tag{13}$$

Table 3 Sample complexities and computational complexities for the different steps of Algorithm 3, where the efficient identification Algorithm 4 is used in Step 2 and the multiple rank-1 lattice approach from [26, Algorithm 1] in Step 3

	Sample complexity	Computational complexity
Step 1	$d r N_\Gamma$	$d r N_\Gamma \log N_\Gamma$
Step 2 (w.h.p.)	$d r \max(s, N_\Gamma) \log \frac{s N_\Gamma}{\gamma_{\text{A}}}$	$d^2 r s N_\Gamma \log^2\left(\frac{s N_\Gamma}{\gamma_{\text{A}}}\right)$
Step 2 (w.c.)	$d r \max(s, N_\Gamma) \log \frac{r s N_\Gamma}{\gamma_{\text{A}}}$	$d^2 r^2 s N_\Gamma \log^2\left(\frac{r s N_\Gamma}{\gamma_{\text{A}}}\right)$
Step 3	$\max(s, N_\Gamma) \log(s/\gamma)$	$\max(s, N_\Gamma) \log(s/\gamma)(d + \log(s N_\Gamma))$

This estimate, however, is far from order optimal in many cases, as the bound on $|J_t|$ is approximately tight only when the sets $\tilde{J}_{t-1,i}$ constructed for different values of i have very little overlap.

In the event that Step 1 of Algorithm 3, i.e., $P_t(I) \subset I^{(t)}$ and, in addition, all instances of Algorithm 4 are successful (in the sense of Corollary 2, cf. Section 3.2.3), these sets will have large overlap. As we will discuss in the next section, this event has high probability for suitable parameter choices. That is, in such scenarios, the computational complexity will be considerably smaller with high probability. More precisely, in that event one has $|J_t| \leq s N_\Gamma$, so one obtains a computational complexity that is reduced by a factor of r. This yields a computational complexity in

$$\mathcal{O}\left(d\,r\,(\max(s, N_\Gamma)\log(s\,N_\Gamma) + d\,s\,N_\Gamma)\log(s\,N_\Gamma/\gamma_A)\right). \tag{14}$$

A similar argument shows that with high probability, one also observes a slight improvement of the sample complexity in the sense that the logarithmic term will no longer depend on r.

3.2.3 Choosing the parameters in Algorithm 3 for Algorithm 4 in the role of Algorithm A

Up to now, we have not discussed how to choose the parameters r, γ_A, and γ. Following the probability estimates from [26, Lemma 4.4] motivates the choice $r := \lceil 2 s \log\left(\frac{3 d s}{\delta}\right)\rceil$. This choice ensures that each of the non-zero Fourier coefficients $\hat{p}_{\mathbf{k}}$ of p as in (12) can be detected in Step 1 as well as Step 2 with high probability. The key idea of [26] that we are also exploiting here, is that fixing the last $d - t$ components of all vectors in the sampling set to the same fixed values, allows for the estimation of a certain projection of the Fourier coefficients. Repeating this process r times for different choices of the $d - t$ components ensures that with a probability of at least $1 - \frac{\delta}{3 d}$, each of the active Fourier coefficients is projected to some coefficient that is not less than θ at least once among the different projections. Since we take the union of the frequency sets $\tilde{J}_{t,i}$, $i = 1, \ldots, \tilde{r}$, in Step 2d, it is sufficient to only regard the frequencies that belong to coefficients of at least θ in modulus in Step 2c. Clearly, Algorithm 2 will compute all projections of the active Fourier coefficients, and in particular those that are at least θ, with failure probabilities estimated in Corollary 2. Accordingly, the failure probabilities for detecting all the frequencies with coefficients not less than θ are also bounded by these estimates and as a consequence, we can apply the estimates on the failure probabilities in Corollary 2 to Algorithm 4 since $\tilde{s} \geq s$ is fulfilled.

That is why we set $\gamma_A := \delta/(3 d r)$ in Step 2, which entails that for each fixed t and fixed i, Algorithm 4 has a failure probability of at most $\delta/(3 d r)$ for detecting all frequencies belonging to Fourier coefficients of at least θ. Similarly, we fix $\gamma := \delta/(3 d)$ for Step 3.

In analogy to [26, Theorem 4.6], the total failure probability can now be estimated via a union bound over the different parts. We obtain failure probabilities of $\delta/3$ for Step 1, $(d-2)\delta/(3 d) + ((d-2)r + 1)\delta/(3 d r)$ for Step 2, and $\delta/(3 d)$ for Step 3. Accordingly, the total failure probability is less than δ.

The resulting sample complexities and computational complexities for this choice of parameters are summarized in Table 4. For Step 2 we list a worst case (w.c.) bound as well as a bound that hold with a high probability of at least $1 - \delta$ (w.h.p.).

For the computational complexity of Step 2, we obtain simpler expressions by bounding the worst case estimate via

$$d\,s \left(\log \frac{d\,s}{\delta} \right) \left(\max(s, N_\Gamma) \log(s\,N_\Gamma) + d\,s^2\,N_\Gamma \, \log \frac{d\,s}{\delta} \right) \log \frac{d\,s\,N_\Gamma}{\delta}$$

$$\lesssim d^2 s^3 N_\Gamma \log^3 \frac{d\,s\,N_\Gamma}{\delta}$$

and the high probability estimate via

$$d\,s \left(\log \frac{d\,s}{\delta} \right) (\max(s, N_\Gamma) \log(s\,N_\Gamma) + d\,s\,N_\Gamma) \log \frac{d\,s\,N_\Gamma}{\delta}$$

$$\lesssim d^2 s^2 N_\Gamma \log^3 \frac{d\,s\,N_\Gamma}{\delta}.$$

4 Numerical results

In this section, we validate our theoretical findings by numerical experiments. We first demonstrate the feasibility for unstructured candidate sets by choosing both the candidate set and the set of active frequencies at random. Second, as an example of highly structured candidate sets, we investigate the hyperbolic cross. Lastly we study the performance of our method in the context of a dimension-incremental sparse FFT in high dimensions.

4.1 Sparse FFT for arbitrary candidate sets as introduced in Section 2

We start by validating our approach in the framework of Section 2. That is, we consider different frequency candidate sets $\Gamma \subset \mathbb{Z}^d$, $|\Gamma| < \infty$, and apply our method to recover trigonometric polynomials p, cf. (3), with frequencies supported on small subsets $I \subset \Gamma$.

Table 4 Sample complexities and computational complexities for the different steps of Algorithm 3, where the efficient identification Algorithm 4 is used in Step 2 and $r := \left\lceil 2\,s\,\ln \frac{3\,d\,s}{\delta} \right\rceil$ is chosen in line with [26, Theorem 4.6]; Step 3 is realized via the multiple rank-1 lattice approach of [26, Algorithm 1]

	Sample complexity	Computational complexity
Step 1	$d\,s\,N_\Gamma \, \log \frac{d\,s}{\delta}$	$d\,s\,N_\Gamma \, \log^2 \frac{d\,s\,N_\Gamma}{\delta}$
Step 2 (w.h.p.)	$d\,s\,\max(s, N_\Gamma) \log^2 \frac{d\,s\,N_\Gamma}{\delta}$	$d^2 s^2 N_\Gamma \log^3 \frac{d\,s\,N_\Gamma}{\delta}$
Step 2 (w.c.)	$d\,s\,\max(s, N_\Gamma) \log^2 \frac{d\,s\,N_\Gamma}{\delta}$	$d^2 s^3 N_\Gamma \log^3 \frac{d\,s\,N_\Gamma}{\delta}$
Step 3	$\max(s, N_\Gamma) \log(d\,s/\delta)$	$\max(s, N_\Gamma) \log(d\,s/\delta)(d + \log(s\,N_\Gamma))$

4.1.1 Identifying potential false positives and potential false negatives

As explained above, a main advantage of our recovery guarantees as compared to other results with similar sample complexity is that we do not assume a random signal model, but we show recovery with high probability for an arbitrary signal. Even stronger, we only analyze aliasing properties of the support, cf. formula (5) and the related discussion in Section 2.1. So for fixed support I we guarantee that with high probability all signals supported on I can be recovered.

In our numerical simulations, we also aim to illustrate this strong property. For that, we employ a worst case measure for the support detection. For a given support I, we compute all the *potential* false positives (PFP) and *potential* false negatives (PFN), that is, all index vectors $\mathbf{j}$ that arise as an alias of some $\mathbf{k} \in I$. When $\mathbf{j} \in I$, a cancellation of the true coefficient and the aliased coefficient can have the effect that the associated frequency is not detected as active. For $\mathbf{j} \notin I$, the aliased coefficient will result in an unjustified detection of the coefficient. Especially the potential false negatives will only be realized for very specific choices of coefficients (that would be unlikely under a random model), and it is not clear if there actually are coefficients that realize multiple of them simultaneously. For this reason, the bounds on the success rates we empirically compute are somewhat conservative, but they certainly form a lower bound for the true rates.

More precisely, our empirical evaluation is based on the observation that to decide if a given frequency $\mathbf{k}$ is in I, our method considers a set of measurements, each of which constitutes the sum over the true (potentially zero if $\mathbf{k} \notin I$) coefficient corresponding to $\mathbf{k}$ and all the aliased coefficients with respect to some random rank-1 lattice, and classifies $\mathbf{k}$ as a member of I if fewer than half of these measurements are zero. Thus a sufficient condition to prevent that $\mathbf{k}$ is wrongly classified as a member or not a member of I is that for more than half of these random lattices, $\mathbf{k}$ aliases to no other $\mathbf{j} \in I$.

To count how many $\mathbf{j} \in I$ the frequency $\mathbf{k}$ aliases to, we use the following trick: We apply the method to the auxiliary polynomial

$$p(\mathbf{x}) = \sum_{\mathbf{k} \in I} e^{2\pi i \mathbf{k} \cdot \mathbf{x}}. \tag{15}$$

As all the coefficients indexed by I are 1, the aforementioned sum over the aliased coefficients for a realization of the random lattice is nothing but a counting measure applied to the intersection of I and the set of indices that $\mathbf{k}$ aliases to. Consequently, when no aliasing happens, the sum will be 0 for $\mathbf{k} \notin I$ and 1 for $\mathbf{k} \in I$. Both of these values are the smallest possible, so if the median $\check{p}_{\mathbf{k}}$ in (11) takes this value, aliasing happens in fewer than half of the cases, as desired, and hence both false positives and false negatives are excluded for any choice of values of the active coefficients. Otherwise, we count $\mathbf{k}$ as a potential false positive (PFP) or potential false negative (PFN), respectively. This method to empirically identify the potential false detections is illustrated in Table 5.

Remark 10 On the one hand, the empirical detection of the PFNs and PFPs via Algorithm 2 applied to the auxiliary trigonometric polynomial (15) will always give

Table 5 Empirical detection of potential false negatives (PFN) and potential false positives (PFP) for known Γ and I via Algorithm 2 applied to the auxiliary trigonometric polynomial given in (15)

		Coefficient recovered by Algorithm 2		
		$\check{p}_{\mathbf{k}} = 0$	$\check{p}_{\mathbf{k}} = 1$	$\check{p}_{\mathbf{k}} \geq 2$
true Fourier coefficient of p	$\hat{p}_{\mathbf{k}} = 1$ (i.e., $\mathbf{k} \in I$)	Not possible	not a PFN	PFN
	$\hat{p}_{\mathbf{k}} = 0$ (i.e., $\mathbf{k} \in \Gamma \setminus I$)	Not a PFP	PFP	

correct results. On the other hand, when additionally using the postprocessing discussed in Remark 9, we may obtain "wrong" counts if there are any potential false negatives present. The reason for this is that the PFN frequencies will be in the output $\tilde{I}$ of Algorithm 2 and consequently, taken into consideration by the postprocessing discussed in Remark 9. Whenever the set of used rank-1 lattices provides a spatial discretization of the space $\Pi_{\tilde{I}}$ of trigonometric polynomials, all PFNs will be filtered out. In reality, however, it may happen that a potential false negative does not appear in the output $\tilde{I}$ at all if aliasing Fourier coefficients cancel out each other, and consequently, the postprocessing could possibly yield incorrect Fourier coefficients and filter out energetic frequencies. As the cancellation strongly depends on the used rank-1 lattices and the Fourier coefficients of the function under consideration, the cancellation should be unlikely in practice.

4.1.2 Random frequency sets and candidate sets in three dimensions

Here, we investigate the accuracy of multiple random rank-1 lattice sampling for the exactly sparse case. We consider the reconstruction of the Fourier coefficients $\hat{p}_{\mathbf{k}}$ of three-variate trigonometric polynomials p of sparsity $|I| = 1000$. For each $L \in \{9, 11, 13, \ldots, 37\}$, we fix a multiple random rank-1 lattice configuration $\Lambda(\mathbf{z}_1, M_1, \ldots, \mathbf{z}_L, M_L)$, where each single rank-1 lattice is of size $M_\ell = 10\,331 > 10.33 \cdot 1000$. Now we repeat the following computations 1 000 times. We choose an index set of possible frequencies $\Gamma \subset \{-1000, -999, \ldots, 1000\}^3$, $|\Gamma| = 10^7$, and the index set of active frequencies $I \subset \Gamma$, $|I| = 1000$, both uniformly at random. For these choices of Γ and I we identify potential false positives (PFP) and potential false negatives (PFN) as explained in Section 4.1.1 and illustrated in Table 5.

In Fig. 1, we plot the success and failure rates for these experiments as solid lines and filled circles for various choices for the number L of rank-1 lattices used in the configuration. Here we count an experiment as a success when no PFPs and no PFNs are identified.

To put these success rate into perspective, recall that by Corollary 2, the probability that Algorithm 2 will correctly determine a trigonometric polynomial using the multiple random rank-1 lattice construction is bounded from below by $\max(0, 1 - 10^7 \cdot 9.331^{-8.331/41.324\,L})$. This number is positive for odd $L \geq 37$; for $L = 41$ one already obtains a lower bound on the success probability of 0.90. As expected some estimates in the proof of the corollary are not tight. On the one hand, for a similar target failure rate, the number L of used rank-1 lattices in the numerical tests is
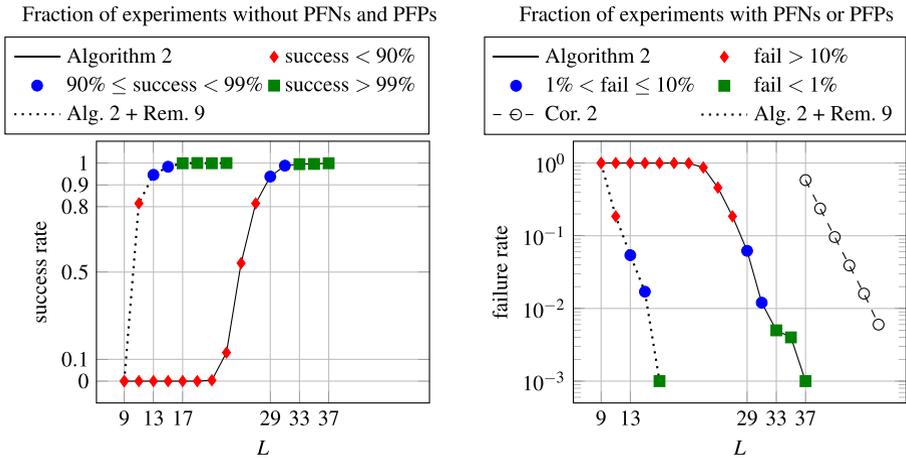
Fraction of experiments without PFNs and PFPs   Fraction of experiments with PFNs or PFPs

Fig. 1 Success rate and failure rate with respect to the number L of used rank-1 lattices for random frequency sets I and Γ, $|I| = 1000$ and $|\Gamma| = 10^7$; solid lines correspond to Algorithm 2, the dotted lines correspond to Algorithm 2 with postprocessing as discussed in Remark 9, and the dashed line shows the theoretical bounds from Corollary 2

lower by approximately one third compared to the corresponding theoretical bound in Corollary 2. On the other hand, for $L = 37$ rank-1 lattices, the empirical failure rate behaves distinctly better with a value of only 0.001 compared to the higher theoretical bound by Corollary 2 of 0.583. To illustrate this difference, we include our bound for the failure rate given by Corollary 2 for comparison as dashed line and unfilled circles. We observe that despite large constants, the exponential decay of both theory and experiments match well.

Furthermore, we observe that the empirical failure rate is less than 0.01 for $L \geq 33$. For $L = 33$, we require 340 891 samples, which is only $\approx 1/29$ of the samples required for a full discrete Fourier transform on Γ and only $\approx 1/23\,500$ of the samples of a fast Fourier transform on $\{-1000, -999, \ldots, 1000\}^3$. Another remarkable observation is that the success probability increases from less than 0.1 to more than 0.9 within a small range of L, which also reflects the logarithmic dependence of L on the failure probability.

In Fig. 2, we consider the aliasing effects in more detail. For $L \geq 29$, we observe no potential false negatives, which means that the output $\tilde{I}$ of Algorithm 2 contains all frequencies that belong to non-zero Fourier coefficients, i.e., $I \subset \tilde{I}$. However, for $20 \leq L \leq 37$, we still observe that $\tilde{I} \neq I$ in some test runs due to a small number of potential false positives. For our goal of identifying the active frequencies and their associated coefficients, this is much less severe than a false negative would be.

This observation directly yields a refined definition of success. Namely, we consider Algorithm 2 to be successful if no potential false negative and at most a predefined number of potential false positives are observed. The lower plots in Fig. 2 visualize this refined success rate when this predefined number is chosen to be 10 and 100, respectively. We observe success in more than 99% of the test runs for $L \geq 23$ and $L \geq 17$ for at most ten potential false positives and for at most 100, respectively.
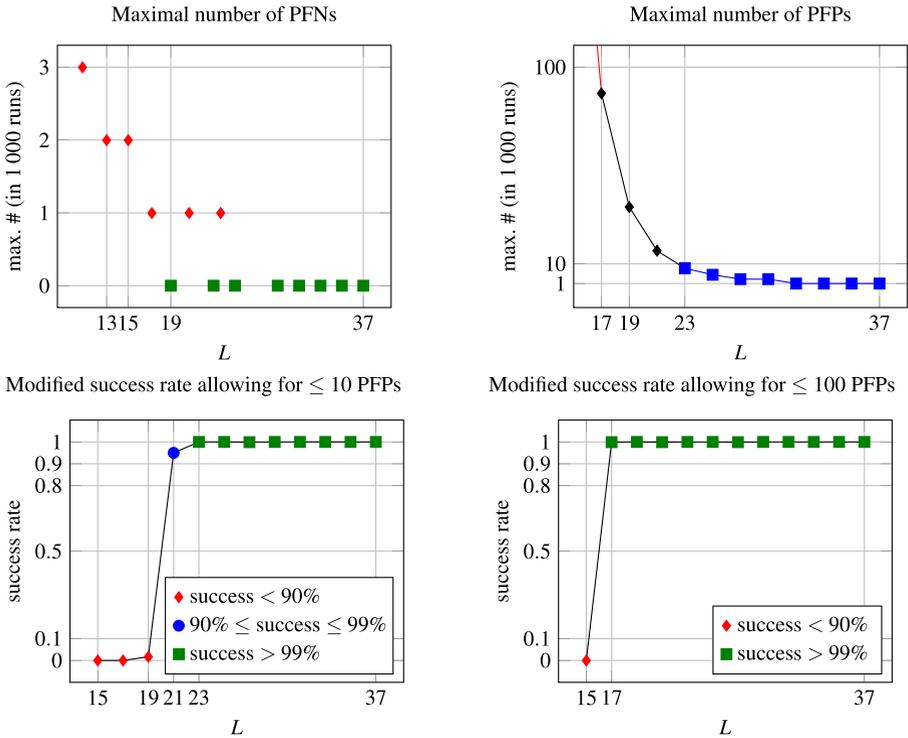
Fig. 2 Additional performance measures for the random frequency sets I and Γ from Fig. 1 for Algorithm 2; *First row:* Maximal numbers of aliasing frequencies within I (PFNs) and aliasing frequencies in $\Gamma \setminus I$ with frequencies within I (PFPs); *Second row:* Modified success rates with no potential false negatives, but some potential false positives allowed

Furthermore, we note that the success rate seems to exhibit a sharp phase transition, increasing from less than 0.02 to more than 0.95 from one odd L to the next.

Without providing any further details, we point out that even for significantly smaller $L \geq 17$, we observe at most a small number of false negatives and at most a small number of false positives, which could be a good reason for iterative applications of Algorithm 2 with relatively small numbers L of rank-1 lattices and successively reduced frequency support I.

As an alternative to this idea, one can additionally use the available rank-1 lattice information to filter the false positives in a postprocessing step as discussed in Remark 9. In accordance with Remark 10, we only apply the postprocessing if there are no potential false negatives. The corresponding success rates are plotted in Fig. 1 as dotted lines and filled circles. We observe a distinct improvement having success rates of more than 99% already for $L \geq 17$.

As briefly indicated in Remark 9, one could also determine the Fourier coefficients by solving the linear system arising from the restriction to all identified frequencies (including false negatives). Already for minor oversampling, this system will often

have a unique solution—with zero coefficients associated to the false positives. Naturally this approach will require that there are no false negatives and only a small number of false positives.

4.1.3 Weighted and unweighted hyperbolic crosses as frequency set and candidate set in eight dimensions

In the following, we investigate the detection accuracy for deterministic, structured frequency index sets. To this end, we fix an unweighted eight-dimensional hyperbolic cross

$$\Gamma := \left\{ \mathbf{k} \in \mathbb{Z}^8 : \prod_{t=1}^{8} \max(1, |k_t|) \leq 32 \right\} \subset [-32, 32]^8,$$

$|\Gamma| = 10\,665\,297$, as a frequency candidate set. Furthermore, we fix the set of active frequencies $I \subset \Gamma$ of cardinality $|I| = 1069$ given by

$$I := \left\{ \mathbf{k} \in \mathbb{Z}^8 : \prod_{t=1}^{8} \max(1, t^{1.08}|k_t|) \leq 32 \right\} \subset \Gamma,$$

which is an eight-dimensional weighted hyperbolic cross.

For each $L \in \{9, 11, 13, \ldots, 37\}$, we repeat the following procedure 1000 times: We draw a multiple random rank-1 lattice configuration $\Lambda(\mathbf{z}_1, M_1, \ldots, \mathbf{z}_L, M_L)$, where each single rank-1 lattice is of size $M_\ell = 11\,047 > 10.33 \cdot 1069$, and we identify potential false positives (PFP) and potential false negatives (PFN) using Algorithm 2 as explained in Section 4.1.1 and illustrated in Table 5.

The plots in Figs. 3 and 4 are analogous to those in Section 4.1.2. We also observe a similar behavior of the success rates and the failure rates, in line with the decay rates predicted by Corollary 2. For $L = 31$ – this corresponds to 342 427 samples – we achieve a success rate of more than 0.99, if we do not allow for any PFNs and PFPs. For the modified notion of success with no PFNs, but 10 or 100 PFPs allowed, a success rate of more than 0.99 can be achieved already for $L = 23$ or $L = 19$, respectively.

As before, if one additionally uses the available rank-1 lattice information, cf. Remarks 9 and 10, then we observe success rates of more than 99% already for $L \geq 15$, see the dotted lines and filled circles in Fig. 3.

4.1.4 Weighted hyperbolic crosses as frequency set and candidate set in 40 dimensions

In analogy to the setting in Section 4.1.3, we now consider a higher dimensional example, where the candidate set

$$\Gamma := \left\{ \mathbf{k} \in \mathbb{Z}^{40} : \prod_{t=1}^{40} \max(1, t^{0.30311}|k_t|) \leq 32 \right\} \subset [-32, 32]^{40},$$
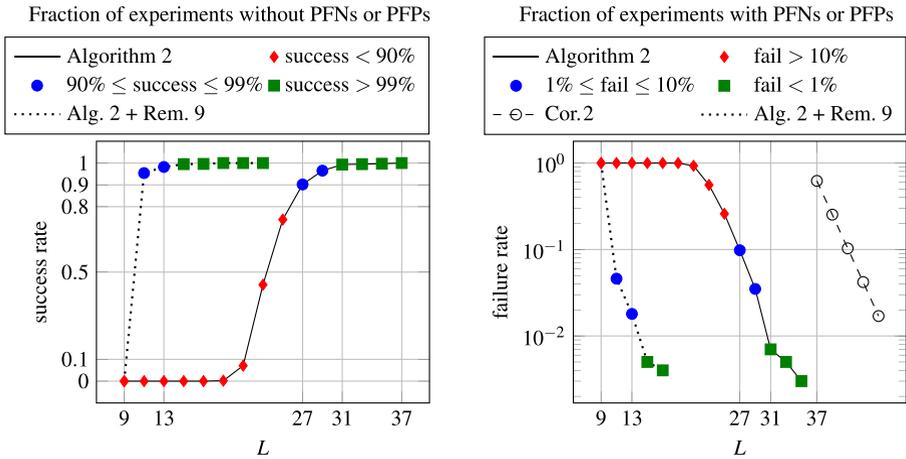
Fig. 3 Success rate and failure rate with respect to the number of used rank-1 lattices for multiple random rank-1 lattices, where $\Gamma \subset \mathbb{Z}^8$ is a symmetric hyperbolic cross and $I \subset \Gamma$ a weighted hyperbolic cross; solid lines correspond to Algorithm 2, the dotted lines correspond to Algorithm 2 with postprocessing as discussed in Remark 9, and the dashed line shows the theoretical bounds from Corollary 2
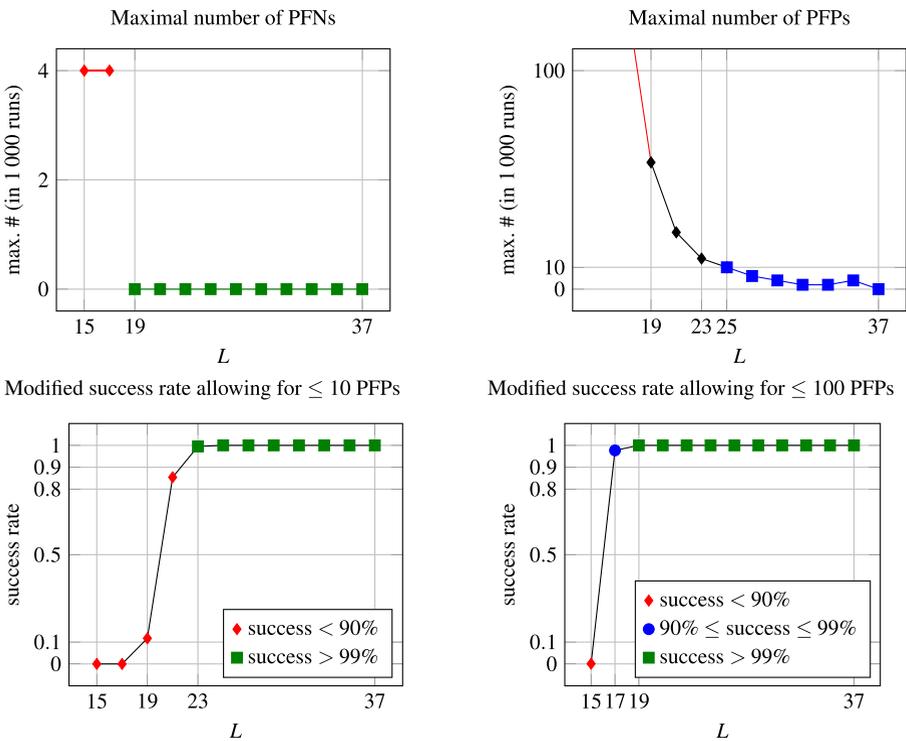

Fig. 4 Additional performance measures for Γ and I unweighted and weighted hyperbolic crosses, respectively, from Fig. 3 for Algorithm 2; *First row:* Maximal numbers of aliasing frequencies within I (PFNs) and aliasing frequencies in $\Gamma \setminus I$ with frequencies within I (PFPs); *Second row:* Modified success rates with no potential false negatives, but some potential false positives allowed

$|\Gamma| = 10\,008\,793$, is a 40-dimensional weighted hyperbolic cross of similar cardinality as before, and the set of active frequencies $I \subset \Gamma$ is given by

$$I := \left\{ \mathbf{k} \in \mathbb{Z}^{40} : \prod_{t=1}^{40} \max(1, t^{1.15}|k_t|) \le 32 \right\} \subset \Gamma,$$

which is a 40-dimensional weighted hyperbolic cross of cardinality $|I| = 1001$. We proceed exactly as in Section 4.1.3, and now, each rank-1 lattice is of size $M_\ell = 10\,343 > 10.33 \cdot 1001$, which is slightly smaller than in Section 4.1.3. In Fig. 5 (left), we show the success rates with respect to $L \in \{9, 11, 13, \ldots, 37\}$ and observe very similar results as in Fig. 3. Likewise, in Fig. 5 (right), the modified success rates correspond to those in Fig. 4 (bottom right).

These results are not surprising since we have used similar cardinalities for the active frequency set and the candidate set as in Section 4.1.3, except for the higher spatial dimension $d = 40$. This confirms that the proven performance of Algorithm 2 does not depend on d, cf. Corollary 2.

4.2 Dimension-incremental sparse FFT

We continue by numerically exploring the dimension-incremental sparse FFT method of [26, 33], and Algorithm 3 with our multiple random rank-1 lattice sampling method described in Algorithm 4 in the role of Algorithm A.

4.2.1 Random sparse trigonometric polynomial

As in [33, Section 3.1] and [26, Section 3.1], we construct random multivariate trigonometric polynomials p of the form (3) with frequencies supported in the cube
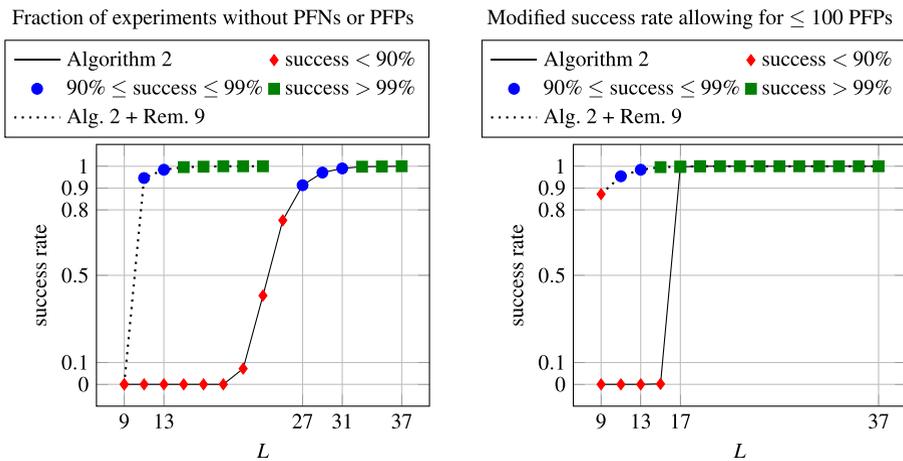

Fig. 5 Success rate with respect to the number of used rank-1 lattices for multiple random rank-1 lattices, where $\Gamma \subset \mathbb{Z}^{40}$ is a weighted hyperbolic cross and $I \subset \Gamma$ a smaller weighted hyperbolic cross; solid lines correspond to Algorithm 2, and the dotted lines correspond to Algorithm 2 with postprocessing as discussed in Remark 9

$\hat{G}_N^d := [-N, N]^d \cap \mathbb{Z}^d$. For this, we choose $|I|$ frequencies $\mathbf{k} \in \hat{G}_N^d$ uniformly at random and draw the corresponding Fourier coefficients $\hat{p}_{\mathbf{k}} \in [-1, 1) + [-1, 1)i$, $|\hat{p}_{\mathbf{k}}| \geq 10^{-6}$, uniformly at random for all $\mathbf{k} \in I = \text{supp}\,\hat{p}$. For the reconstruction of the trigonometric polynomials p, we only assume that the search domain $\Gamma := \hat{G}_N^d \supset I$.

We set the expansion parameter $N := 32$, which corresponds to $N_\Gamma = 64$. Now, we compare the results of single reconstructing rank-1 lattice sampling from [26, Algorithm 5], multiple reconstructing rank-1 lattice sampling from [26, Algorithm 4], and Algorithm 3 combined with our multiple random rank-1 lattice approach, Algorithm 4. Note that [26, Algorithm 5] also follows the framework of Algorithm 3, with a single reconstructing rank-1 lattice sampling method used in Step 3 and also taking the role of Algorithm A. Likewise, [26, Algorithm 4] corresponds to Algorithm 3 with multiple reconstructing rank-1 lattice sampling taking the role of Algorithm A. For all of these approaches, we choose the absolute threshold parameter $\theta := 10^{-12}$, the number of detection iterations $r := 1$, and the sparsity parameter $s := |I|$.

For the combination of Algorithm 3 and Algorithm 4, we work with local sparsity parameter of $s_{\text{local}} := 2|I|$ and parameter $\delta := 0.9$. Motivated by the numerical results in Section 4.1 and since we can tolerate up to $|I|$ false positives for each invocation of Algorithm 4, we distinctly reduce the number L of random rank-1 lattices used in Algorithm 4 as compared to the choice postulated by Corollary 2. We only use

$$L := \min\left\{ n \in 2\mathbb{N}+1 : n \geq \frac{1}{4}\,\frac{4c}{(c-2)\,\ln(c-1)}\,(\ln|J_t| - \ln\delta) \right\} \text{ with } c := 10.33, \quad (16)$$

which is a reduction of about 3/4 as compared to Corollary 2. Obviously, this choice of the parameter L is not covered by our theory. However, the numerical tests below are not affected, which impressively corroborates that the worst-case scenarios we used for the theoretical estimates hardly ever occur simultaneously. At this point, we stress on the fact that [26, Algorithm 4] and [26, Algorithm 5] use numerically determined spatial discretizations for the candidate sets J_t in Step 2, where the approaches for constructing these spatial discretizations are also optimized with respect to the number of used sampling values. Thus, both algorithms also already exploit the potential gaps between theory and practice.

For sparsity $|I| \in \{1000, 10\,000, 100\,000\}$, we run tests for spatial dimension $d \in \{5, 10, 15, 20, 25, 30\}$ applying [26, Algorithm 5], [26, Algorithm 4], and Algorithm 3 using Algorithm 4. We only omit the case of sparsity $|I| = 100\,000$ for [26, Algorithm 5] since this would have required quite a large number of samples and very long runtimes. All tests are repeated 10 times with newly chosen frequencies $\mathbf{k} \in \Gamma$ and Fourier coefficients $\hat{p}_{\mathbf{k}} \in \mathbb{C}$. Then, for the 10 repetitions, we determine the maximum of the total number of samples. The numerical results are displayed in Table 6. We also determine the relative ℓ_2-errors of the Fourier coefficients and observe that all errors are near machine precision (below $2 \cdot 10^{-15}$). In particular, all frequencies in all test runs are successfully recovered.

For [26, Algorithm 5] using single reconstructing rank-1 lattices, the results are shown in the third column of Table 6. Moreover, the results for [26, Algorithm 4] using multiple reconstructing rank-1 lattices are presented in the fourth column. We observe that we require significantly fewer samples than if we had used a d-dimensional FFT on a full grid, which would require $|\hat{G}^5_{32}| = 65^5 = 1\,160\,290\,625$ already in the 5-dimensional case. Additionally, for sparsity $|I| := 10\,000$, [26, Algorithm 4] using multiple reconstructing rank-1 lattices required only a fraction of approximately between 1/83 and 1/11 of the samples as compared to [26, Algorithm 5] using single rank-1 lattices.

The results for Algorithm 3 combined with the multiple random rank-1 lattice approach of Algorithm 4 are presented in the fifth column of Table 6. Here, we require only a fraction between 1/179 and 1/21 of the samples as compared to [26, Algorithm 5]. Moreover, we achieve at least a similar number of samples as compared to [26, Algorithm 4] and are able to reduce the samples by up to 2/3.

Furthermore, we also re-ran all the test for an increased expansion parameter $N = 256$, which corresponds to $N_\Gamma = 512$. The numerical results we obtained are shown

Table 6 Results for random sparse trigonometric polynomials applying [26, Algorithm 5], [26, Algorithm 4], and Algorithm 3 using Algorithm 4, when considering frequencies within the search domain $\Gamma = \hat{G}^d_{32}$; the detection was successful in all considered cases and the relative ℓ_2-errors of the Fourier coefficients near machine precision (below $2 \cdot 10^{-15}$)

d	$|I|$	[26, Algorithm 5] (single reco. R1L) Max. #samples	[26, Algorithm 4] (multiple reco. R1L) Max. #samples	Algorithm 3 using Algorithm 4 (random R1L) Max. #samples
5	1 000	6 260 605	581 881	289 914
10	1 000	20 848 685	1 589 349	649 756
15	1 000	35 937 525	2 599 029	1 011 666
20	1 000	52 361 205	3 609 753	1 373 810
25	1 000	67 164 695	4 621 205	1 735 486
30	1 000	80 660 385	5 644 059	2 097 396
5	10 000	190 618 285	4 648 335	3 321 330
10	10 000	1 081 274 675	15 186 447	7 990 386
15	10 000	1 969 412 575	25 662 189	12 639 840
20	10 000	2 935 663 575	36 161 887	17 308 866
25	10 000	3 837 073 825	46 681 103	21 958 610
30	10 000	4 771 398 905	57 203 659	26 567 030
5	100 000	–	33 428 113	34 007 204
10	100 000	–	143 681 689	80 494 280
15	100 000	–	250 232 085	126 776 914
20	100 000	–	356 857 499	173 262 642
25	100 000	–	463 174 925	219 749 054
30	100 000	–	569 711 277	266 435 166

in Table 7; the columns have the same meaning as in Table 6. For this scenario, we observe that Algorithm 3 using the multiple random rank-1 lattice approach from Algorithm 4 requires only a fraction between 1/17 and 1/10 of the number of samples of [26, Algorithm 4] using multiple reconstructing rank-1 lattices as well as only a fraction between 1/1590 and 1/217 of the number of samples of [26, Algorithm 5]. The relative ℓ_2-errors are still near machine precision for all the settings and methods considered (below $2 \cdot 10^{-15}$).

Comparing the results in Table 7 for expansion parameter $N = 256$ with the results in Table 6 for $N = 32$, we observe a significantly higher reduction in the number of samples for the combination of Algorithm 3 and Algorithm 4 in the case $N = 256$. This observation confirms the theoretical results that we are able to reduce the factor N_Γ in the sample complexity of [26, Algorithm 5] and [26, Algorithm 4] to $\log N_\Gamma$ by employing Algorithm 4, see also Table 1.

Table 7 Results for random sparse trigonometric polynomials applying [26, Algorithm 5], [26, Algorithm 4], and Algorithm 3 using Algorithm 4, when considering frequencies within the search domain $\Gamma = \hat{G}_{256}^d$; the detection was successful in all considered cases and the relative ℓ_2-errors of the Fourier coefficients near machine precision (below $2 \cdot 10^{-15}$)

		[26, Algorithm 5] (single reco. R1L)	[26, Algorithm 4] (multiple reco. R1L)	Algorithm 3 using Algorithm 4 (random R1L)
d	$\lvert I \rvert$	Max. #samples	Max. #samples	Max. #samples
5	1 000	81 021 568	4 556 111	372 790
10	1 000	218 672 495	11 545 999	842 668
15	1 000	345 163 622	18 382 277	1 309 654
20	1 000	482 567 506	25 400 061	1 775 348
25	1 000	614 584 563	32 241 219	2 240 656
30	1 000	785 132 009	39 265 929	2 712 170
5	10 000	3 192 096 843	49 697 515	3 745 910
10	10 000	11 402 137 557	132 626 899	8 419 596
15	10 000	19 075 479 723	215 457 509	13 069 812
20	10 000	27 897 681 285	298 481 637	17 741 814
25	10 000	35 745 041 259	381 485 827	22 373 004
30	10 000	43 029 740 781	464 049 693	27 023 214
5	100 000	–	454 666 789	44 144 134
10	100 000	–	1 294 938 567	100 961 292
15	100 000	–	2 134 772 195	157 779 374
20	100 000	–	2 974 838 029	214 596 532
25	100 000	–	3 815 598 519	271 616 236
30	100 000	–	4 655 793 605	328 233 342

4.2.2 Random sparse trigonometric polynomial with Gaussian noise

Similarly to [26, Section 5.2], we test the robustness to noisy samples and construct 10-dimensional random multivariate trigonometric polynomials p of the form (3) with $|I| = 1\,000$ frequencies chosen uniformly at random from the cube $\Gamma = \hat{G}_{256}^{10} = [-256, 256]^{10} \cap \mathbb{Z}^{10}$. We draw the corresponding Fourier coefficients $\hat{p}_{\mathbf{k}} \in [-1, 1) + [-1, 1)\mathrm{i}$, $|\hat{p}_{\mathbf{k}}| \geq 10^{-3}$, uniformly at random for all $\mathbf{k} \in I = \operatorname{supp} \hat{p}$. Please note, that this is a harder task than in [26, Section 5.2] where all Fourier coefficients have an absolute value of one.

As described in [26, Section 5.2], we apply additive complex white Gaussian noise $\eta_j \in \mathbb{C}$ with zero mean and standard deviation σ on the samples of p, i.e., we use measurements of the form $p(\mathbf{x}_j) + \eta_j$ and generate the noise by $\eta_j := \sigma/\sqrt{2}\,(\eta_{1,j} + \mathrm{i}\,\eta_{2,j})$ where $\eta_{1,j}, \eta_{2,j}$ are independent standard normal distributed. Note that the signal-to-noise ratio (SNR) can be approximately computed by

$$\text{SNR} \approx \frac{\sum_{j=0}^{M-1} |p(\mathbf{x}_j)|^2/M}{\sum_{j=0}^{M-1} |\eta_j|^2/M} \approx \frac{\sum_{\mathbf{k} \in \operatorname{supp} \hat{p}} |\hat{p}_{\mathbf{k}}|^2}{\sigma^2}$$

and this leads to the choice $\sigma := \sqrt{\sum_{\mathbf{k} \in \operatorname{supp} \hat{p}} |\hat{p}_{\mathbf{k}}|^2}/\sqrt{\text{SNR}}$ for a desired SNR value. Moreover, the SNR is often measured in the logarithmic decibel scale (dB), $\text{SNR}_{\text{dB}} = 10 \log_{10} \text{SNR}$ and $\text{SNR} = 10^{\text{SNR}_{\text{dB}}/10}$, i.e., a linear $\text{SNR} = 10^6$ corresponds to a logarithmic $\text{SNR}_{\text{dB}} = 60$ and $\text{SNR} = 1000$ corresponds to $\text{SNR}_{\text{dB}} = 30$. Here, we run tests with $\text{SNR}_{\text{dB}} \in \{80, 70, 60, 50, 40, 30, 20, 10\}$.

As in Section 4.2.1, we compare single reconstructing, multiple reconstructing, and random rank-1 lattices. Since we use noisy samples, we increase the number of detection iterations to $r := 5$ and set the local sparsity parameter $s_{\text{local}} := 2|I|$. All tests are repeated 100 times with newly randomly chosen frequencies and Fourier coefficients. Then, for the 100 repetitions, we determine the maximum of the total number of samples and the success rate, where a single test is defined to be successful when all 1000 frequencies are correctly detected.

The numerical results are shown in Table 8. When using [26, Algorithm 5] (single reconstructing rank-1 lattices), cf. the second and third column, up to approximately 1.2 billion samples were required, and a success rate of 1.00 was observed for SNR_{dB} values between 30 and 80. This means for $\text{SNR}_{\text{dB}} \geq 30$, all 1000 frequencies were correctly detected in each of the 6×100 test runs. For $\text{SNR}_{\text{dB}} = 20$ and 10, this was the case in 94 and 30 of the 100 test runs, respectively.

When applying [26, Algorithm 4] (multiple reconstructing rank-1 lattices), the results initially looked considerably worse, achieving a success rate of 0.98 for $\text{SNR}_{\text{dB}} = 80$, 0.83 for $\text{SNR}_{\text{dB}} = 70$ as well as zero success for $\text{SNR}_{\text{dB}} = 50$ and below. Compared to [26, Section 5.2], this behavior might be surprising at a first glance, but it is caused by the different choice of Fourier coefficients, which makes the reconstruction problem harder, and the direct method used for the reconstruction of (intermediate) Fourier coefficients, cf. [24, Algorithm 6], which is susceptible to noise. Combining [24, Algorithm 6] with a small number of conjugate gradient

Table 8 Results for random sparse trigonometric polynomials with noise over 100 test runs applying [26, Algorithm 5], a modified version of [26, Algorithm 4], and Algorithm 3 using Algorithm 4, when considering frequencies within the search domain $\Gamma = \hat{G}_{256}^d$ and using $r := 5$ detection iterations, sparsity $s = 1\,000$, $d = 10$ as well as setting $s_{\text{local}} := 2s$

	[26, Algorithm 5] (single reco. R1L)		modified [26, Algorithm 4] (multiple reco. R1L)		Algorithm 3 using Algorithm 4 (random R1L)	
SNR$_{\text{db}}$	Max. #samples	Success rate	Max. #samples	Success rate	Max. #samples	Success rate
80	1 158 112 377	1.00	60 420 871	1.00	3 778 790	1.00
70	1 189 386 909	1.00	60 296 017	1.00	3 778 790	1.00
60	1 194 519 987	1.00	60 377 589	1.00	3 782 868	1.00
50	1 150 247 061	1.00	60 218 043	0.96	3 774 718	0.99
40	1 181 329 731	1.00	60 380 119	0.89	3 774 718	0.96
30	1 210 924 701	1.00	60 598 701	0.39	3 778 790	0.74
20	1 177 714 107	0.94	61 063 013	0.00	3 778 790	0.10
10	1 206 745 803	0.30	62 051 137	0.00	3 774 718	0.00

(CG) iterations (up to 10 for our example), using the result of [24, Algorithm 6] as starting vector for the utilized efficient CG method, distinctly improved the performance without requiring any additional samples, see also [24, Section 5.2]. We denote this approach by "modified [26, Algorithm 4]" and give the results in the forth and fifth column of Table 8. Now, one observes a success rate of 0.96 or higher for SNR$_{\text{dB}} \geq 50$, 0.89 for SNR$_{\text{dB}} = 40$, and still 0.39 for SNR$_{\text{dB}} = 30$. At the same time, the maximum number of samples is approximately 60 million, which is 1/19 of the single rank-1 lattice approach of [26, Algorithm 5].

For Algorithm 3 using Algorithm 4 (random rank-1 lattices), introduced in this paper, the results look better in two ways compared to [26, Algorithm 4], see also the last two columns in Table 8. The maximum number of samples is only approximately 3.8 million, which is about 1/16 of [26, Algorithm 4] and 1/300 of [26, Algorithm 5]. The success rates are better than [26, Algorithm 4] and still very good compared to [26, Algorithm 5] when taking the drastically reduced number of samples into account.

4.2.3 Approximation of tensor-product function by trigonometric polynomials

In the following, we will demonstrate that our method also works for functions that are only approximately sparse. For that we consider the multivariate periodic test function $f \colon \mathbb{T}^{10} \to \mathbb{R}$,

$$f(\mathbf{x}) := \prod_{t \in \{1,3,8\}} N_2(x_t) + \prod_{t \in \{2,5,6,10\}} N_4(x_t) + \prod_{t \in \{4,7,9\}} N_6(x_t), \qquad (17)$$

from [33, Section 3.3] and [26, Section 3.3] where $N_m : \mathbb{T} \to \mathbb{R}$ is the B-Spline of order $m \in \mathbb{N}$,

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

with a constant $C_m > 0$ such that $\|N_m|L_2(\mathbb{T})\| = 1$.

Each of the three summands has infinitely many non-zero Fourier coefficients, but they are exhibiting a decay pattern described by a hyperbolic cross (different for each of the summands). Thus we expect that the function is well approximated by trigonometric polynomials, namely the one corresponding to the Fourier coefficients indexed by a union of three hyperbolic crosses, each corresponding to significant coefficients of one of the summands.

We aim to demonstrate that our method allows to efficiently find such an approximation of the function f by a multivariate trigonometric polynomial p. More precisely, we apply the dimension-incremental approaches already discussed in the previous examples to determine a frequency index set $I = I^{(1,\ldots,10)} \subset \Gamma$ and to compute approximated Fourier coefficients $\hat{\tilde{p}}_{\mathbf{k}}$, $\mathbf{k} \in I$. As explained, an adaquate choice for the resulting frequency index sets I is given by the union of three sets of frequencies corresponding to the significant coefficients of the three summands. In our example, these are a three-dimensional symmetric hyperbolic cross in the dimensions 1,3,8, a four-dimensional symmetric hyperbolic cross in the dimensions 2,5,6,10, and a three-dimensional symmetric hyperbolic cross in the dimensions 4,7,9.

All tests are performed 10 times and the relative $L_2(\mathbb{T}^{10})$ approximation error

$$\frac{\|f - \tilde{S}_I f | L_2(\mathbb{T}^{10})\|}{\|f|L_2(\mathbb{T}^{10})\|} = \frac{\sqrt{\|f|L_2(\mathbb{T}^{10})\|^2 - \sum_{\mathbf{k} \in I} |\hat{f}_{\mathbf{k}}|^2 + \sum_{\mathbf{k} \in I} |\hat{\tilde{p}}_{\mathbf{k}} - \hat{f}_{\mathbf{k}}|^2}}{\|f|L_2(\mathbb{T}^{10})\|}$$

is computed, where $p = \tilde{S}_I f := \sum_{\mathbf{k} \in I} \hat{\tilde{p}}_{\mathbf{k}} e^{2\pi i \mathbf{k} \cdot \circ}$.

We set the expansion parameter $N := 16, 32, 64$ and we use the full grids $\Gamma = \hat{G}_N^{10} = [-N, N]^{10} \cap \mathbb{Z}^{10}$ as search space. Moreover, we set the number of detection iterations $r := 5$. The used sparsity input parameters s and $s_{\text{local}} = 2s$ are specified in column 2 of Table 9. Furthermore, the results of [26, Algorithm 5] based on single reconstructing rank-1 lattices, of [26, Algorithm 4] based on multiple reconstructing rank-1 lattices, and of Algorithm 3 combined with our multiple random rank-1 lattice approach in Algorithm 4 are shown in columns 3–4, 5–6, and 7–8 of Table 9, respectively. For the combination of Algorithm 3 and 4, we set the parameter $\delta := 0.999$ and the number L of random rank-1 lattices to (16), which again corresponds to a reduction of about 3/4 compared to the theoretical predictions of Corollary 2.

The column "max. rel. L_2-error" contains the maximum of the relative $L_2(\mathbb{T}^{10})$ approximation errors $\|f - \tilde{S}_I f | L_2(\mathbb{T}^{10})\| / \|f|L_2(\mathbb{T}^{10})\|$ of the 10 test runs. The remaining columns have the same meaning as described in Section 4.2.1. We observe that for increasing sparsity parameter, the number of samples increases while the relative $L_2(\mathbb{T}^{10})$ approximation error decreases.

Moreover, we observe that [26, Algorithm 5] and Algorithm 3 combined with Algorithm 4 yield similar relative $L_2(\mathbb{T}^{10})$ approximation errors, whereas [26, Algorithm 4] produces slightly higher errors. Furthermore, Algorithm 3 combined with

Table 9 Results for function $f : \mathbb{T}^{10} \to \mathbb{R}$ from (17) when limiting the number of detected frequencies, $s_{\text{local}} := 2s$

		[26, Algorithm 5] (single reco. R1L)		[26, Algorithm 4] (multiple reco. R1L)		Algorithm 3 using Algorithm 4 (random R1L)	
N	s	Max. #samples	Max. rel. L_2-error	Max. #samples	Max. rel. L_2-error	Max. #samples	Max. rel. L_2-error
16	1 000	50 405 091	1.2e−02	8 094 293	1.3e−02	2 903 576	1.2e−02
16	2 000	128 362 707	4.3e−03	15 449 367	5.0e−03	5 813 898	4.1e−03
16	3 000	222 662 847	3.4e−03	22 285 615	4.1e−03	9 643 162	3.1e−03
32	1 000	84 241 365	1.2e−02	16 623 363	1.3e−02	2 905 176	1.2e−02
32	2 000	265 019 105	3.4e−03	32 226 797	8.8e−03	6 683 344	3.4e−03
32	3 000	565 847 035	1.7e−03	46 948 295	2.2e−03	10 637 178	1.7e−03
32	4 000	767 068 055	1.4e−03	59 623 261	1.6e−03	14 175 646	1.3e−03
64	1 000	242 940 304	1.2e−02	34 236 011	4.5e−01	3 540 792	1.2e−02
64	2 000	563 767 023	3.4e−03	66 914 731	3.5e−03	75 04 972	3.4e−03
64	3 000	946 215 129	1.6e−03	98 926 095	1.8e−03	11 272 744	1.6e−03
64	4 000	1 266 499 683	9.8e−04	130 967 911	1.1e−03	15 005 506	9.8e−04
64	5 000	2 285 094 003	7.1e−04	158 693 803	9.6e−04	18 772 634	7.0e−04
64	10 000	5 577 419 619	4.3e−04	298 886 055	5.1e−04	37 534 358	3.9e−04

Algorithm 4 requires the fewest samples, a fraction of between 1/9 and 1/2 of the number of samples required by [26, Algorithm 4] and a fraction between 1/148 and 1/17 of the number of samples required by [26, Algorithm 5].

For instance in the case $N = 64$ and $s = 5\,000$, the maximal total number of samples for [26, Algorithm 5] (computed over 10 test runs) was about 2.3 billion, for [26, Algorithm 4] about 159 million samples, and for Algorithm 3 using Algorithm 4 about 19 million samples, while achieving comparable errors.

Code availability The code is available from the homepage of the authors, e.g., https://www-user. tu-chemnitz.de/~tovo/software.php.en

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Al-Hassanieh, H., Indyk, P., Katabi, D., Price, E.: Nearly optimal sparse Fourier transform. In: Proceedings of STOC, pp. 563–577 (2012)
2. Al-Hassanieh, H., Indyk, P., Katabi, D., Price, E.: Simple and practical algorithm for sparse Fourier transform. In: Proceedings of SODA, pp. 1183–1194 (2012)
3. Blum, M., Floyd, R.W., Pratt, V., Rivest, R.L., Tarjan, R.E.: Time bounds for selection. J. Comput. Syst. Sci. Int. **7**, 448–461 (1973)
4. Candès, E., Romberg, J.: Sparsity and incoherence in compressive sampling. Inverse Probl. **23**, 969–985 (2007)
5. Candès, E., Tao, T., Romberg, J.: Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. IEEE Trans. Inform. Theory **52**, 489–509 (2006)
6. Choi, B., Christlieb, A., Wang, Y.: Multiscale High-Dimensional Sparse Fourier Algorithms for Noisy Data. Mathematics, Computation and Geometry of Data **1**, 35–58 (2021)
7. Choi, B., Christlieb, A., Wang, Y.: High-dimensional sparse Fourier algorithms. Numer. Algorithms **87**, 161–186 (2021)
8. Choi, B., Iwen, M., Krahmer, F.: Sparse Harmonic Transforms: A New Class of Sublinear-Time Algorithms for Learning Functions of Many Variables. Found. Comput. Math. **21**, 275–329 (2021)
9. Choi, B., Iwen, M., Volkmer, T.: Sparse Harmonic Transforms II: Best s-Term Approximation Guarantees for Bounded Orthonormal Product Bases in Sublinear-Time. Numerische Mathematik **148**, 293–362 (2021)
10. Christlieb, A., Lawlor, D., Wang, Y.: A multiscale sub-linear time Fourier algorithm for noisy data. Appl. Comput. Harmon. Anal. **40**, 553–574 (2016)
11. Cooley, J., Tukey, J.: An algorithm for the machine calculation of complex Fourier series. Math. Comp. **19**, 297–301 (1965)
12. Cormode, G., Muthukrishnan, S.: Combinatorial algorithms for compressed sensing. In: 2006 40th Annual Conference on Information Sciences and Systems, pp. 198–201 (2006)
13. Cuyt, A., Hou, Y., Knaepkens, F., Lee, W.S.: Sparse multidimensional exponential analysis with an application to radar imaging. SIAM J. Sci. Comput. **42**, B675–B695 (2020)
14. Döhler, M., Kunis, S., Potts, D.: Nonequispaced hyperbolic cross fast Fourier transform. SIAM J. Numer. Anal. **47**, 4415–4428 (2010)
15. Dusart, P.: Explicit estimates of some functions over primes. Ramanujan J. **45**, 227–251 (2018)
16. Gilbert, A., Muthukrishnan, S., Strauss, M.: Improved time bounds for near-optimal sparse Fourier representations. In: Proceedings of SPIE (2005)
17. Gilbert, A.C., Guha, S., Indyk, P., Muthukrishnan, S., Strauss, M.: Near-optimal sparse Fourier representations via sampling. In: Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing, STOC '02, pp. 152–161. Association for Computing Machinery, New York (2002)
18. Hoeffding, W.: Probability inequalities for sums of bounded random variables. J. Amer. Statist. Assoc. **58**, 13–30 (1963)
19. Indyk, P., Kapralov, M.: Sample-optimal Fourier sampling in any constant dimension. In: 2014 IEEE 55th Annual Symposium on Foundations of Computer Science, pp. 514–523 (2014)
20. Iwen, M.: A deterministic sub-linear time sparse Fourier algorithm via non-adaptive compressed sensing methods. In: Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 20–29 (2008)
21. Iwen, M.: Combinatorial sublinear-time Fourier algorithms. Found. Comp. Math. **10**, 303–338 (2010)

22. Iwen, M.: Improved approximation guarantees for sublinear-time Fourier algorithms. Appl. Comp. Harmonic Anal. **34**, 57–82 (2013)
23. Kämmerer, L.: High Dimensional Fast Fourier Transform Based on Rank-1 Lattice Sampling. Dissertation Universitätsverlag Chemnitz (2014)
24. Kämmerer, L.: Multiple rank-1 lattices as sampling schemes for multivariate trigonometric polynomials. J. Fourier Anal. Appl. **24**, 17–44 (2018)
25. Kämmerer, L.: Constructing spatial discretizations for sparse multivariate trigonometric polynomials that allow for a fast discrete Fourier transform. Appl. Comput. Harmon. Anal. **42**, 702–729 (2019)
26. Kämmerer, L., Potts, D., Volkmer, T.: High-dimensional sparse FFT based on sampling along multiple rank-1 lattices. Applied and Computational Harmonic Analysis 51, 225–257 (2021)
27. Kämmerer, L., Volkmer, T.: Approximation of multivariate periodic functions based on sampling along multiple rank-1 lattices. J. Approx. Theory **246**, 1–27 (2019)
28. Krahmer, F., Ward, R.: Stable and robust sampling strategies for compressive imaging. IEEE Trans. Image Proc. **23**, 612–622 (2014)
29. Lawlor, D., Wang, Y., Christlieb, A.: Adaptive sub-linear time Fourier algorithms. Adv. Adapt. Data Anal. **5**, 1350003–1/–25 (2013)
30. Morotti, L.: Explicit universal sampling sets in finite vector spaces. Appl. Comput. Harmon. Anal. **43**, 354–369 (2017)
31. Potts, D., Steidl, G., Tasche, M.: Fast Fourier transforms for nonequispaced data: A tutorial. In: Modern Sampling Theory, pp. 247–270. Springer (2001)
32. Potts, D., Tasche, M.: Parameter estimation for multivariate exponential sums. Electron. Trans. Numer. Anal. **40**, 204–224 (2013)
33. Potts, D., Volkmer, T.: Sparse high-dimensional FFT based on rank-1 lattice sampling. Appl. Comput. Harmon. Anal. **41**, 713–748 (2016)
34. Rudelson, M., Vershynin, R.: On sparse reconstruction from Fourier and Gaussian measurements. Comm. Pure Appl. Math. **61**, 1025–1045 (2008)
35. Zippel, R.: Probabilistic algorithms for sparse polynomials. In: Proceedings of the International Symposium on Symbolic and Algebraic Computation, EUROSAM '79, pp. 216–226. Springer, London (1979)

Affiliations

Lutz Kämmerer[1] · **Felix Krahmer**[2] · **Toni Volkmer**[1]

Felix Krahmer
felix.krahmer@tum.de

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

[1]   Faculty of Mathematics, Chemnitz University of Technology, 09107 Chemnitz, Germany

[2]   Department of Mathematics, Technische Universität München, 85748 Garching/Munich, Germany