High-dimensional sparse FFT based on sampling along multiple rank-1 lattices

Lutz Kämmerer[*]   Daniel Potts[†]   Toni Volkmer[‡]

The reconstruction of high-dimensional sparse signals is a challenging task in a wide range of applications. In order to deal with high-dimensional problems, efficient sparse fast Fourier transform algorithms are essential tools. The second and third authors have recently proposed a dimension-incremental approach, which only scales almost linear in the number of required sampling values and almost quadratic in the arithmetic complexity with respect to the spatial dimension d. Using reconstructing rank-1 lattices as sampling scheme, the method showed reliable reconstruction results in numerical tests but suffers from relatively large numbers of samples and arithmetic operations. Combining the preferable properties of reconstructing rank-1 lattices with small sample and arithmetic complexities, the first author developed the concept of multiple rank-1 lattices. In this paper, both concepts — dimension-incremental reconstruction and multiple rank-1 lattices — are coupled, which yields a distinctly improved high-dimensional sparse fast Fourier transform. Moreover, the resulting algorithm is analyzed in detail with respect to success probability, number of required samples, and arithmetic complexity. In comparison to single rank-1 lattices, the utilization of multiple rank-1 lattices results in a reduction in the complexities by an almost linear factor with respect to the sparsity. Various numerical tests confirm the theoretical results, the high performance, and the reliability of the proposed method.

Keywords and phrases : sparse fast Fourier transform, FFT, multivariate trigonometric polynomials, lattice rule, multiple rank-1 lattices, approximation of multivariate functions.

2000 AMS Mathematics Subject Classification : 65T, 65T40, 42A10.

[*]Chemnitz University of Technology, Faculty of Mathematics, 09107 Chemnitz, Germany
  lutz.kaemmerer@mathematik.tu-chemnitz.de, Phone:+49-371-531-37728, Fax:+49-371-531-837728
[†]Chemnitz University of Technology, Faculty of Mathematics, 09107 Chemnitz, Germany
  potts@mathematik.tu-chemnitz.de, Phone:+49-371-531-32150, Fax:+49-371-531-832150
[‡]Chemnitz University of Technology, Faculty of Mathematics, 09107 Chemnitz, Germany
  toni.volkmer@mathematik.tu-chemnitz.de, Phone:+49-371-531-39999, Fax:+49-371-531-839999

1 Introduction

Equally weighted cubature formulas, called quasi Monte-Carlo rules, are substantial tools in the field of numerical integration. One kind of such cubature formulas are so called lattice rules that allow for the very efficient numerical treatment of high-dimensional integrals. In particular rank-1 lattices are very well investigated in this field of mathematics, cf. [27, 4] for more details on this topic.

In a natural way, different researchers used lattice rules in order to approximate integrals that compute Fourier coefficients of functions, and thus, reconstruct periodic signals. This reconstruction can be done by interpolation, cf. e.g. [24, 1], as well as approximation, cf. [28, 20, 21, 22, 2]. In particular, for highly interesting types of functions, e.g. functions of dominating mixed smoothness, suitable approximations use lattices that necessarily require a relatively large number of sampling nodes in comparison to the achievable worst case approximation errors, cf. [2]. A recent idea [15] to prevent the usage of a single huge rank-1 lattice is to sample along multiple rank-1 lattices in order to reconstruct trigonometric polynomials or approximate functions. With high probability, this ansatz significantly reduces the number of required sampling values, cf. [16]. The crucial innovation is the strategy that determines rank-1 lattices at random, which is also used in a more recent paper for numerical integration, cf. [18].

In this paper, we deal with the reconstruction and approximation of multivariate periodic functions based on samples without knowing the locations of the non-zero or approximately largest Fourier coefficients. A method which allows for efficiently performing this task will be called *multivariate sparse fast Fourier transform* in the following. One aim is to exactly reconstruct the Fourier coefficients $\hat{p}_{\boldsymbol{k}}$, $\boldsymbol{k} \in I$, of an arbitrarily chosen trigonometric polynomial

$$p(\boldsymbol{x}) := \sum_{\boldsymbol{k} \in I} \hat{p}_{\boldsymbol{k}} \, \mathrm{e}^{2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}} \tag{1.1}$$

with frequencies supported on an index set $I \subset \mathbb{Z}^d$ of finite cardinality, i.e. $|I| < \infty$, from sampling values of p, where the frequency index set I is unknown. The Fourier coefficients $\hat{p}_{\boldsymbol{k}} \in \mathbb{C}$ of the trigonometric polynomial p are formally given by the Fourier transform of p,

$$\hat{p}_{\boldsymbol{k}} := \int_{\mathbb{T}^d} p(\boldsymbol{x}) \, \mathrm{e}^{-2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}} \mathrm{d}\boldsymbol{x}, \quad \boldsymbol{k} \in I. \tag{1.2}$$

Alternatively, in case of a general function $f \in L_1(\mathbb{T}^d) \cap \mathcal{C}(\mathbb{T}^d)$, we aim to determine an approximation of the (roughly) largest Fourier coefficients

$$\hat{f}_{\boldsymbol{k}} := \int_{\mathbb{T}^d} f(\boldsymbol{x}) \, \mathrm{e}^{-2\pi \mathrm{i} \boldsymbol{k} \cdot \boldsymbol{x}} \mathrm{d}\boldsymbol{x}, \quad \boldsymbol{k} \in I,$$

from sampling values of f. The critical part for both tasks is the determination of the unknown frequency index set I. Using a tensor-product approach and computing a full-dimensional fast Fourier transform (FFT) is not feasible in practice already for medium dimensions (like $d = 5$) due to the curse of dimensionality.

Various methods for multivariate sparse FFTs exist which are based on different ideas. In general these methods require a search domain $\Gamma \subset \mathbb{Z}^d$ of possible frequencies as an input parameter, which is often chosen as a full grid $\hat{G}_N^d := \{-N, -N+1, \ldots, N\}^d$ of expansion $N \in \mathbb{N}$, $|\hat{G}_N^d| = (2N+1)^d$. Usually, an additional input parameter is the number of approximately largest Fourier coefficients $\hat{f}_{\boldsymbol{k}}$ to determine or, in the case of the reconstruction of a

multivariate trigonometric polynomial p, an upper bound on the cardinality $|\operatorname{supp}\hat{p}|$ of the support in frequency domain

$$\operatorname{supp}\hat{p} := \{\boldsymbol{k} \in \mathbb{Z}^d \colon \hat{p}_{\boldsymbol{k}} \neq 0\}.$$

Next, we briefly mention some of the existing methods.

One approach, which requires a relatively small amount of samples, is applying random sampling in compressed sensing [5, 3, 6], but the number of required arithmetic operations contains the cardinality $|\Gamma|$ of the search domain Γ as a linear factor, see e.g. [6, 17, 19], and therefore typically suffers heavily from the curse of dimensionality.

In [10], a multivariate sparse FFT method was discussed which uses randomized sub-sampling and filters. The method is based on the one-dimensional versions from [9, 8, 11]. As search domain Γ, a full grid $\hat{G}_N^d$ is used and the number of required samples is $\mathcal{O}(|\operatorname{supp}\hat{p}|\log N)$ for constant dimension d and the arithmetic complexity is $\mathcal{O}(N^d \log^{\mathcal{O}(1)} N)$. We remark that the sample complexity $\mathcal{O}(|\operatorname{supp}\hat{p}|\log N)$ might contain a factor of $d^{\mathcal{O}(d)}$, cf. [10, Section IV]. This means that both complexities suffer from the curse of dimensionality.

In [12], a deterministic multivariate sparse FFT algorithm was presented, which uses the Chinese Remainder Theorem and which requires $\mathcal{O}(d^4 |\operatorname{supp}\hat{p}|^2 \log^4(dN))$ samples and arithmetic operations. This means there is neither exponential/super-exponential dependency on the dimension $d \in \mathbb{N}$ nor a dependency on a failure probability in the asymptotics of the number of samples and arithmetic operations for this method. Besides this deterministic algorithm, there also exists a randomized version which only requires $\mathcal{O}(d^4 |\operatorname{supp}\hat{p}|\log^4(dN))$ samples and arithmetic operations with high probability.

In this work, we focus on a dimension-incremental method for a high-dimensional sparse FFT published in [26]. The method adaptively constructs the frequency index set I belonging to the non-zero or approximately largest Fourier coefficients in a dimension-incremental way based on sampling values. One preferable property of this strategy is that the sampling complexity is linear in the dimension d up to logarithmic factors. Another advantage is the applicability in practice for relative high sparsities even in the case where the sampling values are perturbed by noise. Moreover, the dimension-incremental strategy provides a straightforward implementation for the (approximate) recovery of multivariate periodic signals. Known properties of the function under consideration, such as smoothness conditions as well as (possibly very large) supersets of the locations of the non-zero or approximately largest Fourier coefficients, can be directly exploited by restricting the search domain Γ of considered frequencies. In general, the dimension-incremental approach is a probabilistic method, which repeatedly performs the sampling using $r \in \mathbb{N}$ many detection iterations. Theoretical bounds on r are currently unknown and will be addressed in this paper.

One crucial challenge of the dimension-incremental method is the construction of spatial discretizations for high-dimensional sparse trigonometric polynomials that allow for an efficient reconstruction. In [26], the authors mainly used single rank-1 lattices, which worked very reliably in various numerical experiments. Besides this reliability, the main advantage of using single rank-1 lattices as spatial discretizations is the available fast Fourier transform for these sampling sets. However, the bottlenecks of this approach are the relatively expensive construction of reconstructing rank-1 lattices and, in addition, the relatively large number of sampling nodes. In detail, for sparsity $s := |I|$ or when determining s many approximate Fourier coefficients, the method requires $\mathcal{O}(d\,r^3 s^2 N)$ samples and $\mathcal{O}(d\,r^3 s^3 + d\,r^3 s^2 N \log(r\,s\,N))$ arithmetic

operations in the case $\sqrt{N} \lesssim s \lesssim N^d$, when restricting the search space Γ of considered frequency candidates to a full grid $\hat{G}_N^d$ of expansion $N \in \mathbb{N}$. In order to reduce these complexities, the paper [26] also numerically investigated alternative sampling sets that allow for efficient reconstructions of sparse signals. The authors observed in the numerical experiments, that the additionally investigated types of sampling sets do not offer the reliability and stability as obtained for single rank-1 lattices.

Again, we emphasize that the dimension-incremental idea from [26] should work with any sampling method which reliably computes Fourier coefficients in a fast way. Accordingly, the dimension-incremental sparse FFT may be improved by providing sampling sets that combine four preferable properties:

- low oversampling factors,

- stability and thus reliability,

- efficient construction methods,

- fast Fourier transform algorithms.

Exactly this combination of objectives motivated the development of the concept of reconstructing multiple rank-1 lattices in [15, 16]. Therein, a reconstruction approach for multivariate trigonometric polynomials p with known frequency index sets I based on samples along multiple rank-1 lattices is proposed. Instead of using a single rank-1 lattice for the reconstruction, which requires $\mathcal{O}(d\,|I|^3)$ arithmetic operations for the construction of a suitable reconstructing rank-1 lattice that consists of up to $\mathcal{O}(|I|^2)$ samples, a union of cleverly chosen rank-1 lattices is utilized, where the latter is named reconstructing multiple rank-1 lattice. Such a reconstructing multiple rank-1 lattice consists of $\mathcal{O}(|I|\log|I|)$ nodes with high probability under mild assumptions and, in this case, the construction only requires $\mathcal{O}(|I|(d + \log|I|)\log|I|)$ arithmetic operations, cf. [16, Corollary 3.7 and Algorithm 4].

As mentioned above, we include the multiple rank-1 lattice idea into the dimension-incremental reconstruction approach in this work. We obtain a method, see Algorithm 3, which requires $\mathcal{O}(dr^2sN\log(rsN))$ samples and $\mathcal{O}\big(d^2r^2sN\log^2(rsN)\big)$ arithmetic operations.

One of the main contributions of this work is the detailed investigation of the overall success probability of the dimension-incremental reconstruction method. For the first time, we show bounds on the number of detection iterations r when using single and multiple rank-1 lattices as sampling sets. When reconstructing a trigonometric polynomial p using Algorithm 3, we prove in Theorem 4.6 that under mild assumptions and when choosing the number of detection iterations $r := \lceil 2|\operatorname{supp}\hat{p}|(\log 3 + \log d + \log|\operatorname{supp}\hat{p}| - \log\varepsilon)\rceil$ for a given failure probability $\varepsilon \in (0,1)$ ensures that the proposed algorithm successfully detects all non-zero Fourier coefficients $\hat{p}_{\boldsymbol{k}} \neq 0$ with probability at least $1 - \varepsilon$. In this case, we require $\mathcal{O}\big(d\,|\operatorname{supp}\hat{p}|^2N\log^2(|\operatorname{supp}\hat{p}|)\,|\log\varepsilon|\big)$ samples and $\mathcal{O}\big(d^2\,|\operatorname{supp}\hat{p}|^2N\log^3(|\operatorname{supp}\hat{p}|)\,|\log\varepsilon|\big)$ arithmetic operations. In our numerical tests, we did not require the linear factor $|\operatorname{supp}\hat{p}|$ in the number of detection iterations r, but we were able to choose this parameter as a constant ≤ 10. It is an open question if a smaller bound on r can be theoretically shown under mild assumptions on the Fourier coefficients $\hat{p}_{\boldsymbol{k}}$. The key result which would need to be improved is currently based on a Markov-type inequality in Lemma 4.1, see also the discussion in Section 4.2. Still, the results in Lemma 4.1, Corollary 4.2 and Corollary 4.3 imply a distinct improvement compared to the previous ones in

[26, Lemma 2.4 and Theorem 2.5], which would yield a factor of $|\operatorname{supp}\hat{p}|^2$ in r when applying the theoretical considerations in Section 4.1 on [26, Theorem 2.5].

If the Fourier coefficients $\hat{p}_{\boldsymbol{k}}$ fulfill certain stronger properties, we obtain distinctly smaller sample and arithmetic complexities by slightly modifying the algorithms, see also Remark 3.2. As discussed in Section 4.3, if we know that the real parts of all non-zero $\hat{p}_{\boldsymbol{k}}$ have identical sign $\neq 0$ or that the imaginary parts of all non-zero $\hat{p}_{\boldsymbol{k}}$ have identical sign $\neq 0$, then we can use a modified version of Algorithm 3 with $r := 1$ detection iteration. The resulting algorithm successfully detects all non-zero Fourier coefficients $\hat{p}_{\boldsymbol{k}} \neq 0$ with probability at least $1 - \varepsilon$, requiring $\mathcal{O}\big(d\,|\operatorname{supp}\hat{p}|N\log(|\operatorname{supp}\hat{p}|\,N)\big)$ samples and $\mathcal{O}\big(d^2\,|\operatorname{supp}\hat{p}|N\log^2(|\operatorname{supp}\hat{p}|\,N)\,(\log d - \log \varepsilon)\big)$ arithmetic operations, cf. Section 4.3.

The remaining parts of this paper are structured as follows. In Section 2, we briefly review important aspects of reconstructing multiple rank-1 lattices and dimension-incremental reconstruction. The new method is discussed in Section 3. In Section 4, we develop bounds on the number of detection iterations r and on the success probability $1 - \varepsilon$ for the reconstruction of multivariate trigonometric polynomials p when using reconstructing single or multiple rank-1 lattices as sampling sets. Numerical tests performed in Section 5 demonstrate the high performance of the proposed method for high-dimensional trigonometric polynomials p as well as for a 10-dimensional test function f which is non-sparse in frequency domain. Finally, in Section 6, we conclude the results of this paper.

2 Prerequisites

For our method, we combine two main ingredients. One is using so-called reconstructing multiple rank-1 lattices for known frequency index sets I from [15, 16]. The second one is the dimension-incremental reconstruction approach for unknown frequency index sets I as presented in [26].

2.1 Reconstructing multiple rank-1 lattices

First, we start with definitions from [16] using slightly adapted symbols in this work.

The sampling sets $\mathcal{X}$ constructed and used by the method presented in Section 3 are based on so-called rank-1 lattices

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

where $\boldsymbol{z} \in \mathbb{Z}^d$ and $M \in \mathbb{N}$ are called generating vector and lattice size of $\Lambda(\boldsymbol{z}, M)$, respectively. For an arbitrary multivariate trigonometric polynomial p, see (1.1), with frequencies $\boldsymbol{k}$ supported on an index set $I \subset \mathbb{Z}^d$, $|I| < \infty$, we can reconstruct all the Fourier coefficients $\hat{p}_{\boldsymbol{k}}$, $\boldsymbol{k} \in I$ given in (1.2), from samples along a rank-1 lattice $\mathcal{X} := \Lambda(\boldsymbol{z}, M)$ if the Fourier matrix

$$\boldsymbol{A}(\mathcal{X}, I) := \left(\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{x}}\right)_{\boldsymbol{x}\in\mathcal{X}, \boldsymbol{k}\in I}$$

has full column rank. This is the case if and only if $\Lambda(\boldsymbol{z}, M)$ is a *reconstructing rank-1 lattice* for I, i.e., the *reconstruction property*

$$\boldsymbol{k} \cdot \boldsymbol{z} \not\equiv \boldsymbol{k}' \cdot \boldsymbol{z} \pmod{M} \quad \text{for all } \boldsymbol{k}, \boldsymbol{k}' \in I, \boldsymbol{k} \neq \boldsymbol{k}',$$

is fulfilled. Such a reconstructing rank-1 lattice $\Lambda(\boldsymbol{z}, M)$ can be easily constructed using a simple component-by-component construction method, cf. [14]. However, the construction

method has rather high computational costs and may require $\mathcal{O}(d|I|^3)$ arithmetic operations. Moreover, under mild assumptions, the lattice size M is bounded by $|I| \leq M \leq |I|^2$, where this number M tends more to the upper bound for many interesting structures of frequency index sets I.

Recently, in [15, 16], a modified approach was presented, which allows for drastically reducing the number of samples. This approach uses rank-1 lattices $\Lambda(\boldsymbol{z}, M)$ as building blocks and builds sampling sets based on multiple instances. The corresponding sampling sets are called multiple rank-1 lattices and they can be constructed by simple and efficient randomized construction algorithms. A multiple rank-1 lattice is the union of $L \in \mathbb{N}$ many rank-1 lattices,

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

and consists of

$$M := |\Lambda(\boldsymbol{z}_1, M_1, \ldots, \boldsymbol{z}_L, M_L)| \leq 1 - L + \sum_{\ell=1}^{L} M_\ell$$

many distinct nodes. If $\Lambda = \Lambda(\boldsymbol{z}_1, M_1, \ldots, \boldsymbol{z}_L, M_L)$ allows for the reconstruction of all multivariate trigonometric polynomials p with frequencies supported on a frequency index set I, it will be called *reconstructing multiple rank-1 lattice* for I.

In simplified terms, the basic idea is that each of the rank-1 lattices $\Lambda(\boldsymbol{z}_\ell, M_\ell)$, $\ell = 1, \ldots, L$, should be a reconstructing one for some index set $I_\ell \subset I$ and that $\bigcup_{\ell=1}^{L} I_\ell = I$. We remark that this condition is not sufficient in general and we require an additional property.

One construction method is Algorithm 1, which searches for a reconstructing multiple rank-1 lattice Λ for a given index set I such that the properties

$$\boldsymbol{k} \cdot \boldsymbol{z}_\ell \not\equiv \boldsymbol{k}' \cdot \boldsymbol{z}_\ell \pmod{M_\ell} \text{ for all } \boldsymbol{k} \in I_\ell, \, \boldsymbol{k}' \in I, \boldsymbol{k} \neq \boldsymbol{k}', \quad \bigcup_{\ell=1}^{L} I_\ell = I, \qquad (2.1)$$

are fulfilled. The rank-1 lattice sizes M_ℓ are chosen distinctly from the set

$$P_{\lambda, L_{\max}}^I := \left\{ p_j \in P^I \colon p_j = \begin{cases} \min\{p \in P^I \colon p > \lambda\} & : j = 1 \\ \min\{p \in P^I \colon p > p_{j-1}\} & : j = 2, \ldots, L_{\max}. \end{cases} \right\} \qquad (2.2)$$

of the $L_{\max} \in \mathbb{N}$ smallest prime numbers in

$$P^I := \{M' \in \mathbb{N} \colon M' \text{ prime with } |\{\boldsymbol{k} \bmod M' \colon \boldsymbol{k} \in I\}| = |I|\}$$

larger than a certain $\lambda \in \mathbb{N}$. With probability at least $1 - \gamma$, Λ returned by Algorithm 1 is a reconstructing multiple rank-1 lattice fulfilling properties (2.1). For fixed oversampling factor $c > 0$, the cardinality $M := |\Lambda|$ of the multiple rank-1 lattice is in

$$\mathcal{O}\big(\max\{|I|, N_I, -\log\gamma \, |\log|\log\gamma||\}(\log|I| - \log\gamma) \big), \qquad (2.3)$$

where $N_I := \max_{j=1,\ldots,d}\{\max_{\boldsymbol{k} \in I} k_j - \min_{\boldsymbol{l} \in I} l_j\}$ is the expansion of I. This follows since $M \lesssim L \max\{|I|, N_I, L \log L\} \lesssim \max\{|I|, N_I, L \log L\}(\log|I| - \log\gamma)$, cf. [16, Corollary 3.7], and

$$L \log L \lesssim \log|I| \log\log|I| - \log\gamma \log|\log\gamma| - \log\gamma \log\log|I| + \log|I| \log|\log\gamma|$$
$$\lesssim \log|I| \log\log|I| - \log\gamma \log|\log\gamma| \lesssim |I| - \log\gamma \, |\log|\log\gamma||.$$

At this point, we stress on the fact that the oversampling factor $M/|I|$ does not depend on the dimension d. Moreover, the construction using Algorithm 1 requires

$$\mathcal{O}\big(\max\{|I|, N_I, -\log\gamma\,|\log|\log\gamma||\}(\log|I| + d + \log\log N_I)(\log|I| - \log\gamma)\big), \qquad (2.4)$$

arithmetic operations. This is caused by the complexity of determining the lattice sizes M_ℓ, $\ell = 1, \ldots, L_{\max}$, e.g. using the sieve of Eratosthenes, which is

$$\begin{aligned}
&\lesssim M_{L_{\max}} \log\log M_{L_{\max}} \lesssim \max\{|I|, N_I, L\log L\}\,\log\log\max\{|I|, N_I, L\log L\} \\
&\lesssim \max\{|I|, N_I, -\log\gamma\,|\log|\log\gamma||\}(\log\log|I| + \log\log N_I + \log\log L) \\
&\lesssim \max\{|I|, N_I, -\log\gamma\,|\log|\log\gamma||\}\,(\log\log N_I)\,(\log\log|I| + \log\log L) \\
&\lesssim \max\{|I|, N_I, -\log\gamma\,|\log|\log\gamma||\}\,(\log\log N_I)\,(\log|I| - \log\gamma)
\end{aligned}$$

and of the for loop, which is

$$L_{\max}(|I|\log|I| + d|I|) \lesssim |I|(\log|I| + d)(\log|I| - \log\gamma).$$

We remark that the proposed method represents a minor modification of [16, Algorithm 4]. After randomly choosing the generating vector $\boldsymbol{z}_L$ in line 7 of Algorithm 1, the original version in [16, Algorithm 4] checks if the rank-1 lattice $\Lambda(\boldsymbol{z}_L, M_L)$ is able to reconstruct additional Fourier coefficients $\hat{p}_{\boldsymbol{k}}$, $\boldsymbol{k} \in I \setminus \tilde{I}$. Otherwise, a new generating vector $\boldsymbol{z}_L$ is randomly chosen and the check is repeated until at least one additional Fourier coefficient $\hat{p}_{\boldsymbol{k}}$, $\boldsymbol{k} \in I \setminus \tilde{I}$, can be reconstructed. The analysis in [16, Corollary 3.7] yields the same bounds with respect to the success probability of finding a reconstructing multiple rank-1 lattice Λ and with respect to the cardinality $|\Lambda|$ for [16, Algorithm 4] and Algorithm 1. However, the upper bound for the arithmetic complexity of [16, Algorithm 4] is only with high probability, whereas the arithmetic complexity is guaranteed for Algorithm 1 due to the realized modifications. In practice, we observe that Algorithm 1 may require slightly more samples than [16, Algorithm 4].

Another construction method is [16, Algorithm 6], which searches for a reconstructing multiple rank-1 lattice such that the properties

$$\boldsymbol{k} \cdot \boldsymbol{z}_\ell \not\equiv \boldsymbol{k}' \cdot \boldsymbol{z}_\ell \pmod{M_\ell} \text{ for all } \boldsymbol{k} \in I_\ell,\ \boldsymbol{k}' \in I \setminus \left\{\bigcup_{\ell'=1}^{\ell-1} I_{\ell'}\right\},\ \boldsymbol{k} \neq \boldsymbol{k}',\quad \bigcup_{\ell=1}^{L} I_\ell = I,$$

are fulfilled. Under mild assumptions, [16, Algorithm 6] returns a reconstructing multiple rank-1 lattice of cardinality $\mathcal{O}(|I|\log^2|I|)$ with high probability, where the constants do not depend on the dimension d, and requires $\mathcal{O}(|I|(d + \log|I|)\log^3|I|)$ arithmetic operations with high probability.

Besides the small cardinalities and fast construction algorithms, a further main advantage of reconstructing multiple rank-1 lattices is the existence of a direct and fast inversion method for computing Fourier coefficients $\hat{p}_{\boldsymbol{k}}$ from sampling values, cf. Algorithm 2 for multiple rank-1 lattices constructed by [16, Algorithm 4] or Algorithm 1 as well as cf. [15, Algorithm 6] for multiple rank-1 lattices constructed by [16, Algorithm 6].

2.2 Dimension-incremental method

We use the dimension-incremental approach from [26]. This approach proceeds similarly as a dimension-incremental method for anharmonic trigonometric polynomials based on Prony's

Algorithm 1 (Minor modification of [16, Algorithm 4]). Determining reconstructing multiple rank-1 lattices with pairwise distinct lattice sizes, fulfilling condition (2.1) with high probability.

Input:	$I \subset \mathbb{Z}^d$	frequency index set
	$c \in (1, \infty) \subset \mathbb{R}$	oversampling factor
	$\gamma \in (0, 1) \subset \mathbb{R}$	upper bound on failure probability

1: Determine maximal number of rank-1 lattices $L_{\max} := \lceil c^2/(c-1)^2 (\ln |I| - \ln \gamma)/2 \rceil$.
2: Determine lower bound $\lambda := c\,(|I| - 1)$ for lattice sizes.
3: Determine set $P^I_{\lambda, L_{\max}}$ of prime lattice sizes, cf. (2.2), and arrange $p_1 < \ldots < p_{L_{\max}}$.
4: Initialize index set $\tilde{I} := \emptyset$ of frequencies belonging to reconstructible Fourier coefficients.
5: **for** $L := 1, \ldots, L_{\max}$ **do**
6: $\quad$ Choose lattice size $M_L = p_L \in P^I_{\lambda, L_{\max}}$.
7: $\quad$ Choose generating vector $\boldsymbol{z}_L$ from $[0, M_L - 1]^d \cap \mathbb{Z}^d$ uniformly at random.
8: $\quad$ Determine $I_L := \{\boldsymbol{k} \in I: \ \nexists \boldsymbol{h} \in I \setminus \{\boldsymbol{k}\} \text{ with } \boldsymbol{k} \cdot \boldsymbol{z}_L \equiv \boldsymbol{h} \cdot \boldsymbol{z}_L \pmod{M_L}\}$, cf. (2.1).
9: $\quad$ Update $\tilde{I} := \tilde{I} \cup I_L$.
10: $\quad$ If $|\tilde{I}| = |I|$, then exit **for** loop.
11: **end for** L

Output:	$M_1, \ldots, M_L$	lattice sizes of rank-1 lattices and
	$\boldsymbol{z}_1, \ldots, \boldsymbol{z}_L$	generating vectors of rank-1 lattices such that
	$\Lambda(\boldsymbol{z}_1, M_1, \ldots, \boldsymbol{z}_L, M_L)$	is a reconstructing multiple rank-1 lattice for I with probability at least $1 - \gamma$

Complexity: $\mathcal{O}\big(|I|\,(\log|I| + d)\log|I|\big)$	for $|I| \gtrsim N_I$, fixed c and fixed γ, where $N_I := \max_{j=1,\ldots,d}\{\max_{\boldsymbol{k}\in I} k_j - \min_{\boldsymbol{l}\in I} l_j\}$ is the expansion of I.

Algorithm 2 Non-iterative reconstruction of a trigonometric polynomial p from sampling values along reconstructing multiple rank-1 lattices for known frequency index set I.

Input:	$I \subset \mathbb{Z}^d$	frequency index set, $|I| < \infty$
	$\Lambda := \Lambda(\boldsymbol{z}_1, M_1, \ldots, \boldsymbol{z}_L, M_L)$	reconstructing multiple rank-1 lattice for I fulfilling condition (2.1)
	$(p(\tilde{\boldsymbol{x}}_j))_{\tilde{\boldsymbol{x}}_j \in \Lambda}$	sampling values of trigonometric polynomial p

Initialize $\texttt{counter}[\boldsymbol{k}] := 0$ and $\tilde{\hat{p}}_{\boldsymbol{k}} := 0$ for $\boldsymbol{k} \in I$.
for $\ell := 1, \ldots, L$ **do**
$\quad$ Determine $I_\ell := \{\boldsymbol{k} \in I: \ \nexists \boldsymbol{h} \in I \setminus \{\boldsymbol{k}\} \text{ with } \boldsymbol{k} \cdot \boldsymbol{z}_\ell \equiv \boldsymbol{h} \cdot \boldsymbol{z}_\ell \pmod{M_\ell}\}$, cf. (2.1).
$\quad$ Compute $\tilde{\hat{g}}_{\boldsymbol{k}} := \frac{1}{M_\ell} \sum_{j=0}^{M_\ell - 1} p\left(\frac{j}{M_\ell}\boldsymbol{z}_\ell \bmod \mathbf{1}\right) \mathrm{e}^{-2\pi \mathrm{i} j \boldsymbol{k}\cdot\boldsymbol{z}_\ell/M_\ell}$ for $\boldsymbol{k} \in I_\ell$, using inverse rank-1 lattice FFT, cf. [13, Algorithm 3.2].
$\quad$ Set $\texttt{counter}[\boldsymbol{k}] := \texttt{counter}[\boldsymbol{k}] + 1$ and $\tilde{\hat{p}}_{\boldsymbol{k}} := \tilde{\hat{p}}_{\boldsymbol{k}} + \tilde{\hat{g}}_{\boldsymbol{k}}$ for $\boldsymbol{k} \in I_\ell$.
end for ℓ
Set $\tilde{\hat{p}}_{\boldsymbol{k}} := \tilde{\hat{p}}_{\boldsymbol{k}}/\texttt{counter}[\boldsymbol{k}]$ for $\boldsymbol{k} \in I$.

Output:	$\left(\tilde{\hat{p}}_{\boldsymbol{k}}\right)_{\boldsymbol{k}\in I}$	reconstructed Fourier coefficients

Complexity: $\mathcal{O}\big(\widetilde{M}\log\widetilde{M} + L\,|I|\,(d + \log|I|)\big)$	$|\Lambda| \leq \sum_{\ell=1}^L M_\ell =: \widetilde{M}$

method in [25]. For a given search domain $\Gamma \subset \mathbb{Z}^d$, $|\Gamma| < \infty$, which may be very large like e.g. a full grid $\hat{G}_N^d$, a frequency index set $I \subset \Gamma$ containing the approximately largest Fourier coefficients of a function f under consideration is determined based on samples. In doing so, the index set I is constructed component-by-component in a dimension-incremental way starting with the first component.

In order to describe the method, we introduce additional notation from [26] and we assume that the function under consideration is a multivariate trigonometric polynomial p. We denote the projection of a frequency $\boldsymbol{k} := (k_1, \ldots, k_d)^\top \in \mathbb{Z}^d$ to the components $\boldsymbol{i} := (i_1, \ldots, i_m) \in \{\boldsymbol{\iota} \in \{1, \ldots, d\}^m \colon \iota_t \neq \iota_{t'} \text{ for } t \neq t'\}$ by $\mathcal{P}_{\boldsymbol{i}}(\boldsymbol{k}) := (k_{i_1}, \ldots, k_{i_m})^\top \in \mathbb{Z}^m$. Correspondingly, we define the projection of a frequency index set $I \subset \mathbb{Z}^d$ to the components $\boldsymbol{i}$ by $\mathcal{P}_{\boldsymbol{i}}(I) := \{(k_{i_1}, \ldots, k_{i_m}) \colon \boldsymbol{k} \in I\}$. Using this notation, the general approach is the following:

1. Determine the first components of the unknown frequency index set, i.e., determine an index set $I^{(1)} \subseteq \mathcal{P}_1(\Gamma)$ which should be identical to the projection $\mathcal{P}_1(\operatorname{supp} \hat{p})$ or contain this projection, $I^{(1)} \supseteq \mathcal{P}_1(\operatorname{supp} \hat{p})$.

2. For dimension increment step $t = 2, \ldots, d$, i.e., for each additional dimension:

   a) Determine the t-th components of the unknown frequency index set, i.e., determine an index set $I^{(t)} \subseteq \mathcal{P}_t(\Gamma)$ which should be identical to the projection $\mathcal{P}_t(\operatorname{supp} \hat{p})$ or contain this projection, $I^{(t)} \supseteq \mathcal{P}_t(\operatorname{supp} \hat{p})$.

   b) Determine a suitable sampling set $\mathcal{X}^{(1,\ldots,t)} \subset \mathbb{T}^d$, $|\mathcal{X}^{(1,\ldots,t)}| \ll |\Gamma|$, which allows to detect those frequencies from the index set $(I^{(1,\ldots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\ldots,t)}(\Gamma)$ belonging to non-zero Fourier coefficients $\hat{p}_{\boldsymbol{k}}$.

   c) Sample the trigonometric polynomial p along the nodes of the sampling set $\mathcal{X}^{(1,\ldots,t)}$.

   d) Compute the Fourier coefficients $\tilde{\hat{p}}_{(1,\ldots,t),\boldsymbol{k}}$, $\boldsymbol{k} \in (I^{(1,\ldots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\ldots,t)}(\Gamma)$.

   e) Determine the non-zero Fourier coefficients from $\tilde{\hat{p}}_{(1,\ldots,t),\boldsymbol{k}}$, $\boldsymbol{k} \in (I^{(1,\ldots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\ldots,t)}(\Gamma)$, and obtain the index set $I^{(1,\ldots,t)}$ of detected frequencies. The $I^{(1,\ldots,t)}$ index set should be equal to the projection $\mathcal{P}_{(1,\ldots,t)}(\operatorname{supp} \hat{p})$.

3. Use the index set $I^{(1,\ldots,d)}$ and the computed Fourier coefficients $\tilde{\hat{p}}_{(1,\ldots,d),\boldsymbol{k}}$, $\boldsymbol{k} \in I^{(1,\ldots,d)}$, as an approximation for the support $\operatorname{supp} \hat{p}$ and the Fourier coefficients $\hat{p}_{\boldsymbol{k}}$, $\boldsymbol{k} \in \operatorname{supp} \hat{p}$.

The proposed approach includes the construction of a suitable sampling set in step 2b and the computation of (projected) Fourier coefficients in step 2d. There exist different methods for the realization of these steps.

In [26], the utilization of reconstructing rank-1 lattices was mainly considered as sampling sets, which allows for efficiently computing the (projected) Fourier coefficients using a so-called rank-1 lattice FFT, i.e., a single one-dimensional FFT followed by a simple index transform, cf. [13, Algorithm 3.2]. This approach yielded high stability and reliability in numerical tests, but required a relatively high number of samples and the construction of reconstructing rank-1 lattices may involve high computational costs.

Alternatively, for computing the (projected) Fourier coefficients, sub-sampling along reconstructing rank-1 lattices using ℓ_1 minimization or using a variant of Prony's method as well as randomly obtained generated sets using ℓ_1 minimization were also considered in [26]. These approaches performed well when determining the unknown frequency index sets I and Fourier

coefficients $\hat{p}_{\boldsymbol{k}}$ of sparse multivariate trigonometric polynomials p based on unperturbed sampling values. However, for the case of noisy sampling values or functions $f \in L_1(\mathbb{T}^d) \cap \mathcal{C}(\mathbb{T}^d)$ with infinite support in frequency domain, these methods failed in numerical tests.

In the next section, we present a novel approach, which is based on using multiple rank-1 lattices, which does not have these limitations.

3 Method

In this section, we present a substantial improvement of the dimension-incremental approach developed in [26]. On the one hand, we change the sampling sets to multiple rank-1 lattices $\Lambda = \Lambda(\boldsymbol{z}_1, M_1, \ldots, \boldsymbol{z}_L, M_L)$, which significantly reduces the number of required sampling values compared to single rank-1 lattices $\Lambda(\boldsymbol{z}, M)$. On the other hand, we describe modifications on [26, Algorithm 2], that allow for estimates of the success probability, the total number of required sampling values, and the overall complexity of the resulting algorithms. In particular for sparse trigonometric polynomials, these modifications will entail detailed estimates that only depend on properties of the frequency support of the sparse trigonometric polynomial and a factor that is logarithmic in the failure probability, cf. Section 4.

In Section 3.1, we present two algorithms based on sampling along multiple rank-1 lattices. These algorithms perform the steps as generally explained in Section 2.2. Afterwards, in Section 3.2, we discuss the number of required samples and arithmetic operations of the proposed methods in general.

3.1 Algorithm

We present the first proposed realization of the dimension-incremental approach from Section 2.2 as Algorithm 3, which uses multiple rank-1 lattices as sampling nodes, cf. Section 2.1.

Compared to [26, Algorithm 2], we perform the following modifications. In step 2b, we apply Algorithm 1 on the frequency index set $(I^{(1,\ldots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\ldots,t)}(\Gamma)$ in order to build a reconstructing multiple rank-1 lattice. This change of the construction of the sampling sets drastically reduces the arithmetic complexity of the overall algorithm as discussed in Section 3.2. Additionally, the resulting sampling set $\mathcal{X}^{(1,\ldots,t)}$ consists of distinctly less nodes and, consequently, the number of sampling values in step 2c is reduced. Afterwards in step 2d, the projected Fourier coefficients $\hat{\tilde{p}}_{(1,\ldots,t),\boldsymbol{k}}$, $\boldsymbol{k} \in (I^{(1,\ldots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\ldots,t)}(\Gamma)$, are computed using Algorithm 2. Moreover, we do not need to perform the additional step 2f from [26, Algorithm 2], which also reduces the number of arithmetic operations.

Another essential modification compared to the algorithms presented in [26] concerns the thresholding behavior. In [26] a relative threshold parameter $\theta \in \mathbb{R}^+$ with respect to a currently numerically computed Fourier coefficient $\hat{\tilde{p}}_{(1,\ldots,t),\boldsymbol{k}}$ was used. Here, we utilize an absolute threshold $\delta \in \mathbb{R}^+$, which is an input parameter of Algorithm 3. This absolute threshold is required for our theoretical considerations in Section 4, where we show estimates for failure probabilities and for the number of required detection iterations $r \in \mathbb{N}$.

Moreover, we introduce an additional input parameter $b \in \mathbb{N}$, which is the maximal number of multiple rank-1 lattice searches per component for the search of reconstructing multiple rank-1 lattices. This parameter b is required in order to achieve a deterministic runtime behavior and is considered theoretically in Lemma 4.5 of Section 4.1. In practice, choosing the parameters r and b as a small constant (e.g. ≤ 10) should often suffice as observed in the numerical tests in Section 5.

Algorithm 3 Reconstruction of a multivariate trigonometric polynomial p from sampling values along multiple rank-1 lattices.

Input: $\Gamma \subset \mathbb{Z}^d$ search space in frequency domain, superset for $\operatorname{supp} \hat{p}$

              $p(\circ)$ trigonometric polynomial p as black box (function handle)

              $\delta \in \mathbb{R}^+$ absolute threshold

              $s, s_{\mathrm{local}} \in \mathbb{N}$ sparsity parameter ($s_{\mathrm{local}} := s$ by default)

              $r \in \mathbb{N}$ number of detection iterations (e.g. see Theorem 4.6 or Section 5)

              $b \in \mathbb{N}$ maximal number of multiple rank-1 lattice searches per component

(step 1)

    Set $K_1 := \max(\mathcal{P}_1(\Gamma)) - \min(\mathcal{P}_1(\Gamma)) + 1$, $I^{(1)} := \emptyset$.

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

       Choose components $x_2', \ldots, x_d' \in \mathbb{T}$ of sampling nodes uniformly at random.

       Compute $\tilde{\hat{p}}_{1,k_1} := \frac{1}{K_1} \sum_{\ell=0}^{K_1-1} p\left(\frac{\ell}{K_1}, x_2', \ldots, x_d'\right) \mathrm{e}^{-2\pi \mathrm{i} \ell k_1 / K_1}$, $k_1 \in \mathcal{P}_1(\Gamma)$, with FFT.

       Set $I^{(1)} := I^{(1)} \cup \{k_1 \in \mathcal{P}_1(\Gamma) \colon$ (up to) s_{local}-largest values $|\tilde{\hat{p}}_{1,k_1}| \geq \delta\}$.

    end for i

(step 2)

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

(step 2a)

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

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

       Choose components $x_1', \ldots, x_{t-1}', x_{t+1}', \ldots, x_d' \in \mathbb{T}$ of sampling nodes uniformly at random.

       $\tilde{\hat{p}}_{t,k_t} := \sum_{\ell=0}^{K_t-1} p\left(x_1', \ldots, x_{t-1}', \frac{\ell}{K_t}, x_{t+1}', \ldots, x_d'\right) \mathrm{e}^{-2\pi \mathrm{i} \ell k_t / K_t}$, $k_t \in \mathcal{P}_t(\Gamma)$, using FFT.

       Set $I^{(t)} := I^{(t)} \cup \{k_t \in \mathcal{P}_t(\Gamma) \colon$ (up to) s_{local}-largest values $|\tilde{\hat{p}}_{t,k_t}| \geq \delta\}$.

    end for i

(step 2b)

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

    Search for reconstructing multiple rank-1 lattice $\Lambda(\boldsymbol{z}_1, M_1, \ldots, \boldsymbol{z}_L, M_L)$ for $(I^{(1,\ldots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\ldots,t)}(\Gamma)$ using **Algorithm 1** with algorithm parameters $c := 2$ and $\gamma := 0.5$; repeat up to $b-1$ times or until reconstruction property (2.1) is fulfilled.

    $\Lambda := \Lambda(\boldsymbol{z}_1, M_1, \ldots, \boldsymbol{z}_L, M_L)$ consists of nodes $\tilde{\boldsymbol{x}}_j \in \mathbb{T}^t$, $j = 0, \ldots, M^{(t)} - 1$, $M^{(t)} = |\Lambda|$.

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

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

       Set $\mathcal{X}^{(1,\ldots,t)} := \{\boldsymbol{x}_j := (\tilde{\boldsymbol{x}}_j, x_{t+1}', \ldots, x_d') \bmod \mathbf{1} \colon j = 0, \ldots, M^{(t)} - 1\} \subset \mathbb{T}^d$.

(step 2c) Sample p along the nodes of the sampling set $\mathcal{X}^{(1,\ldots,t)}$.

(step 2d) Compute $\tilde{\hat{p}}_{(1,\ldots,t),\boldsymbol{k}}$ for $\boldsymbol{k} \in (I^{(1,\ldots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\ldots,t)}(\Gamma)$ from sampling values $p(\boldsymbol{x}_j)$, $\boldsymbol{x}_j \in \mathcal{X}^{(1,\ldots,t)}$, with inverse multiple rank-1 lattice FFT using **Algorithm 2**.

(step 2e) Set $I^{(1,\ldots,t)} := I^{(1,\ldots,t)} \cup \{\boldsymbol{k} \in (I^{(1,\ldots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\ldots,t)}(\Gamma) \colon$

    (up to) $\tilde{s}$-largest values $|\tilde{\hat{p}}_{(1,\ldots,t),\boldsymbol{k}}| \geq \delta\}$.

    end for i

end for t

Algorithm 3 continued.

(step 3) Set $I := I^{(1,\ldots,d)}$ and $\tilde{\tilde{\boldsymbol{p}}} := \left(\tilde{\tilde{p}}_{(1,\ldots,d),\boldsymbol{k}}\right)_{\boldsymbol{k}\in I}$.

Output: $I \subset \Gamma \subset \mathbb{Z}^d$ index set of detected frequencies, $|I| \leq \min\{s, |\Gamma|\}$

        $\tilde{\tilde{\boldsymbol{p}}} \in \mathbb{C}^{|I|}$ corresponding Fourier coefficients, $|\tilde{\tilde{p}}_{(1,\ldots,d),\boldsymbol{k}}| \geq \delta$

Additionally, we present a second dimensional-incremental reconstruction algorithm. To this end, we modify Algorithm 3 by using [16, Algorithm 6] for the determination of the reconstructing multiple rank-1 lattices and obtain Algorithm 4. We stress on the fact that [16, Algorithm 6] does not ensure to terminate. However, if it is terminating, the output is always a reconstructing multiple rank-1 lattice. Therefore, we omit a repeatedly invocation of [16, Algorithm 6] in step 2b and we remove the parameter b. Since the construction approach of the multiple rank-1 lattices changes, we need to apply the corresponding FFT algorithms, i.e., the projected Fourier coefficients $\tilde{\tilde{p}}_{(1,\ldots,t),\boldsymbol{k}}$, $\boldsymbol{k} \in (I^{(1,\ldots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\ldots,t)}(\Gamma)$, in step 2d are now computed using the inverse multiple rank-1 lattice FFT from [15, Algorithm 6]. As discussed in Section 3.2, the sample and arithmetic complexities of Algorithm 4 are slightly higher by a logarithmic factor and non-deterministic. However, in the numerical tests in Section 5, the observed total number of samples and runtimes are usually smaller for Algorithm 4.

Moreover, we will apply parts of the theoretical results from Section 4 to the dimension-incremental approach using single rank-1 lattices as sampling sets. For this, we need to modify the thresholding behavior of [26, Algorithm 2]. As described above, we use an absolute threshold $\delta \in \mathbb{R}^+$ as input parameter and this yields Algorithm 5.

Remark 3.1. The dimension-incremental method as described in Section 2.2 and realized by [26, Algorithm 1,2] as well as Algorithm 3, 4 and 5 proceeds component-by-component starting with the first dimension, then the second et cetera. In principle, any order / permutation of the components $\{1, 2, \ldots, d\}$ may be used. Depending on the actual function under consideration, a different order may be beneficial or not, since this may influence the aliasing behavior in the various dimension increment steps t. $\square$

Remark 3.2. As discussed in [26, Section 2.2.1], we may set the number of detection iterations $r := 1$ for [26, Algorithm 1,2], Algorithm 3, Algorithm 4 and Algorithm 5 if the Fourier coefficients $\hat{p}_{\boldsymbol{k}}$ of the trigonometric polynomial p under consideration fulfill a certain property, namely, that the signs of the real part $\text{Re}(\hat{p}_{\boldsymbol{k}})$ of all non-zero Fourier coefficients $\hat{p}_{\boldsymbol{k}}$ are identical $\neq 0$ or that the signs of the imaginary part $\text{Im}(\hat{p}_{\boldsymbol{k}})$ are identical $\neq 0$. Then, we do not choose the components $x_1', \ldots, x_d' \in \mathbb{T}$ randomly but fix them to zero. In the case of [26, Algorithm 1,2] and Algorithm 5, this yields a purely deterministic dimension-incremental sparse FFT method. For Algorithm 3 and 4, we still perform probabilistic constructions since we apply Algorithm 1 and [16, Algorithm 6] for obtaining reconstructing multiple rank-1 lattices, respectively. We discuss the resulting sample and arithmetic complexities for Algorithm 3 and 5 in Section 4.3. $\square$

Remark 3.3. In the description of Algorithm 3, 4 and 5, we mention that the input parameter p is a trigonometric polynomial as a black box. Alternatively, we may also insert a function $f \in L_1(\mathbb{T}^d) \cap \mathcal{C}(\mathbb{T}^d)$ with infinitely many non-zero Fourier coefficients $\hat{f}_{\boldsymbol{k}}$. If these Fourier coefficients decay in a certain way, we may also successfully detect the approximately largest

Algorithm 4 (Modified Algorithm 3). Reconstruction of a multivariate trigonometric polynomial p from sampling values along reconstructing multiple rank-1 lattices.

Input:	$\Gamma \subset \mathbb{Z}^d$	search space in frequency domain, superset for $\operatorname{supp} \hat{p}$
	$p(\circ)$	trigonometric polynomial p as black box (function handle)
	$\delta \in \mathbb{R}^+$	absolute threshold
	$s, s_{\text{local}} \in \mathbb{N}$	sparsity parameter ($s_{\text{local}} := s$ by default)
	$r \in \mathbb{N}$	number of detection iterations

(step 1)

$\vdots$

(step 2)

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

 $\vdots$

(step 2b)

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

 Search for reconstructing multiple rank-1 lattice $\Lambda(\boldsymbol{z}_1, M_1, \ldots, \boldsymbol{z}_L, M_L)$ for $(I^{(1,\ldots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\ldots,t)}(\Gamma)$ using [**16, Algorithm 6**] with algorithm parameters $c := 2$ and $\delta' := 0.5$

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

  $\vdots$

(step 2d) Compute $\tilde{\hat{p}}_{(1,\ldots,t),\boldsymbol{k}}$ for $\boldsymbol{k} \in (I^{(1,\ldots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\ldots,t)}(\Gamma)$ from sampling values $p(\boldsymbol{x}_j)$, $\boldsymbol{x}_j \in \mathcal{X}^{(1,\ldots,t)}$, with inverse multiple rank-1 lattice FFT using [**15, Algorithm 6**].

  $\vdots$

 end for i

end for t

(step 3)

 Compute $\tilde{\hat{p}}_{(1,\ldots,d),\boldsymbol{k}}$ for $\boldsymbol{k} \in I^{(1,\ldots,d)}$ from sampling values $p(\boldsymbol{x}_j)$, $\boldsymbol{x}_j \in \mathcal{X}^{(1,\ldots,d)}$, with inverse multiple rank-1 lattice FFT using [15, Algorithm 6].

 Set $I := I^{(1,\ldots,d)}$ and $\tilde{\hat{\boldsymbol{p}}} := \left(\tilde{\hat{p}}_{(1,\ldots,d),\boldsymbol{k}} \right)_{\boldsymbol{k} \in I}$.

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

ones by choosing the input parameters appropriately as the numerical results in Section 5.3 indicate. $\qquad\square$

After presenting the new algorithms, we give its sample and arithmetic complexities.

3.2 Number of samples and arithmetic complexity

We provide a detailed description of the complexities for various steps of Algorithm 3, 4 and 5. We perform the analysis for the case where the search domain Γ is the full grid $\hat{G}_N^d$ of expansion $N \in \mathbb{N}$ and the sparsity parameter $s \gtrsim N$. For other cases, the resulting sample and arithmetic complexities can be determined analogously. When estimating the complexities, we keep track of the number of detection iterations $r \in \mathbb{N}$ and the theoretical size of this parameter is discussed later in Section 4. In practice, as the numerical results in Section 5

Algorithm 5 (Modified version of [26, Algorithm 2]). Reconstruction of a multivariate trigonometric polynomial p from sampling values along (single) reconstructing rank-1 lattices.

Input:	$\Gamma \subset \mathbb{Z}^d$	search space in frequency domain, superset for supp $\hat{p}$
	$p(\circ)$	trigonometric polynomial p as black box (function handle)
	$\delta \in \mathbb{R}^+$	absolute threshold
	$s, s_{\text{local}} \in \mathbb{N}$	sparsity parameter ($s_{\text{local}} := s$ by default)
	$r \in \mathbb{N}$	number of detection iterations (e.g. see Theorem 4.7 or Section 5)

(step 1)

$\vdots$

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

    $\vdots$

    $I^{(1)} := I^{(1)} \cup \{k_1 \in \mathcal{P}_1(\Gamma) \colon \text{(up to) } s_{\text{local}}\text{-largest values } |\tilde{\hat{p}}_{1,k_1}| \geq \delta\}$

  end for i

$\vdots$

(step 2)

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

  $\vdots$

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

    $\vdots$

    Set $I^{(t)} := I^{(t)} \cup \{k_t \in \mathcal{P}_t(\Gamma) \colon \text{(up to) } s_{\text{local}}\text{-largest values } |\tilde{\hat{p}}_{t,k_t}| \geq \delta\}$.

  end for i

(step 2b)

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

  $\vdots$

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

    $\vdots$

(step 2e)

    Set $I^{(1,\ldots,t)} := I^{(1,\ldots,t)} \cup \{\boldsymbol{k} \in (I^{(1,\ldots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\ldots,t)}(\Gamma) \colon$

  (up to) $\tilde{s}$-largest values $|\tilde{\hat{p}}_{(1,\ldots,t),\boldsymbol{k}}| \geq \delta\}$.

  end for i

  $\vdots$

end for t

(step 3) Set $I := I^{(1,\ldots,d)}$ and $\tilde{\boldsymbol{\hat{p}}} := \left(\tilde{\hat{p}}_{(1,\ldots,d),\boldsymbol{k}}\right)_{\boldsymbol{k} \in I}$.

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

suggest, the number of detection iterations r may be chosen as a constant value in many cases, e.g. $r = 1$ for multivariate trigonometric polynomials p in the noiseless case or ≤ 10 for other considered test settings.

In step 2 of Algorithm 3 in dimension increment step t, the index sets $I^{(1,\dots,t-1)}$ and $I^{(t)}$ consist of at most rs and $|\hat{G}_N^1| = 2N+1$ many frequencies, respectively. This yields that the index set $I^{(1,\dots,t-1)} \times I^{(t)}$ consists of $|I^{(1,\dots,t-1)} \times I^{(t)}| \leq rs\,(2N+1) \lesssim rsN$ frequency candidates. The sampling set $\mathcal{X}^{(1,\dots,t)}$ built in step 2b of Algorithm 3, where $\mathcal{P}_{(1,\dots,t)}(\mathcal{X}^{(1,\dots,t)})$ is a multiple rank-1 lattice returned by applying Algorithm 1 on the index set $J_t := (I^{(1,\dots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\dots,t)}(\Gamma)$, has the size

$$|\mathcal{X}^{(1,\dots,t)}| < 2\tilde{L}\max\{2(rs\,(2N+1)-1), 4\tilde{L}\ln\tilde{L}\} \lesssim rsN\log(rsN),$$

where $\tilde{L} := \lceil 2\ln(rs\,(2N+1)) + 2\ln 2\rceil$, cf. the proof of Lemma 4.5 with $c := 2$ and $\gamma := 1/2$. Moreover, the construction of the sampling set $\mathcal{X}^{(1,\dots,t)}$ requires $\mathcal{O}\big(b\,rsN(\log(rsN) + t)\log(rsN) + d\big)$ arithmetic operations. The inverse multiple rank-1 lattice FFT, cf. Algorithm 2, in step 2d requires $\lesssim |\mathcal{X}^{(1,\dots,t)}|\log|\mathcal{X}^{(1,\dots,t)}| + \tilde{L}|J_t|\,(t + \log|J_t|) \lesssim rsN\big(\log(rsN) + t\big)\log(rsN)$ arithmetic operations for each detection iteration $i \in \{1,\dots,r\}$ and each dimension increment step $t \in \{2,\dots,d\}$.

In total, this yields

$$\lesssim dr^2sN\log(rsN) + d\,r\,N \lesssim dr^2sN\log(rsN) \tag{3.1}$$

many samples and

$$\lesssim d\,r^2sN\,(d + \log(rsN))\,\log(rsN) \lesssim d^2\,r^2sN\,\log^2(rsN) \tag{3.2}$$

arithmetic operations for Algorithm 3 assuming $r \gtrsim b$. We remark that we do not have any exponential dependence in the dimension d, neither for the number of samples nor for the arithmetic complexity, but only a polynomial dependence of degree ≤ 2. The sample and arithmetic complexities are distinctly reduced by using Algorithm 3 (multiple rank-1 lattices) and compared to Algorithm 5 or [26, Algorithm 2] (single rank-1 lattices), where the latter two algorithms require $\lesssim dr^3s^2N$ samples and $\lesssim dr^3s^3 + dr^3s^2N\log(rsN)$ arithmetic operations.

In particular, the number of used sampling values as well as the arithmetic complexity is distinctly improved to an almost linear term with respect to the sparsity parameter s for Algorithm 3 and multiple rank-1 lattices.

When analyzing the sample and arithmetic complexity of Algorithm 4, we obtain similar numbers except for additional logarithmic factors. In total, we require

$$\lesssim dr^2sN\log^2(rsN) + d\,r\,N \lesssim dr^2sN\log^2(rsN)$$

many samples with high probability and

$$\lesssim drsN\big(d\log(rsN) + r\log(rsN) + d\big)\log^2(rsN) \lesssim d^2r^2sN\,\log^3(rsN)$$

arithmetic operations with high probability for Algorithm 4. We remark that we do not observe these additional logarithmic factors in numerical tests. Presumably, this is due to the relatively rough estimates of the complexities in [16]. Moreover, the numerical results in Section 5 suggest that Algorithm 4 requires even less samples and has smaller runtime than Algorithm 3 in practice.

We remark that for obtaining the estimates of the sample and arithmetic complexities in this section, we did not assume that the function under consideration is a multivariate trigonometric polynomial p, see also Remark 3.3. In the next section, we will especially address the special case, when the function under consideration is sparse in frequency domain, and we will give possible choices for the input parameters of Algorithm 3 and 5 depending on the targeted success probability.

4 Theoretical results

In this section, we analyze in detail the sample and arithmetic complexities of Algorithm 3 and 5 applied to multivariate trigonometric polynomials p. We start with the case of sparse multivariate trigonometric polynomials p having arbitrary Fourier coefficients $\hat{p}_{\boldsymbol{k}}$ in Section 4.1. Afterwards, in Section 4.2, we discuss the obtained results. Later, in Section 4.3, we consider the case where the Fourier coefficients $\hat{p}_{\boldsymbol{k}}$ fulfill certain stronger properties, resulting in a modification of Algorithm 3 and 5 with small sample and arithmetic complexities.

4.1 Multivariate trigonometric polynomial with arbitrary Fourier coefficients

We analyze the computation of (projected) Fourier coefficients in step t. We denote the sampling set $\Lambda = \Lambda(\boldsymbol{z}_1, M_1, \ldots, \boldsymbol{z}_L, M_L)$ for short and assume that Λ is a reconstructing multiple rank-1 lattice for the frequency index set

$$J_t := \left(I^{(1,\ldots,t-1)} \times I^{(t)} \right) \cap \mathcal{P}_{(1,\ldots,t)}(\Gamma),$$

$J_t \supset \mathcal{P}_{(1,\ldots,t)}(\operatorname{supp} \hat{p})$, determined by Algorithm 1 in Section 2.1 or by one of Algorithms 1 to 4 from [16]. We denote $\tilde{\boldsymbol{x}} := (x'_{t+1}, \ldots, x'_d)$. We use Algorithm 2 in order to determine the projected Fourier coefficients $\hat{\tilde{p}}^{\Lambda}_{(1,\ldots,t),\boldsymbol{k}}$, $\boldsymbol{k} \in J_t$. Due to the construction of the reconstructing multiple rank-1 lattice Λ, for each $\boldsymbol{k} \in J_t$, there exists at least one $\ell \in \{1, \ldots, L\}$ such that

$$\boldsymbol{k} \cdot \boldsymbol{z}_\ell \not\equiv \boldsymbol{k}' \cdot \boldsymbol{z}_\ell \pmod{M_\ell} \quad \text{for all } \boldsymbol{k} \neq \boldsymbol{k}' \in J_t \tag{4.1}$$

holds and we have

$$
\begin{aligned}
\hat{\tilde{p}}^{\Lambda(\boldsymbol{z}_\ell, M_\ell)}_{(1,\ldots,t),\boldsymbol{k}} &:= \frac{1}{M_\ell} \sum_{j=1}^{M_\ell-1} p\left(\frac{j}{M_\ell} \boldsymbol{z}_\ell, \tilde{\boldsymbol{x}} \right) \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{z}_\ell \frac{j}{M_\ell}} \\
&= \sum_{\boldsymbol{h}\in\operatorname{supp}\hat{p}} \frac{1}{M_\ell} \sum_{j=1}^{M_\ell-1} \hat{p}_{\boldsymbol{h}} \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{h}\cdot(\boldsymbol{z}_\ell \frac{j}{M_\ell}, \tilde{\boldsymbol{x}})} \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{z}_\ell \frac{j}{M_\ell}} \\
&= \sum_{\substack{\boldsymbol{h}\in\operatorname{supp}\hat{p} \\ ((h_1,\ldots,h_t)-\boldsymbol{k})\cdot\boldsymbol{z}_\ell\equiv 0 \pmod{M_\ell}}} \hat{p}_{\boldsymbol{h}} \, \mathrm{e}^{2\pi\mathrm{i}(h_{t+1},\ldots,h_d)\cdot\tilde{\boldsymbol{x}}} \\
&= \sum_{\substack{\boldsymbol{h}\in\operatorname{supp}\hat{p} \\ (h_1,\ldots,h_t)=\boldsymbol{k}}} \hat{p}_{(\boldsymbol{k},h_{t+1},\ldots,h_d)} \, \mathrm{e}^{2\pi\mathrm{i}(h_{t+1},\ldots,h_d)\cdot\tilde{\boldsymbol{x}}}.
\end{aligned}
$$

If there exists more than one $\ell \in \{1, \ldots, L\}$ such that (4.1) holds, then Algorithm 2 computes the average of all these coefficients $\hat{\tilde{p}}^{\Lambda(\boldsymbol{z}_\ell, M_\ell)}_{(1,\ldots,t),\boldsymbol{k}}$, where the latter are identical. Consequently, we

obtain projected Fourier coefficients

$$\tilde{\hat{p}}^{\Lambda}_{(1,\ldots,t),\boldsymbol{k}} = \sum_{\substack{\boldsymbol{h}\in\operatorname{supp}\hat{p} \\ (h_1,\ldots,h_t)=\boldsymbol{k}}} \hat{p}_{(\boldsymbol{k},h_{t+1},\ldots,h_d)}\,\mathrm{e}^{2\pi\mathrm{i}(h_{t+1},\ldots,h_d)\cdot\tilde{\boldsymbol{x}}}.$$

As in [26, Section 2.2.2], we interpret the projected Fourier coefficients $\tilde{\hat{p}}^{\Lambda}_{(1,\ldots,t),\boldsymbol{k}}$ as a trigonometric polynomial $g(\tilde{\boldsymbol{x}}) := \tilde{\hat{p}}^{\Lambda}_{(1,\ldots,t),\boldsymbol{k}}$ with respect to $\tilde{\boldsymbol{x}} \in \mathbb{T}^{d-t}$. In Algorithm 3, we choose $\tilde{\boldsymbol{x}} \in \mathbb{T}^{d-t}$ uniformly at random. As an improvement to [26, Theorem 2.5], we estimate the probability that a projected Fourier coefficient $\tilde{\hat{p}}^{\Lambda}_{(1,\ldots,t),\boldsymbol{k}}$ is of a certain absolute value, i.e., we estimate the measure of the following set

$$\left\{ \tilde{\boldsymbol{x}} \in \mathbb{T}^{d-t}\colon |g(\tilde{\boldsymbol{x}})| = \left|\sum_{\boldsymbol{h}\in\tilde{I}} \hat{g}_{\boldsymbol{h}}\,\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{h}\cdot\tilde{\boldsymbol{x}}}\right| < \delta \right\}.$$

Similarly, for $\Gamma \supset \operatorname{supp}\hat{p}$ and $t \in \{1,\ldots,d\}$, we have for the projected Fourier coefficient

$$\begin{aligned}
\tilde{\hat{p}}_{t,k_t} &:= \sum_{\ell=0}^{K_t-1} p\left(x'_1,\ldots,x'_{t-1},\frac{\ell}{K_t},x'_{t+1},\ldots,x'_d\right)\mathrm{e}^{-2\pi\mathrm{i}\ell k_t/K_t} \\
&= \sum_{\substack{\boldsymbol{h}\in\operatorname{supp}\hat{p} \\ h_t=k_t}} \hat{p}_{\boldsymbol{h}}\,\mathrm{e}^{2\pi\mathrm{i}(h_1,\ldots,h_{t-1},h_{t+1},\ldots,h_d)\cdot\tilde{\boldsymbol{x}}} \quad \text{for } k_t \in \mathcal{P}_t(\Gamma)
\end{aligned} \tag{4.2}$$

in step 1 and 2a of Algorithm 3, and we interpret $\tilde{\hat{p}}_{t,k_t}$ as a trigonometric polynomial $g(\tilde{\boldsymbol{x}}) := \tilde{\hat{p}}_{t,k_t}$ with respect to $\tilde{\boldsymbol{x}} := (x'_1,\ldots,x'_{t-1},x'_{t+1},\ldots,x'_d) \in \mathbb{T}^{d-1}$.

Lemma 4.1. *Let a trigonometric polynomial* $g : \mathbb{T}^n \to \mathbb{C}$, $n \in \mathbb{N}$, $g(\boldsymbol{x}) := \sum_{\boldsymbol{h}\in\tilde{I}} \hat{g}_{\boldsymbol{h}}\,\mathrm{e}^{2\pi\mathrm{i}\boldsymbol{h}\cdot\boldsymbol{x}} \not\equiv 0$, $\tilde{I} \subset \mathbb{Z}^n$, $|\tilde{I}| < \infty$, *be given such that the property* $\|g|L^1(\mathbb{T}^n)\| > \delta$ *for a threshold* $\delta > 0$ *is fulfilled. Moreover, let* $X_1,\ldots,X_n \in \mathbb{T}$ *be independent, identical, uniformly distributed random variables and we denote by* $\boldsymbol{X} := (X_1,\ldots,X_n)^{\top} \in \mathbb{T}^n$ *the random vector. Then, the probability*

$$\mathbb{P}(|g(\boldsymbol{X})| < \delta) \leq 1 - \frac{\|g|L_1(\mathbb{T}^n)\| - \delta}{\|g|L_\infty(\mathbb{T}^n)\|} = \frac{\|g|L_\infty(\mathbb{T}^n)\| - \|g|L_1(\mathbb{T}^n)\| + \delta}{\|g|L_\infty(\mathbb{T}^n)\|} < 1.$$

If $\max_{\boldsymbol{h}\in\tilde{I}} |\hat{g}_{\boldsymbol{h}}| > \delta$, *then*

$$\mathbb{P}(|g(\boldsymbol{X})| < \delta) \leq 1 - \frac{\max_{\boldsymbol{h}\in\tilde{I}} |\hat{g}_{\boldsymbol{h}}| - \delta}{\sum_{\boldsymbol{h}\in\tilde{I}} |\hat{g}_{\boldsymbol{h}}|} =: q < 1.$$

Choosing r *random vectors* $\boldsymbol{X}_1,\ldots,\boldsymbol{X}_r \in \mathbb{T}^n$ *independently, we observe*

$$\mathbb{P}\left(\bigcap_{i=1}^r \{|g(\boldsymbol{X}_i)| < \delta\}\right) \leq q^r.$$

Proof. Let Y be a real valued random variable. We refer to the lower bound

$$\frac{\mathbb{E}h(Y) - h(\delta)}{\|h(Y)|L_\infty(\mathbb{R})\|} \leq \mathbb{P}(|Y| \geq \delta)$$

in [23, Par. 9.3.A] and apply this for the even and on $[0, \infty)$ nondecreasing function $h(t) := |t|$. We set $Y = |g(\boldsymbol{X})|$ and we obtain

$$\mathbb{P}(|g(\boldsymbol{X})| \geq \delta) \geq \frac{\|g|L_1(\mathbb{T}^n)\| - \delta}{\|g|L_\infty(\mathbb{T}^n)\|}$$

and

$$\mathbb{P}(|g(\boldsymbol{X})| < \delta) \leq 1 - \frac{\|g|L_1(\mathbb{T}^n)\| - \delta}{\|g|L_\infty(\mathbb{T}^n)\|} = \frac{\|g|L_\infty(\mathbb{T}^n)\| - \|g|L_1(\mathbb{T}^n)\| + \delta}{\|g|L_\infty(\mathbb{T}^n)\|}.$$

The estimate $|\hat{g}_{\boldsymbol{h}}| = |\int_{\mathbb{T}^n} g(\boldsymbol{x}) \, e^{-2\pi i \boldsymbol{h} \cdot \boldsymbol{x}} \mathrm{d}\boldsymbol{x}| \leq \int_{\mathbb{T}^n} |g(\boldsymbol{x})| \mathrm{d}\boldsymbol{x} = \|g|L^1(\mathbb{T}^n)\|$ for all $\boldsymbol{h} \in \tilde{I}$ yields $\max_{\boldsymbol{h} \in \tilde{I}} |\hat{g}_{\boldsymbol{h}}| \leq \|g|L^1(\mathbb{T}^n)\|$. Since we have $\|g|L^\infty(\mathbb{T}^n)\| = \operatorname{ess\,sup}_{\boldsymbol{x} \in \mathbb{T}^n} |g(\boldsymbol{x})| \leq \sum_{\boldsymbol{h} \in \tilde{I}} |\hat{g}_{\boldsymbol{h}}|$ and $\max_{\boldsymbol{h} \in \tilde{I}} |\hat{g}_{\boldsymbol{h}}| > \delta$, the assertion follows. $\blacksquare$

Based on these estimates, we determine bounds on the probability that a projected Fourier coefficient is below an absolute threshold δ. Since the projected Fourier coefficient may consist only of minimal non-zero Fourier coefficients $\hat{p}_{\boldsymbol{h}}$, the threshold δ has to be chosen $\delta < \min_{\boldsymbol{h} \in \operatorname{supp} \hat{p}} |\hat{p}_{\boldsymbol{h}}|$ in order to apply the second part of Lemma 4.1.

Corollary 4.2. *Let a threshold value $\delta > 0$, a trigonometric polynomial p with the property $\min_{\boldsymbol{h} \in \operatorname{supp} \hat{p}} |\hat{p}_{\boldsymbol{h}}| > \delta$ and a search space $\Gamma \supset \operatorname{supp} \hat{p}$ of finite cardinality be given. For fixed $t \in \{1, \ldots, d\}$ and $K_t := \max(\mathcal{P}_t(\Gamma)) - \min(\mathcal{P}_t(\Gamma)) + 1$, we compute the one-dimensional projected Fourier coefficients for the t-th component by*

$$\tilde{\hat{p}}_{t,k_t} = \tilde{\hat{p}}_{t,k_t}(x'_1, \ldots, x'_{t-1}, x'_{t+1}, \ldots, x'_d)$$

$$:= \sum_{\ell=0}^{K_t-1} p\left(x'_1, \ldots, x'_{t-1}, \frac{\ell}{K_t}, x'_{t+1}, \ldots, x'_d\right) e^{-2\pi i \ell k_t / K_t}, \quad k_t \in \mathcal{P}_t(\Gamma),$$

where the values $x'_1, \ldots, x'_{t-1}, x'_{t+1}, \ldots, x'_d \in \mathbb{T}$ are independently chosen uniformly at random. Then, the probability

$$\mathbb{P}(|\tilde{\hat{p}}_{t,k_t}| < \delta) \leq 1 - \frac{\|\tilde{\hat{p}}_{t,k_t}|L_1(\mathbb{T}^{d-1})\| - \delta}{\|\tilde{\hat{p}}_{t,k_t}|L_\infty(\mathbb{T}^{d-1})\|}$$

$$\leq 1 - \frac{\max_{\boldsymbol{h}=(h_1,\ldots,h_{t-1},k_t,h_{t+1},\ldots,h_d) \in \operatorname{supp} \hat{p}} |\hat{p}_{\boldsymbol{h}}| - \delta}{\sum_{\boldsymbol{h}=(h_1,\ldots,h_{t-1},k_t,h_{t+1},\ldots,h_d) \in \operatorname{supp} \hat{p}} |\hat{p}_{\boldsymbol{h}}|} =: q_{t,k_t} < 1 \quad \text{for } k_t \in \mathcal{P}_t(\operatorname{supp} \hat{p})$$

holds due to Lemma 4.1. Repeating the computation of $\tilde{\hat{p}}_{t,k_t} = \tilde{\hat{p}}_{t,k_t}(\tilde{\boldsymbol{x}})$ for different randomly chosen $\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_r \in \mathbb{T}^{d-1}$, $r \in \mathbb{N}$, we estimate

$$\mathbb{P}\left(\max_{\nu=1,\ldots,r} |\tilde{\hat{p}}_{t,k_t}(\tilde{\boldsymbol{x}}_\nu)| < \delta\right) \leq (q_{t,k_t})^r.$$

Applying the union bound yields

$$\mathbb{P}\left(\bigcup_{k_t \in \mathcal{P}_t(\operatorname{supp} \hat{p})} \left\{\max_{\nu=1,\ldots,r} |\tilde{\hat{p}}_{t,k_t}(\tilde{\boldsymbol{x}}_\nu)| < \delta\right\}\right) \leq \sum_{k_t \in \mathcal{P}_t(\operatorname{supp} \hat{p})} (q_{t,k_t})^r$$

$$\leq \min\left\{|\operatorname{supp} \hat{p}|, K_t\right\} \left(\max_{k_t \in \mathcal{P}_t(\operatorname{supp} \hat{p})} q_{t,k_t}\right)^r. \quad (4.3)$$

$\blacksquare$

Corollary 4.3. *Let a threshold value $\delta > 0$, a trigonometric polynomial p with the property $\min_{\boldsymbol{h} \in \operatorname{supp} \hat{p}} |\hat{p}_{\boldsymbol{h}}| > \delta$ and the search space $\Gamma \supset \operatorname{supp} \hat{p}$ of finite cardinality be given. For fixed $t \in \{2, \ldots, d-1\}$, we compute the t-dimensional projected Fourier coefficients $\hat{\tilde{p}}^\Lambda_{(1,\ldots,t),\boldsymbol{k}}$, $\boldsymbol{k} \in \big(I^{(1,\ldots,t-1)} \times I^{(t)} \big) \cap \mathcal{P}_{(1,\ldots,t)}(\Gamma)$, in step 2d of Algorithm 3 by applying Algorithm 2 on $p(\circ, \tilde{\boldsymbol{x}})$, where $\tilde{\boldsymbol{x}}$ is choosen uniformly at random in $\mathbb{T}^{d-t}$. If $\Lambda(\boldsymbol{z}_1, M_1, \ldots, \boldsymbol{z}_L, M_L)$ is a reconstructing multiple rank-1 lattice for $\mathcal{P}_{(1,\ldots,t)}(\operatorname{supp} \hat{p})$ determined by Algorithm 1 in Section 2.1 or by one of [16, Algorithms 1 to 4], then the probability*

$$
\begin{aligned}
\mathbb{P}(|\hat{\tilde{p}}^\Lambda_{(1,\ldots,t),\boldsymbol{k}}| < \delta) &\leq 1 - \frac{\|\hat{\tilde{p}}^\Lambda_{(1,\ldots,t),\boldsymbol{k}}|L_1(\mathbb{T}^{d-t})\| - \delta}{\|\hat{\tilde{p}}^\Lambda_{(1,\ldots,t),\boldsymbol{k}}|L_\infty(\mathbb{T}^{d-t})\|} \\
&\leq 1 - \frac{\max_{\boldsymbol{h}=(\boldsymbol{k},h_{t+1},\ldots,h_d) \in \operatorname{supp} \hat{p}} |\hat{p}_{\boldsymbol{h}}| - \delta}{\sum_{\boldsymbol{h}=(\boldsymbol{k},h_{t+1},\ldots,h_d) \in \operatorname{supp} \hat{p}} |\hat{p}_{\boldsymbol{h}}|} \\
&=: q_{(1,\ldots,t),\boldsymbol{k}} < 1 \quad \text{for } \boldsymbol{k} \in \mathcal{P}_{(1,\ldots,t)}(\operatorname{supp} \hat{p}) \cap \big(I^{(1,\ldots,t-1)} \times I^{(t)} \big) \quad (4.4)
\end{aligned}
$$

due to Lemma 4.1. Repeating the computation of $\hat{\tilde{p}}^\Lambda_{(1,\ldots,t),\boldsymbol{k}} = \hat{\tilde{p}}^\Lambda_{(1,\ldots,t),\boldsymbol{k}}(\tilde{\boldsymbol{x}})$ for different randomly chosen $\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_r \in \mathbb{T}^{d-t}$, $r \in \mathbb{N}$, we estimate

$$
\mathbb{P}\left(\max_{\nu=1,\ldots,r} |\hat{\tilde{p}}^\Lambda_{(1,\ldots,t),\boldsymbol{k}}(\tilde{\boldsymbol{x}}_\nu)| < \delta \right) \leq (q_{(1,\ldots,t),\boldsymbol{k}})^r.
$$

Applying the union bound yields

$$
\begin{aligned}
&\mathbb{P}\left(\bigcup_{\boldsymbol{k} \in \mathcal{P}_{(1,\ldots,t)}(\operatorname{supp} \hat{p}) \cap \big(I^{(1,\ldots,t-1)} \times I^{(t)} \big)} \left\{ \max_{\nu=1,\ldots,r} |\hat{\tilde{p}}^\Lambda_{(1,\ldots,t),\boldsymbol{k}}(\tilde{\boldsymbol{x}}_\nu)| < \delta \right\} \right) \\
&\leq \sum_{\boldsymbol{k} \in \mathcal{P}_{(1,\ldots,t)}(\operatorname{supp} \hat{p}) \cap \big(I^{(1,\ldots,t-1)} \times I^{(t)} \big)} (q_{(1,\ldots,t),\boldsymbol{k}})^r \\
&\leq |\operatorname{supp} \hat{p}| \left(\max_{\boldsymbol{k} \in \mathcal{P}_{(1,\ldots,t)}(\operatorname{supp} \hat{p}) \cap \big(I^{(1,\ldots,t-1)} \times I^{(t)} \big)} q_{(1,\ldots,t),\boldsymbol{k}} \right)^r. \quad (4.5)
\end{aligned}
$$

$\blacksquare$

Analyzing Algorithm 3 in detail, we compute the adaptive approximation in d dimension increment steps. In each of the dimension increment steps, at most three probabilistic sub-steps are performed, cf. step 1, 2a, 2b. If each of these probabilistic sub-steps is successful, we detect all frequencies from $\operatorname{supp} \hat{p}$ correctly. We use the union bound to estimate the corresponding probability

$$
\begin{aligned}
\mathbb{P}\left(\bigcap_{t=1}^d A_t \cap \bigcap_{t=2}^d B_t \cap \bigcap_{t=2}^d C_t \right) &= 1 - \mathbb{P}\left(\bigcup_{t=1}^d A_t^\complement \cup \bigcup_{t=2}^d B_t^\complement \cup \bigcup_{t=2}^d C_t^\complement \right) \\
&\geq 1 - \sum_{t=1}^d \mathbb{P}(A_t^\complement) - \sum_{t=2}^d \mathbb{P}(B_t^\complement) - \sum_{t=2}^d \mathbb{P}(C_t^\complement), \quad (4.6)
\end{aligned}
$$

19

where the events

$$A_t := \left\{ \mathcal{P}_t(\operatorname{supp}\hat{p}) \subset I^{(t)} \right\},$$

$$B_t := \left\{ \Lambda(\boldsymbol{z}_1, M_1, \ldots, \boldsymbol{z}_L, M_L) \text{ is a reconstructing multiple rank-1 lattice} \right.$$

$$\left. \text{for } \mathcal{P}_{(1,\ldots,t)}(\operatorname{supp}\hat{p}) \right\},$$

$$C_t := \left\{ \mathcal{P}_{(1,\ldots,t)}(\operatorname{supp}\hat{p}) \subset I^{(1,\ldots,t)} \right\}.$$

The probabilities $\mathbb{P}(A_t^{\complement})$ and $\mathbb{P}(C_t^{\complement})$ of the complements of A_t and C_t were estimated in (4.3) and (4.5), respectively. An upper bound on $\mathbb{P}(B_t^{\complement})$ is obtained via the maximal number b of multiple rank-1 lattice searches per component. For the whole algorithm, we allow a failure probability $\varepsilon \in (0,1)$. We split this up such that each probabilistic sub-step has an equal upper bound on its failure probability of $\varepsilon/(3d)$. This strategy allows for estimates of

- the number of detection iterations r,

- the maximal number b of multiple rank-1 lattice searches per component,

First, we estimate the required number of detection iterations r.

Lemma 4.4. *Let a threshold value $\delta \geq 0$, a trigonometric polynomial $p \not\equiv 0$ with the property $\min_{\boldsymbol{h} \in \operatorname{supp}\hat{p}} |\hat{p}_{\boldsymbol{h}}| \geq 3\delta$ and the search space $\Gamma \supset \operatorname{supp}\hat{p}$ of finite cardinality be given. Choosing the number of detection iterations*

$$r \geq 2|\operatorname{supp}\hat{p}|(\log 3 + \log d + \log|\operatorname{supp}\hat{p}| - \log \varepsilon) \tag{4.7}$$

in Algorithm 3 guarantees that each of the probabilities $\mathbb{P}(A_t^{\complement})$ and $\mathbb{P}(C_t^{\complement})$ is bounded from above by $\varepsilon/(3d)$.

Proof. We estimate the probability $\mathbb{P}(C_t^{\complement})$ by (4.5) and we increase r such that

$$\mathbb{P}\left(\bigcup_{\boldsymbol{k} \in \mathcal{P}_{(1,\ldots,t)}(\operatorname{supp}\hat{p}) \cap \left(I^{(1,\ldots,t-1)} \times I^{(t)}\right)} \left\{ \max_{\nu=1,\ldots,r} |\tilde{\hat{p}}^{\Lambda}_{(1,\ldots,t),\boldsymbol{k}}(\tilde{\boldsymbol{x}}_\nu)| < \delta \right\} \right)$$

$$\leq |\operatorname{supp}\hat{p}| \left(\max_{\boldsymbol{k} \in \mathcal{P}_{(1,\ldots,t)}(\operatorname{supp}\hat{p}) \cap \left(I^{(1,\ldots,t-1)} \times I^{(t)}\right)} q_{(1,\ldots,t),\boldsymbol{k}} \right)^r \leq \frac{\varepsilon}{3d}.$$

is fulfilled. Consequently r has to be bounded from below by

$$\frac{\log 3 + \log d + \log|\operatorname{supp}\hat{p}| - \log \varepsilon}{-\log \max_{\boldsymbol{k} \in \mathcal{P}_{(1,\ldots,t)}(\operatorname{supp}\hat{p}) \cap \left(I^{(1,\ldots,t-1)} \times I^{(t)}\right)} q_{(1,\ldots,t),\boldsymbol{k}}}. \tag{4.8}$$

We denote the index set $\tilde{I} := \mathcal{P}_{(1,\ldots,t)}(\operatorname{supp}\hat{p}) \cap \left(I^{(1,\ldots,t-1)} \times I^{(t)}\right)$ and we estimate

$$\left(\max_{\boldsymbol{k}\in\tilde{I}} q_{(1,\ldots,t),\boldsymbol{k}}\right)^{-1}$$

$$\overset{(4.4)}{=} \min_{\boldsymbol{k}\in\tilde{I}} \frac{\sum_{\boldsymbol{h}=(\boldsymbol{k},h_{t+1},\ldots,h_d)\in\operatorname{supp}\hat{p}} |\hat{p}_{\boldsymbol{h}}|}{\sum_{\boldsymbol{h}=(\boldsymbol{k},h_{t+1},\ldots,h_d)\in\operatorname{supp}\hat{p}} |\hat{p}_{\boldsymbol{h}}| - \max_{\boldsymbol{h}=(\boldsymbol{k},h_{t+1},\ldots,h_d)\in\operatorname{supp}\hat{p}} |\hat{p}_{\boldsymbol{h}}| + \delta}$$

$$= 1 + \min_{\boldsymbol{k}\in\tilde{I}} \frac{\max_{\boldsymbol{h}=(\boldsymbol{k},h_{t+1},\ldots,h_d)\in\operatorname{supp}\hat{p}} |\hat{p}_{\boldsymbol{h}}| - \delta}{\sum_{\boldsymbol{h}=(\boldsymbol{k},h_{t+1},\ldots,h_d)\in\operatorname{supp}\hat{p}} |\hat{p}_{\boldsymbol{h}}|}$$

$$\geq 1 + \min_{\boldsymbol{k}\in\tilde{I}} \frac{2\max_{\boldsymbol{h}=(\boldsymbol{k},h_{t+1},\ldots,h_d)\in\operatorname{supp}\hat{p}} |\hat{p}_{\boldsymbol{h}}|}{3\sum_{\boldsymbol{h}=(\boldsymbol{k},h_{t+1},\ldots,h_d)\in\operatorname{supp}\hat{p}} |\hat{p}_{\boldsymbol{h}}|} \geq 1 + \frac{2}{3|\operatorname{supp}\hat{p}|} = \frac{3|\operatorname{supp}\hat{p}| + 2}{3|\operatorname{supp}\hat{p}|}.$$

Consequently, we obtain

$$\frac{1}{-\log\left(\max_{\boldsymbol{k}\in\tilde{I}} q_{(1,\ldots,t),\boldsymbol{k}}\right)} = \frac{1}{\log\left(\left(\max_{\boldsymbol{k}\in\tilde{I}} q_{(1,\ldots,t),\boldsymbol{k}}\right)^{-1}\right)}$$

$$\leq \frac{1}{\log\left(\frac{3|\operatorname{supp}\hat{p}|+2}{3|\operatorname{supp}\hat{p}|}\right)} = \frac{1}{\log(3|\operatorname{supp}\hat{p}| + 2) - \log(3|\operatorname{supp}\hat{p}|)} < 2|\operatorname{supp}\hat{p}|.$$

Choosing r as in (4.7) satisfies the lower bound (4.8) and $\mathbb{P}(C_t^\complement) \leq \varepsilon/(3d)$ is fulfilled. The bound $\mathbb{P}(A_t^\complement) \leq \varepsilon/(3d)$ follows analogously. ∎

Next, we deal with the choice of the maximal number b of multiple rank-1 lattice searches per component.

Lemma 4.5. *Let a frequency index set $\tilde{I} \subset \mathbb{Z}^d$, $|\tilde{I}| < \infty$, be given. We apply Algorithm 1 on $\tilde{I}$ with fixed minimal oversampling factor $c > 1$ and fixed failure probability $\gamma \in (0,1)$ in order to try determining a reconstructing multiple rank-1 lattice $\Lambda = \Lambda(\boldsymbol{z}_1, M_1, \ldots, \boldsymbol{z}_L, M_L)$. We repeatedly perform this at most $b \in \mathbb{N}$ many times and stop if Λ fulfills properties (2.1). Then, the total lattice size $M = |\Lambda(\boldsymbol{z}_1, M_1, \ldots, \boldsymbol{z}_L, M_L)|$ is in $\mathcal{O}\left(\max\{|\tilde{I}|, N_{\tilde{I}}\}\log|\tilde{I}|\right)$, and the total number of arithmetic operations is in $\mathcal{O}\left(b \max\{|\tilde{I}|, N_{\tilde{I}}\}(\log|\tilde{I}| + d + \log\log N_{\tilde{I}})\log|\tilde{I}|\right)$, where $N_{\tilde{I}} := \max_{j=1,\ldots,d}\{\max_{\boldsymbol{k}\in\tilde{I}} k_j - \min_{\boldsymbol{l}\in\tilde{I}} l_j\}$ is the expansion of $\tilde{I}$. Moreover,*

$$b = \left\lceil \frac{\log 3 + \log d - \log \varepsilon}{|\log\gamma|} \right\rceil$$

calls of Algorithm 1 guarantee that Λ is a reconstructing multiple rank-1 lattice $\Lambda(\boldsymbol{z}_1, M_1, \ldots, \boldsymbol{z}_L, M_L)$ with probability at least $1 - \varepsilon/(3d)$, $\varepsilon \in (0,1)$.

Proof. With probability at least $1 - \gamma$, $\Lambda(\boldsymbol{z}_1, M_1, \ldots, \boldsymbol{z}_L, M_L)$ returned by Algorithm 1 is a reconstructing multiple rank-1 lattice for $\tilde{I}$. From (2.3) and (2.4) for fixed γ, we immediately obtain the claimed upper bound for the total lattice size M and the number of arithmetic operations, respectively.

Repeatedly applying Algorithm 1 in case of failure up to totally $\tilde{b}$ times results in a success property of at least $1 - \gamma^{\tilde{b}}$. Consequently, ensuring a success property of at least $1 - \varepsilon/(3d)$ yields the claimed upper bound b on the number of calls. ∎

Finally, we combine the obtained results in order to estimate the sample and arithmetic complexity of Algorithm 3 for a given success probability $1 - \varepsilon$.

Theorem 4.6. *Let a failure probability $\varepsilon \in (0,1)$ be given. We apply Algorithm 3 on a multivariate trigonometric polynomial p using the following parameters. We choose*

- *the search space Γ such that $\operatorname{supp}\hat{p} \subset \Gamma \subset \hat{G}_N^d$,*

- *an absolute threshold $0 < \delta \le \min_{\boldsymbol{h}\in\operatorname{supp}\hat{p}} |\hat{p}_{\boldsymbol{h}}|/3$,*

- *the sparsity parameter $s \ge |\operatorname{supp}\hat{p}|$,*

- *the number of detection iterations $r := \lceil 2|\operatorname{supp}\hat{p}|(\log 3 + \log d + \log|\operatorname{supp}\hat{p}| - \log\varepsilon)\rceil$,*

- *the number of multiple rank-1 lattice searches $b := \lceil(\log 3 + \log d - \log\varepsilon)/\log 2\rceil$.*

Moreover, we assume $|\operatorname{supp}\hat{p}| \gtrsim N$ and $|\operatorname{supp}\hat{p}| \gtrsim d$. Then, with probability $1-\varepsilon$, the output of the index set I of Algorithm 3 is $\operatorname{supp}\hat{p}$, the total number of sampling nodes is in $\mathcal{O}\left(d\,|\operatorname{supp}\hat{p}|^2 N(\log|\operatorname{supp}\hat{p}|)^2|\log\varepsilon|\right)$, and the total number of arithmetic operations is in $\mathcal{O}\left(d^2|\operatorname{supp}\hat{p}|^2 N(\log|\operatorname{supp}\hat{p}|)^3|\log\varepsilon|\right)$.
When choosing the sparsity parameter $s \le C\,|\operatorname{supp}\hat{p}|$ for a constant $C \ge 1$, we always require $\lesssim d\,|\operatorname{supp}\hat{p}|^3 N(\log|\operatorname{supp}\hat{p}|)^3|\log\varepsilon|\,|\log|\log\varepsilon||$ many samples and $\lesssim d^2|\operatorname{supp}\hat{p}|^3 N(\log|\operatorname{supp}\hat{p}|)^4|\log\varepsilon|^2\,|\log|\log\varepsilon||^2$ arithmetic operations.

Proof. As discussed above, we use the union bound to estimate

$$\mathbb{P}\left(\operatorname{supp}\hat{p} \subset I\right) \ge \mathbb{P}\left(\bigcap_{t=1}^{d} A_t \cap \bigcap_{t=2}^{d} B_t \cap \bigcap_{t=2}^{d} C_t\right) \overset{(4.6)}{\ge} 1 - \sum_{t=1}^{d}\mathbb{P}(A_t^{\complement}) - \sum_{t=2}^{d}\mathbb{P}(B_t^{\complement}) - \sum_{t=2}^{d}\mathbb{P}(C_t^{\complement})$$

for the output I of Algorithm 3. From Lemma 4.4, we obtain $\mathbb{P}(A_t^{\complement}) \ge \varepsilon/(3d)$ and $\mathbb{P}(C_t^{\complement}) \ge \varepsilon/(3d)$. Moreover, we have $\mathbb{P}(B_t^{\complement}) \ge \varepsilon/(3d)$ due to Lemma 4.5. Altogether, this yields $\mathbb{P}\left(\operatorname{supp}\hat{p} \subset I\right) \ge 1 - \varepsilon$. Next, we discuss the complexities.
In the general case, the number of frequencies within $I^{(1,\dots,t)}$ is bounded from above by rs, cf. Section 3.2. We show that a successful application of Algorithm 3 on the multivariate trigonometric polynomial p does not detect more than $|\operatorname{supp}\hat{p}|$ different frequencies in each dimension increment step. In particular, we proof by contradiction that $I^{(t)} = \mathcal{P}_t(\operatorname{supp}\hat{p})$ and $I^{(1,\dots,t)} = \mathcal{P}_{(1,\dots,t)}(\operatorname{supp}\hat{p})$ holds.
First, we consider the construction of the frequency index sets $I^{(t)}$, $t = 1,\dots,d$, i.e., steps 1 and 2a. We assume that there exists $k_t \notin \mathcal{P}_t(\operatorname{supp}\hat{p})$ such that $|\tilde{\hat{p}}_{t,k_t}| \ge \delta > 0$,

$$\tilde{\hat{p}}_{t,k_t} \overset{(4.2)}{:=} \sum_{\ell=0}^{K_t-1} p\left(x_1', \dots, x_{t-1}', \frac{\ell}{K_t}, x_{t+1}', \dots, x_d'\right) \mathrm{e}^{-2\pi\mathrm{i}\ell k_t/K_t}$$

$$= \sum_{\substack{\boldsymbol{h}\in\operatorname{supp}\hat{p}\\ h_t\equiv k_t\ (\mathrm{mod}\ K_t)}} \hat{p}_{\boldsymbol{h}}\, \mathrm{e}^{2\pi\mathrm{i}(h_1,\dots,h_{t-1},h_{t+1},\dots,h_d)\cdot\tilde{\boldsymbol{x}}},$$

holds, which implies that there exists at least one $h_t \in \mathbb{Z}\setminus\{k_t\}$, such that $\hat{p}_{\boldsymbol{h}} \ne 0$ and $h_t \equiv k_t$ (mod K_t). Since $K_t := \max(\mathcal{P}_t(\Gamma)) - \min(\mathcal{P}_t(\Gamma)) + 1$ and $k_t \in \mathcal{P}_t(\Gamma)$, we observe $h_t \notin \mathcal{P}_t(\Gamma)$ and this yields $\Gamma \not\supset \operatorname{supp}\hat{p}$, which is in contradiction to the choice of Γ. Consequently, the inclusion $I^{(t)} \subset \mathcal{P}_t(\operatorname{supp}\hat{p})$ holds and with probability at least $1 - \varepsilon/(3d)$, we have the equality $I^{(t)} = \mathcal{P}_t(\operatorname{supp}\hat{p})$.
Second, in step 2e of Algorithm 3, the frequency index sets $I^{(1,\dots,t)}$ are constructed. For

the index set of frequency candidates $J_t := (I^{(1,\ldots,t-1)} \times I^{(t)}) \cap \mathcal{P}_{(1,\ldots,t)}(\Gamma)$, we require $J_t \supset \mathcal{P}_{(1,\ldots,t)}(\operatorname{supp}\hat{p})$ and that Λ is a reconstructing multiple rank-1 lattice for J_t fulfilling condition (2.1). Then, for each $\boldsymbol{k} \in J_t$ fulfilling $|\hat{\tilde{p}}^{\Lambda}_{(1,\ldots,t),\boldsymbol{k}}| \geq \delta > 0$, we have $\boldsymbol{k} \in \mathcal{P}_{(1,\ldots,t)}(\operatorname{supp}\hat{p})$. We show this by contradiction and for this, we assume that there exists a frequency $\boldsymbol{k} \in J_t \setminus \mathcal{P}_{(1,\ldots,t)}(\operatorname{supp}\hat{p})$ with $|\hat{\tilde{p}}^{\Lambda}_{(1,\ldots,t),\boldsymbol{k}}| \geq \delta > 0$. The projected Fourier coefficient $\hat{\tilde{p}}^{\Lambda}_{(1,\ldots,t),\boldsymbol{k}}$ returned by Algorithm 2 is computed as an arithmetic mean of at least one coefficient

$$
\begin{aligned}
\hat{\tilde{p}}^{\Lambda(\boldsymbol{z}_\ell, M_\ell)}_{(1,\ldots,t),\boldsymbol{k}} &:= \frac{1}{M_\ell} \sum_{j=1}^{M_\ell - 1} p\left(\frac{j}{M_\ell}\boldsymbol{z}_\ell, \tilde{\boldsymbol{x}}\right) \mathrm{e}^{-2\pi\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{z}_\ell \frac{j}{M_\ell}} \\
&= \sum_{\substack{(\boldsymbol{k}',h_{t+1},\ldots,h_d)\in\operatorname{supp}\hat{p} \\ \boldsymbol{k}'\cdot\boldsymbol{z}_\ell \equiv \boldsymbol{k}\cdot\boldsymbol{z}_\ell \pmod{M_\ell}}} \hat{p}_{(\boldsymbol{k}',h_{t+1},\ldots,h_d)} \mathrm{e}^{2\pi\mathrm{i}(h_{t+1},\ldots,h_d)\cdot\tilde{\boldsymbol{x}}}.
\end{aligned}
$$

Since $\hat{\tilde{p}}^{\Lambda}_{(1,\ldots,t),\boldsymbol{k}} \neq 0$, there exists at least one ℓ such that $\hat{\tilde{p}}^{\Lambda(\boldsymbol{z}_\ell, M_\ell)}_{(1,\ldots,t),\boldsymbol{k}} \neq 0$, and thus, there exists at least one $\hat{p}_{(\boldsymbol{k}',h_{t+1},\ldots,h_d)} \neq 0$, where $(\boldsymbol{k}',h_{t+1},\ldots,h_d) \in \operatorname{supp}\hat{p}$ and $\boldsymbol{k}'\cdot\boldsymbol{z}_\ell \equiv \boldsymbol{k}\cdot\boldsymbol{z}_\ell \pmod{M_\ell}$. On the one hand, due to the assumption $\boldsymbol{k} \notin \mathcal{P}_{(1,\ldots,t)}(\operatorname{supp}\hat{p})$, we have $\boldsymbol{k}' \neq \boldsymbol{k}$. On the other hand, Λ is a reconstructing multiple rank-1 lattice constructed by Algorithm 1 fulfilling the properties (2.1). Consequently, we obtain $\boldsymbol{k}' \notin J_t$, which is in contradiction to the requirement $J_t \supset \mathcal{P}_{(1,\ldots,t)}(\operatorname{supp}\hat{p})$.

If all events A_1, and A_t, B_t, C_t for $t \in \{2,\ldots,d\}$ occur, the equalities $I^{(1)} = \mathcal{P}_1(\operatorname{supp}\hat{p})$, $I^{(t)} = \mathcal{P}_t(\operatorname{supp}\hat{p})$, and $I^{(1,\ldots,t)} = \mathcal{P}_{(1,\ldots,t)}(\operatorname{supp}\hat{p})$ hold. In particular, this means that the cardinalities are bounded by $|I^{(1,\ldots,t)}| \leq |\operatorname{supp}\hat{p}|$. This leads to a distinct reduction in the upper bounds of the complexities in (3.1) and (3.2). In more detail, the factors $r\,s$ can be replaced by $|\operatorname{supp}\hat{p}|$, which leads to $\mathcal{O}\left(dr|\operatorname{supp}\hat{p}|N\log(|\operatorname{supp}\hat{p}|N)\right)$ samples and $\mathcal{O}\left(d^2\,r|\operatorname{supp}\hat{p}|N\,\log^2(|\operatorname{supp}\hat{p}|N)\right)$ arithmetic operations.

The latter discussed events occur with probability $1 - \varepsilon$ as mentioned above. Taking the assumptions $|\operatorname{supp}\hat{p}| \gtrsim N$ and $|\operatorname{supp}\hat{p}| \gtrsim d$ as well as the choice of the parameters r and b into account, we obtain with probability at least $1 - \varepsilon$ that the number of samples is in $\mathcal{O}\left(d|\operatorname{supp}\hat{p}|^2 N\log^2|\operatorname{supp}\hat{p}||\log\varepsilon|\right)$ and that the number of arithmetic operations is in $\mathcal{O}\left(d^2|\operatorname{supp}\hat{p}|^2 N\,\log^3|\operatorname{supp}\hat{p}||\log\varepsilon|\right)$.

The worst case number of samples and arithmetic complexities follow from the parameter choices and the discussions in Section 3.2. $\blacksquare$

Similarly, we obtain estimates for the sample and arithmetic complexity of Algorithm 5.

Theorem 4.7. *Let a failure probability $\varepsilon \in (0,1)$ be given. We apply Algorithm 5 on a trigonometric polynomial p using the following parameters. We choose*

- *the search space Γ such that $\operatorname{supp}\hat{p} \subset \Gamma \subset \hat{G}_N^d$,*

- *an absolute threshold $0 < \delta \leq \min_{\boldsymbol{h}\in\operatorname{supp}\hat{p}} |\hat{p}_{\boldsymbol{h}}|/3$,*

- *the sparsity parameter $s \geq |\operatorname{supp}\hat{p}|$,*

- *the number of detection iterations $r := \lceil 2|\operatorname{supp}\hat{p}|(\log 2 + \log d + \log|\operatorname{supp}\hat{p}| - \log\varepsilon)\rceil$.*

Moreover, we assume $|\operatorname{supp}\hat{p}| \gtrsim N$ *and* $|\operatorname{supp}\hat{p}| \gtrsim d$. *Then, with probability* $1 - \varepsilon$, *the output* I *of Algorithm 3 is* $\operatorname{supp}\hat{p}$, *the total number of sampling nodes is in* $\mathcal{O}\left(d\,|\operatorname{supp}\hat{p}|^{3}N\left(\log|\operatorname{supp}\hat{p}|\right)|\log\varepsilon|\right)$, *and the total number of arithmetic operations is in* $\mathcal{O}\left(d\,|\operatorname{supp}\hat{p}|^{3}N\left(\log|\operatorname{supp}\hat{p}|\right)^{2}|\log\varepsilon|\right)$.

When choosing the sparsity parameter $s \leq C\,|\operatorname{supp}\hat{p}|$ *for a constant* $C \geq 1$, *we always require* $\lesssim d\,|\operatorname{supp}\hat{p}|^{5}N(\log|\operatorname{supp}\hat{p}|)^{3}|\log\varepsilon|^{3}$ *many samples and* $\lesssim d\,|\operatorname{supp}\hat{p}|^{6}\left(\log|\operatorname{supp}\hat{p}|\right)^{3}|\log\varepsilon|^{3} + d\,|\operatorname{supp}\hat{p}|^{5}N(\log|\operatorname{supp}\hat{p}|)^{4}|\log\varepsilon|^{3}\,|\log|\log\varepsilon||$ *arithmetic operations.*

Proof. We proceed similarly as in the proof of Theorem 4.6. The main difference is that we use reconstructing single rank-1 lattices, which are constructed by a deterministic algorithm. Consequently, we have

$$\mathbb{P}\left(\operatorname{supp}\hat{p} \subset I\right) \geq \mathbb{P}\left(\bigcap_{t=1}^{d}A_{t} \cap \bigcap_{t=2}^{d}C_{t}\right) \overset{(4.6)}{\geq} 1 - \sum_{t=1}^{d}\mathbb{P}(A_{t}^{\complement}) - \sum_{t=2}^{d}\mathbb{P}(C_{t}^{\complement})$$

for the output I of Algorithm 3. For each probability $\mathbb{P}(A_{t}^{\complement})$ and $\mathbb{P}(C_{t}^{\complement})$, we ensure an lower bound of $\varepsilon/(2d)$. Consequently, we have a slightly smaller choice for the number of detection iterations r in Lemma 4.4.

For the number of samples, the dominating terms are the sizes $M^{(t)}$ of the single rank-1 lattices in each dimension increment step $t \in \{2, \dots, d\}$. If all of the previous steps succeeded, we have $M^{(t)} \lesssim |\operatorname{supp}\hat{p}|^{2}N$ by [26, Corollary 2.3] as a consequence of [14]. In total, this means

$$\lesssim d\,r\,|\operatorname{supp}\hat{p}|^{2}N \lesssim d\,|\operatorname{supp}\hat{p}|^{3}N\left(\log|\operatorname{supp}\hat{p}|\right)|\log\varepsilon|$$

many samples with probability at least $1 - \varepsilon$.

Building the reconstructing single rank-1 lattices requires $\mathcal{O}\left(d\,|\operatorname{supp}\hat{p}|^{3}\right)$ arithmetic operations in total and computing the rank-1 lattice FFTs $\mathcal{O}\left(d\,r\,|\operatorname{supp}\hat{p}|^{2}N\log|\operatorname{supp}\hat{p}|\right)$ many in total with probability at least $1 - \varepsilon$, which yields the claimed arithmetic complexity $\mathcal{O}\left(d\,|\operatorname{supp}\hat{p}|^{3}N\left(\log|\operatorname{supp}\hat{p}|\right)^{2}|\log\varepsilon|\right)$.

Since Algorithm 5 requires $\lesssim dr^{3}s^{2}N$ samples and $\lesssim dr^{3}s^{3} + dr^{3}s^{2}N\log(rsN)$ arithmetic operations, cf. [26, Section 2.2.3], the worst case complexities follow due to the parameter choices of r and s. ∎

4.2 Discussion

In Section 4.1, we obtained detailed theoretical estimates for the sample and arithmetic complexities of Algorithm 3 and 5 when applied on multivariate trigonometric polynomials p. Starting point of the theoretical results was Lemma 4.1, which estimates the probability that the absolute value of a multivariate trigonometric polynomial is below or equal to a certain threshold δ for a uniformly random sample. Afterwards, this result was used in Corollary 4.2 and 4.3 to bound the probability that the absolute value of a single aliased Fourier coefficient in Algorithm 3 or 5 is below or equal to a given absolute threshold δ, which would mean that the corresponding frequency component is not detected. Using a certain number of detection iterations r, repeated sampling can be used to decrease the overall failure probability and increase the overall probability of successfully detecting all non-zero Fourier coefficients $\hat{p}_{\boldsymbol{k}} \neq 0$ of the multivariate trigonometric polynomial p under consideration. Based on the previously mentioned results, the required number of detection iterations r was estimated by

$$r \geq 2|\operatorname{supp}\hat{p}|(\log 3 + \log d + \log|\operatorname{supp}\hat{p}| - \log\varepsilon)$$

in Lemma 4.4 for a given failure probability $\leq \varepsilon \in (0,1)$. This lower bound for r contains the number $|\operatorname{supp}\hat{p}|$ of non-zero Fourier coefficients as a linear factor which is probably overestimated. Moreover, even if we would decrease the absolute threshold δ to almost zero, the currently used proof technique would still result in a lower bound which contains $|\operatorname{supp}\hat{p}|$ as a linear factor. However, this does not match our expectations and the observed numerical results in Section 5.

We remark that Lemma 4.1 still leads to a distinct improvement compared to the previous results based on [26, Lemma 2.4], which would yield an even larger bound on r containing the factor $|\operatorname{supp}\hat{p}|^2$ in Section 4.1.

In order to obtain a smaller bound on r and, consequently, lower sample and arithmetic complexities, an improved version of Lemma 4.1 would be required. For instance, an answer to the following more general question for trigonometric polynomials would immediately give such an improvement. Assume that a trigonometric polynomial $g\colon \mathbb{T}^n \to \mathbb{C}$, $n \in \mathbb{N}$, is given and let

$$q := \int_{\{\boldsymbol{x} \in \mathbb{T}^n \,:\, |g(\boldsymbol{x})| \leq \delta\}} \mathrm{d}\,\boldsymbol{x} = \left| \left\{ \tilde{\boldsymbol{x}} \in \mathbb{T}^n \colon |g(\tilde{\boldsymbol{x}})| = \left| \sum_{\boldsymbol{k} \in \operatorname{supp}\hat{g}} \hat{g}_{\boldsymbol{k}}\, \mathrm{e}^{2\pi\mathrm{i}\boldsymbol{h}\cdot\tilde{\boldsymbol{x}}} \right| \leq \delta \right\} \right|.$$

How is it possible to show

$$\frac{1}{\log\frac{1}{q}} \lll \sim |\operatorname{supp}\hat{g}|$$

and which conditions on the Fourier coefficients $\hat{g}_{\boldsymbol{k}}$ are sufficient. Of course, these conditions should be as weak as possible.

As already mentioned in [26], assuming a special property of the Fourier coefficients $\hat{p}_{\boldsymbol{k}}$ of a multivariate trigonometric polynomial p allows for choosing the number of detection iterations $r := 1$ as we discuss in the next section. However, for a more general case, we currently do not know how to obtain improved bounds on r.

4.3 Multivariate trigonometric polynomial with special Fourier coefficients allowing for $r = 1$ detection iteration

If we assume that the function under consideration is a multivariate trigonometric polynomial p with the additional property that the signs of the real part $\operatorname{Re}(\hat{p}_{\boldsymbol{k}})$ of all non-zero Fourier coefficients $\hat{p}_{\boldsymbol{k}}$ are identical $\neq 0$ or that the signs of the imaginary part $\operatorname{Im}(\hat{p}_{\boldsymbol{k}})$ are identical $\neq 0$, then we obtain a deterministic version of Algorithm 5 as explained in Remark 3.2. This strategy requires $\mathcal{O}\big(d\,|\operatorname{supp}\hat{p}|^2 N\big)$ samples and $\mathcal{O}\big(d\,|\operatorname{supp}\hat{p}|^3 + d\,|\operatorname{supp}\hat{p}|^2 N\big)$ arithmetic operations.

In this setting, Algorithm 3 is not purely deterministic due to the probabilistic search for the reconstructing multiple rank-1 lattices. From the proof of Theorem 4.6, we obtain that Algorithm 3 detects all non-zero Fourier coefficients with probability at least $1 - \varepsilon$ when setting the number of multiple rank-1 lattice searches parameter $b := \lceil (\log d - \log \varepsilon)/\log 2 \rceil$ and the sparsity $s \geq |\operatorname{supp}\hat{p}|$. This requires $\mathcal{O}\big(d\,|\operatorname{supp}\hat{p}|N\log(|\operatorname{supp}\hat{p}|\,N)\big)$ samples and $\mathcal{O}\big(d^2\,|\operatorname{supp}\hat{p}|N\log^2(|\operatorname{supp}\hat{p}|\,N)\,(\log(d) - \log\varepsilon)\big)$ arithmetic operations with probability at least $1 - \varepsilon$. Choosing the sparsity parameter $s \leq C\,|\operatorname{supp}\hat{p}|$ for a constant $C \geq 1$ guarantees these complexities.

5 Numerical results

In this section, we present several numerical results, which empirically confirm the high effectiveness and robustness of the proposed method. In Section 5.1, we start with the reconstruction of random multivariate trigonometric polynomials p in up to $d = 30$ spatial dimensions, where the sampling values are not perturbed by noise. Additionally, we consider the case, where the sampling values are perturbed by additive Gaussian noise in Section 5.2. Moreover, in Section 5.3, we successfully determine the approximately largest Fourier coefficients and the corresponding frequencies of a 10-dimensional test function.

In all numerical tests, we compare the new algorithms based on multiple rank-1 lattice sampling with Algorithm 5, which uses single reconstructing rank-1 lattices. On the one hand, we consider Algorithm 3 to confirm theoretical results from Section 4. On the other hand, all numerical test settings will be treated by Algorithm 4, for which we currently do not have an extensive theoretical framework. The main difference in both algorithms is the construction of the multiple rank-1 lattices. The construction method used in Algorithm 4 does not allow for the application of the proof techniques we developed in Section 4 and may result in higher theoretical upper bounds on the sampling and arithmetic complexities, see also the discussion in [16, Section 4.1]. However, for the numerical test settings in [16], the number of sampling nodes was distinctly lower, when using this construction method, compared to the construction method used in Algorithm 3. Since we observe a similar preferable behavior when using Algorithm 4, we also present the corresponding results in detail.

We implemented the methods and numerical tests in MATLAB. All tests were run using MATLAB R2015b and IEEE 754 double precision arithmetic. The time measurements were performed on a computer with Intel Xeon 5160 CPU (3 GHz) and 64 GB RAM using 1 thread.

In general, we observe that the total number of samples is distinctly lower when using multiple rank-1 lattices instead of single rank-1 lattices.

We remark that when performing the numerical tests, we implicitly assumed that we have at least some basic knowledge about the function under consideration, like e.g. a reasonable search domain Γ or estimates of the minimal absolute value of the Fourier coefficients $\hat{p}_{\boldsymbol{k}}$, the sparsity $|\operatorname{supp}\hat{p}|$, the noise level, etc.

5.1 Random sparse trigonometric polynomial

As in [26, Section 3.1], we construct random multivariate trigonometric polynomials p with frequencies supported within the cube $\hat{G}_N^d = [-N, N]^d \cap \mathbb{Z}^d$. For this, we choose $|\operatorname{supp}\hat{p}|$ frequencies $\boldsymbol{k} \in \hat{G}_N^d$ uniformly at random and corresponding Fourier coefficients $\hat{p}_{\boldsymbol{k}} \in [-1, 1) + [-1, 1)\mathrm{i}$, $|\hat{p}_{\boldsymbol{k}}| \geq 10^{-6}$, $\boldsymbol{k} \in I = \operatorname{supp}\hat{p}$. For the reconstruction of the trigonometric polynomials p, we only assume that the search domain $\Gamma := \hat{G}_N^d \supset \operatorname{supp}\hat{p}$. Additionally, we do not truncate the frequency index sets of detected frequencies $I^{(1,\ldots,t)}$, $t \in \{2, \ldots, d\}$, i.e., we set the sparsity parameter $s := |\Gamma|$. Alternatively, one may choose $s := |\operatorname{supp}\hat{p}|$ and this yields identical results for the considered random sparse trigonometric polynomials.

Example 5.1. (Number of samples and errors). We set the expansion $N := 32$. Now, we compare the results of Algorithm 5, which is a modification of [26, Algorithm 2] with absolute thresholding and which uses reconstructing single rank-1 lattices, with Algorithm 3 and 4, which use multiple rank-1 lattices as sampling sets. We set the absolute threshold parameter $\delta := 10^{-12}$ and the number of detection iterations $r := 1$. For sparsity $|\operatorname{supp}\hat{p}| := 1\,000$ and $10\,000$, we run tests for dimension $d \in \{5, 10, 15, 20, 25, 30\}$. All tests are repeated 10

times with newly chosen frequencies $\boldsymbol{k} \in \Gamma$ and Fourier coefficients $\hat{p}_{\boldsymbol{k}} \in \mathbb{C}$. Then, for the 10 repetitions, we determine the maximum of the total number of samples and the maximum of the relative ℓ_2-errors of the Fourier coefficients. For Algorithm 5 using reconstructing single rank-1 lattices, the results are shown in columns 3–4 of Table 5.1. Moreover, the results for Algorithm 3 and 4 using multiple rank-1 lattices are presented in columns 5–6 and 7–8, respectively. For multiple rank-1 lattices, we also run the tests for sparsity $|\mathrm{supp}\,\hat{p}| :=$ 100 000. We observe that in the considered cases, the obtained relative ℓ_2-errors are near machine precision. Moreover, we require distinctly less samples than if we had used a d-dimensional FFT on a full grid, which would require $|\hat{\Gamma}_{32}^5| = 1\,160\,290\,625$ already in the 5-dimensional case. Additionally, for sparsity $|\mathrm{supp}\,\hat{p}| := 10\,000$, Algorithm 3 using multiple rank-1 lattices required only about 1/9 of the samples compared to Algorithm 5 using single rank-1 lattices. We observe that the number of samples may be even further reduced by using Algorithm 4, which required only 1/84 to 1/41 of the samples compared to single rank-1 lattices for sparsity $|\mathrm{supp}\,\hat{p}| := 10\,000$. For sparsity $|\mathrm{supp}\,\hat{p}| := 100\,000$, we were not able to apply Algorithm 5, which uses single rank-1 lattices, since this would have required too many samples and extremely long runtimes.

Instead of choosing the absolute threshold parameter $\delta := 10^{-12}$, we also repeated the tests for a larger absolute threshold of $\delta := 10^{-7} \le \min_{\boldsymbol{k} \in \mathrm{supp}\,\hat{p}} |\hat{p}_{\boldsymbol{k}}|/10$ and we obtained almost identical results as in Table 5.1.

We remark that we also ran the numerical tests applying a modified version of Algorithm 3, which uses [16, Algorithm 4] instead of Algorithm 1 for determining reconstructing multiple rank-1 lattices. Then, we obtained results comparable to those of Algorithm 3 in Table 5.1 requiring slightly less samples, i.e., between 2 and 10 percent less in most cases. □

From the previous example, we observe that for fixed dimension d and increasing sparsity $|\mathrm{supp}\,\hat{p}|$, the number of samples increases only moderately for Algorithm 3 and 4. In the next example, we investigate the behavior in more detail and additionally measure the runtimes using only one CPU thread.

Example 5.2. (Number of samples and runtimes with respect to sparsity). As in Example 5.1, we set the expansion $N := 32$, the threshold parameter $\delta := 10^{-12}$ and the number of detection iterations $r := 1$. Since we want to compare our methods with a full d-dimensional FFT, we restrict ourselves to the 5-dimensional case, which already yields a search space $\Gamma = \hat{G}_{32}^5$ of cardinality $|\Gamma| = 65^5 = 1\,160\,290\,625$. For sparsity $s := |\mathrm{supp}\,\hat{p}| \in \{1\,000, 2\,000, 5\,000, 10\,000, 20\,000, 50\,000\}$, we determine the runtimes of Algorithm 5 as well as the runtimes of Algorithm 3 and 4. For Algorithm 3 and 4, we additionally determine the runtimes for sparsity $s = 100\,000$. For each of the considered cases, we subtract the time required for sampling. We repeat the tests 10 times and capture the maximum of the obtained numbers of samples and runtimes. The results are depicted in Figure 5.1. In Figure 5.1a, we show the maximal number of samples with respect to the sparsity s. We denote the results of Algorithm 5 using single rank-1 lattices by "A2R1L", of Algorithm 3 using multiple rank-1 lattices by "MLFFT" and of Algorithm 4 using multiple rank-1 lattices by "MLFFT6". Additionally, we plot the number of samples required by a 5-dimensional FFT on a full grid and denote this by "5d FFT". We observe that the number of samples for Algorithm 5 using rank-1 lattices increases roughly like $\sim s^{1.5}$ in this example, which is less than the theoretical results of $\mathcal{O}(s^2)$ suggest. For sparsity $s = 50\,000$, the number of samples almost equals the cardinality of the search space $|\hat{G}_{32}^5|$, which is the number of samples required by a 5-dimensional FFT. Moreover, the number of samples is distinctly lower for Algorithm 3 using multiple rank-1

d	$|\mathrm{supp}\,\hat{p}|$	Algorithm 5 using single rank-1 lattices		Algorithm 3 using multiple rank-1 lattices		Algorithm 4 using multiple rank-1 lattices	
		max. total #samples	max. rel. ℓ_2-error	max. total #samples	max. rel. ℓ_2-error	max. total #samples	max. rel. ℓ_2-error
5	1 000	6 260 605	8.0e-16	4 525 799	5.3e-16	581 881	5.8e-16
10	1 000	20 848 685	5.4e-16	12 115 199	5.3e-16	1 589 349	5.1e-16
15	1 000	35 937 525	7.2e-16	18 827 483	5.3e-16	2 599 029	5.2e-16
20	1 000	52 361 205	6.9e-16	27 030 181	5.3e-16	3 609 753	5.2e-16
25	1 000	67 164 695	5.0e-16	33 666 561	5.3e-16	4 621 205	5.1e-16
30	1 000	80 660 385	5.7e-16	40 085 363	5.3e-16	5 644 059	5.1e-16
5	10 000	190 618 285	8.6e-16	40 854 167	3.5e-16	4 648 335	4.0e-16
10	10 000	1 081 274 675	6.3e-16	129 929 699	3.4e-16	15 186 447	3.8e-16
15	10 000	1 969 412 575	1.3e-15	225 282 365	3.5e-16	25 662 189	3.8e-16
20	10 000	2 935 663 575	5.3e-16	316 957 693	3.4e-16	36 161 887	3.8e-16
25	10 000	3 837 073 825	7.1e-16	409 305 929	3.4e-16	46 681 103	3.8e-16
30	10 000	4 771 398 905	1.3e-15	492 181 865	3.5e-16	57 203 659	3.8e-16
5	100 000	–	–	419 479 177	2.2e-16	33 428 113	3.0e-16
10	100 000	–	–	1 445 678 877	2.1e-16	143 681 689	2.1e-16
15	100 000	–	–	2 572 013 451	2.1e-16	250 232 085	2.1e-16
20	100 000	–	–	3 623 019 125	2.0e-16	356 857 499	2.1e-16
25	100 000	–	–	4 678 498 335	2.1e-16	463 174 925	2.9e-16
30	100 000	–	–	5 680 886 471	2.0e-16	569 711 277	2.1e-16

Table 5.1: Results for random sparse trigonometric polynomials using Algorithm 5, 3, 4, when considering frequencies within the search domain $\Gamma = \hat{G}_{32}^d$.

lattices and dramatically lower for Algorithm 4. For both, Algorithm 3 and 4, the number of samples behaves approximately like $\sim s$ which is slightly better than the theoretical results of $\mathcal{O}(s\log s)$ and $\mathcal{O}(s\log^2 s)$ from Section 3.2, respectively.

We also compare the runtimes of the different approaches and visualize the results in Figure 5.1b. Here, we denote the runtime of the 5-dimensional FFT computed using the MAT-LAB function `fftn`, which is based on FFTW [7], by "FFTW". The observed runtime when using Algorithm 5 (single rank-1 lattices) is smaller for sparsity s up to $5\,000$ and behaves approximately like $\sim s^2\log s$, which is distinctly lower than the theoretical results of $\mathcal{O}(s^3 + s^2\log s)$ suggest. Additionally, when using Algorithm 3 (multiple rank-1 lattices, "MLFFT"), we observe that the runtimes behave approximately like $\sim s\log^2 s$, which corresponds to the theoretical results of $\mathcal{O}(s\log^2 s)$ for fixed r, d, N. In particular, these runtimes are distinctly smaller compared to using single rank-1 lattices for higher sparsity s. Furthermore, Algorithm 3 was faster than the 5-dimensional FFT for sparsities s up to $10\,000$ as well as faster than Algorithm 5 for sparsity $s = 5\,000$ and above. Moreover, the runtimes of Algorithm 4 (multiple rank-1 lattices, "MLFFT6") seem to behave slightly better than Algorithm 3, whereas the theoretical results suggest $\mathcal{O}(s\log^3 s)$ arithmetic operations.

We remark that for higher-dimensional search spaces, the direct application of a d-dimensional FFT algorithm is usually not possible in practice due to the exponential growth in the degrees of freedom. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\Box$

In the previous examples, we chose the threshold parameter δ very low compared to the minimal allowed size $|\hat{p}_{\boldsymbol{k}}| \geq 10^{-6}$ of the Fourier coefficients $\hat{p}_{\boldsymbol{k}}$. Next, we consider larger
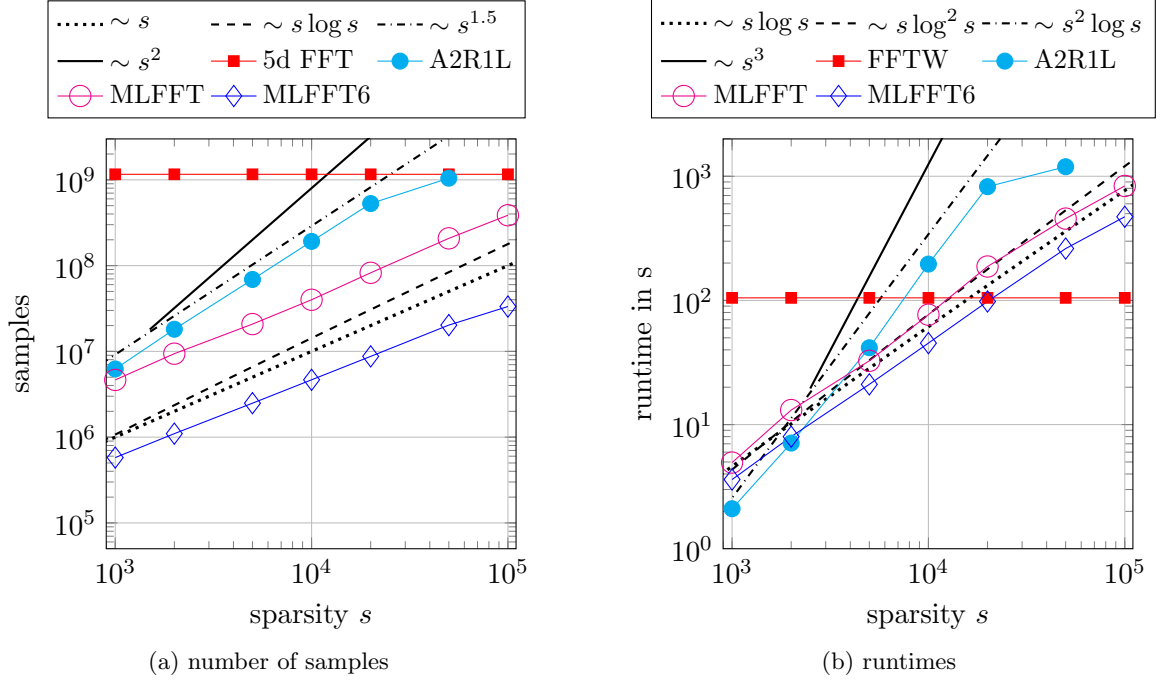
Figure 5.1: Number of samples and runtimes with respect to the sparsity for the reconstruction of random 5-dimensional sparse trigonometric polynomials when considering frequencies within $\Gamma = \hat{G}_{32}^5$, $r = 1$.

threshold parameters δ and observe that setting the number of detection iterations $r := 1$ does not always suffice. Still, the observed results are very good.

Example 5.3. (Absolute threshold parameter δ and number of detection iterations r). We set the expansion $N := 32$ and the dimension $d := 5$. In contrast to the previous examples, we use Fourier coefficients $\hat{p}_{\boldsymbol{k}} := \mathrm{e}^{2\pi\mathrm{i}\varphi_{\boldsymbol{k}}} \in \mathbb{C}$ with angles $\varphi_{\boldsymbol{k}} \in [0,1)$ chosen uniformly at random. For sparsity $|\operatorname{supp} \hat{p}| := 1\,000$ and $1\,000$ repetitions with newly chosen random frequencies $\boldsymbol{k} \in \Gamma := \hat{G}_{32}^5$ and Fourier coefficients $\hat{p}_{\boldsymbol{k}}$, we investigate the minimal number of correctly detected frequencies and the success rate for correctly detecting all frequencies of Algorithm 3, 4 and 5 for different choices of the absolute threshold parameter δ and number of detection iterations r. The results are depicted in Table 5.2. We observe that for increasing number of detection iterations and decreasing threshold parameters δ, the minimal number of correctly detected frequencies and the success rate for correctly detecting all frequencies increase. The three considered algorithms perform similarly well. For instance, setting $r := 3$ and $\delta := 10^{-2}$, all $1\,000$ frequencies for all $1\,000$ repetitions were successfully detected by each of the three algorithms. $\qquad\square$

5.2 Random sparse trigonometric polynomial with complex Gaussian noise

In this subsection, we test the robustness to noise. To this end, we build sparse multivariate trigonometric polynomials p with frequencies $\boldsymbol{k}$ randomly chosen from the cube $\hat{G}_N^d := [-N, N]^d \cap \mathbb{Z}^d$ and corresponding Fourier coefficients $\hat{p}_{\boldsymbol{k}} := \mathrm{e}^{2\pi\mathrm{i}\varphi_{\boldsymbol{k}}} \in \mathbb{C}$ with angles

δ	Algorithm 5 using single rank-1 lattices			Algorithm 3 using multiple rank-1 lattices			Algorithm 4 using multiple rank-1 lattices		
	3e-1	1e-1	1e-2	3e-1	1e-1	1e-2	3e-1	1e-1	1e-2
$r=1$	875 / 0.000	948 / 0.042	987 / 0.747	876 / 0.000	949 / 0.037	993 / 0.726	859 / 0.000	954 / 0.041	987 / 0.709
$r=2$	975 / 0.395	984 / 0.913	998 / 0.999	979 / 0.409	996 / 0.919	998 / 0.998	982 / 0.386	984 / 0.915	1000 / 1.000
$r=3$	996 / 0.912	998 / 0.996	1000 / 1.000	996 / 0.931	998 / 0.997	1000 / 1.000	995 / 0.907	998 / 0.997	1000 / 1.000
$r=4$	998 / 0.995	1000 / 1.000	1000 / 1.000	998 / 0.988	1000 / 1.000	1000 / 1.000	997 / 0.992	1000 / 1.000	1000 / 1.000
$r=5$	998 / 0.997	1000 / 1.000	1000 / 1.000	1000 / 1.000	1000 / 1.000	1000 / 1.000	1000 / 1.000	1000 / 1.000	1000 / 1.000

Table 5.2: Minimal number of correctly detected frequencies (first number) and success rate for correctly detecting all frequencies (second number) for different threshold parameters δ and numbers of detection iterations r when applying Algorithm 3, 4, 5 on 5-dimensional random trigonometric polynomials p of sparsity $|\operatorname{supp} \hat{p}| := 1\,000$.

$\varphi_{\boldsymbol{k}} \in [0,1)$ chosen uniformly at random. We perturb the sampling values of the trigonometric polynomial p by additive complex white Gaussian noise $\eta_j \in \mathbb{C}$ with zero mean and standard deviation σ, i.e., we have measurements $\tilde{p}(\boldsymbol{x}_j) = p(\boldsymbol{x}_j) + \eta_j$. Then, we may approximately compute the signal-to-noise ratio (SNR) in our case by

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

and we choose $\sigma := \sqrt{|\operatorname{supp}\hat{p}|}/\sqrt{\mathrm{SNR}}$ for a targeted SNR value. For our numerical tests in MATLAB, we generate the noise by $\eta_j := \sigma/\sqrt{2} \, * \, (\texttt{randn + 1i*randn})$, $j = 0,\ldots, M - 1$. The SNR is often measured using the logarithmic decibel scale (dB), where $\mathrm{SNR}_{\mathrm{dB}} = 10\log_{10}\mathrm{SNR}$ and $\mathrm{SNR} = 10^{\mathrm{SNR}_{\mathrm{dB}}/10}$, i.e., a linear $\mathrm{SNR} = 10^8$ corresponds to a logarithmic $\mathrm{SNR}_{\mathrm{dB}} = 80\mathrm{dB}$ and $\mathrm{SNR} = 1$ corresponds to $\mathrm{SNR}_{\mathrm{dB}} = 0\mathrm{dB}$.

First, we consider the case of Algorithm 5 using single reconstructing rank-1 lattices in Example 5.4. Afterwards, we compare the results with Algorithm 3 using multiple rank-1 lattices in Example 5.5. In doing so, we observe that one can achieve comparable reconstruction success and errors using Algorithm 3 while requiring distinctly less samples compared to using Algorithm 5.

Example 5.4. (see also [26, Example 3.15], reconstruction from noisy sampling values using Algorithm 5 and reconstructing single rank-1 lattices). We set the dimension $d := 10$, the expansion $N := 32$, the sparsity $|\operatorname{supp}\hat{p}| := 1\,000$, and we use the search space $\Gamma := \hat{G}_{32}^{10}$. Moreover, we set the sparsity parameter $s := 1\,000$ and the threshold parameter $\delta := 10^{-12}$. Algorithm 5 was run setting the parameter r for the number of detection iterations to $r := 1, 2, 5$ and using the SNR values $\mathrm{SNR}_{\mathrm{dB}} := 80, 70, \ldots, 10, 0$ (which corresponds to $\mathrm{SNR} = 10^8, 10^7, \ldots, 10, 1$). For each of these test settings, we repeated the tests $1\,000$ times with new randomly chosen frequencies and Fourier coefficients. The numerical results are presented in

Table 5.3. The total number of samples for each of the 1 000 repetitions was computed and the maximum of these numbers for each test setting can be found in the column "#samples". In the column "min #freq. correct", the minimal number of correctly detected frequencies $|I^{(1,\ldots,10)} \cap \operatorname{supp} \hat{p}|$ for the 1 000 repetitions is shown, where $\operatorname{supp} \hat{p}$ denotes the set of true (input) frequencies of a trigonometric polynomial p and $I^{(1,\ldots,10)}$ the frequencies returned by the detection algorithm. The column "success rate (all freq. correct)" represents the relative number of the 1 000 repetitions where all frequencies were successfully detected, $I^{(1,\ldots,10)} = \operatorname{supp} \hat{p}$. Moreover, the relative ℓ_2-error $\|(\tilde{\hat{p}}_{\boldsymbol{k}})_{\boldsymbol{k} \in I} - (\hat{p}_{\boldsymbol{k}})_{\boldsymbol{k} \in I}\|_2 / \|(\hat{p}_{\boldsymbol{k}})_{\boldsymbol{k} \in I}\|_2$ of the computed coefficients $(\tilde{\hat{p}}_{\boldsymbol{k}})_{\boldsymbol{k} \in I^{(1,\ldots,10)}}$ was determined for each repetition, where $I := \operatorname{supp} \hat{p} \cup I^{(1,\ldots,10)}$ and $\tilde{\hat{p}}_{\boldsymbol{k}} := 0$ for $\boldsymbol{k} \in I \setminus I^{(1,\ldots,10)}$, and the column "rel. ℓ_2-error" contains the maximal value of the 1 000 repetitions. We obtain results similar to [26, Example 3.15]. In general, we observe that for decreasing SNR values, the minimal number of correctly detected frequencies and the success rate decrease. When using $r = 1$ detection iteration, there were always some (of the 1 000 test runs), where a few frequencies were incorrect. When we increased the number of detection iterations r, the SNR level at which all frequencies in all of the 1 000 test runs were correctly detected also decreased. For instance for $r = 5$ detection iterations, the success rate was at 100 percent including the case $\mathrm{SNR}_{\mathrm{dB}} = \mathrm{SNR} = 10$. However, we require about 4 times of the samples for $r = 5$ detection iterations compared to the tests with $r = 1$. □

$\mathrm{SNR}_{\mathrm{dB}}$	noise σ	#detect. iter. r	max. total #samples	min #freq. correct	success rate (all freq. correct)	max. rel. ℓ_2-error
80	3.2e-03	1	22 580 805	998	0.995	4.5e-02
70	1.0e-02	1	22 783 605	998	0.991	5.5e-02
60	3.2e-02	1	22 046 375	998	0.973	5.5e-02
50	1.0e-01	1	23 930 465	996	0.944	6.3e-02
40	3.2e-01	1	22 263 475	994	0.760	7.7e-02
30	1.0e+00	1	22 217 585	992	0.443	1.0e-01
80	3.2e-03	2	41 573 935	1 000	1.000	2.5e-06
70	1.0e-02	2	42 387 345	1 000	1.000	7.4e-06
60	3.2e-02	2	42 284 645	1 000	1.000	2.4e-05
50	1.0e-01	2	42 055 975	1 000	1.000	7.2e-05
40	3.2e-01	2	41 578 745	1 000	1.000	2.3e-04
30	1.0e+00	2	41 531 555	998	0.991	5.5e-02
50	1.0e-01	5	94 793 335	1 000	1.000	6.3e-05
40	3.2e-01	5	96 350 345	1 000	1.000	1.9e-04
30	1.0e+00	5	96 262 595	1 000	1.000	6.5e-04
20	3.2e+00	5	95 140 955	1 000	1.000	2.0e-03
10	1.0e+01	5	94 813 745	1 000	1.000	6.4e-03
0	3.2e+01	5	93 780 115	999	0.800	3.8e-02

Table 5.3: Results for random sparse trigonometric polynomials perturbed by additive white Gaussian noise using Algorithm 5 (reconstructing single rank-1 lattices).

Example 5.5. Next, we apply Algorithm 3 using the identical setting and parameters from Example 5.4. We observe similar results as before in Table 5.4. In general, Algorithm 3 requires about 2/3 of the number of samples compared to Algorithm 5 and has slightly better success rates. We suspect that this is due to the computation of the projected Fourier coefficients by Algorithm 2 using averaging.

Additionally, we apply Algorithm 4 and present the results in Table 5.5. When using $r = 1$ detection iteration, there were again always some test runs (of 1 000), where not all frequencies could be exactly detected. Moreover, we observe that for some of the test runs, Algorithm 4 with the used parameters was unable to correctly detect any frequency and the minimal number of correctly detected frequencies "min #freq. correct"=0. When we increase the number of detection iterations r to 2, the reconstruction success was distinctly higher but worse than the results of Algorithm 5 in Table 5.3 and of Algorithm 3 in Table 5.5. Similarly, for $r = 5$ detection iterations, we were able to reconstruct better at higher noise levels but still worse compared to Table 5.3 and 5.4.

Next, we changed the parameter s_{local} of Algorithm 4 to the value 1 200, which means that more frequency candidates are considered and kept in step 2e. Now the minimal number of correctly detected frequencies is distinctly better in many cases but still worse than using Algorithm 5 or 3. We suspect that this is also caused by the distinctly smaller number of used sampling values for Algorithm 4, which is about 1/9 of Algorithm 5 and 1/6 of Algorithm 3.

$\square$

$\mathrm{SNR_{dB}}$	noise σ	#detect. iter. r	max. total #samples	min #freq. correct	success rate (all freq. correct)	max. rel. ℓ_2-error
80	3.2e-03	1	13 032 485	998	0.998	4.6e-02
70	1.0e-02	1	13 536 429	996	0.990	6.5e-02
60	3.2e-02	1	13 310 277	998	0.988	4.7e-02
50	1.0e-01	1	12 911 065	994	0.949	7.8e-02
40	3.2e-01	1	13 301 373	996	0.857	6.7e-02
30	1.0e+00	1	13 405 817	992	0.621	9.2e-02
80	3.2e-03	2	25 290 563	1 000	1.000	4.4e-06
70	1.0e-02	2	24 883 447	1 000	1.000	1.4e-05
60	3.2e-02	2	24 589 823	1 000	1.000	4.4e-05
50	1.0e-01	2	25 252 543	1 000	1.000	1.4e-04
40	3.2e-01	2	24 937 437	1 000	1.000	4.4e-04
30	1.0e+00	2	25 433 681	998	0.997	4.8e-02
50	1.0e-01	5	63 052 299	1 000	1.000	1.4e-04
40	3.2e-01	5	63 618 665	1 000	1.000	4.3e-04
30	1.0e+00	5	63 294 069	1 000	1.000	1.4e-03
20	3.2e+00	5	65 500 751	1 000	1.000	4.4e-03
10	1.0e+01	5	63 537 849	1 000	1.000	1.5e-02
0	3.2e+01	5	67 046 015	998	0.992	6.2e-02

Table 5.4: Results for random sparse trigonometric polynomials perturbed by additive white Gaussian noise using Algorithm 3 (reconstructing multiple rank-1 lattices).

SNR$_{\text{dB}}$	noise σ	#detect. iter. r	max. total #samples	min #freq. correct	success rate (all freq. correct)	max. rel. ℓ_2-error
80	3.2e-03	1	1 617 311	1	0.999	1.0e+00
70	1.0e-02	1	1 616 183	0	0.990	1.0e+00
60	3.2e-02	1	1 616 843	0	0.967	1.0e+00
50	1.0e-01	1	1 614 591	0	0.906	1.0e+00
40	3.2e-01	1	1 617 431	0	0.561	1.0e+00
30	1.0e+00	1	1 614 189	0	0.000	1.0e+00
80	3.2e-03	2	3 063 489	1 000	1.000	9.3e-06
70	1.0e-02	2	3 064 963	1 000	1.000	3.0e-05
60	3.2e-02	2	3 069 693	1 000	1.000	9.4e-05
50	1.0e-01	2	3 130 709	841	0.999	4.2e-01
40	3.2e-01	2	3 257 421	570	0.934	6.8e-01
30	1.0e+00	2	3 379 167	505	0.042	7.3e-01
50	1.0e-01	5	7 582 333	1 000	1.000	2.9e-04
40	3.2e-01	5	7 570 209	1 000	1.000	9.1e-04
30	1.0e+00	5	7 611 301	997	0.100	5.5e-02
20	3.2e+00	5	7 855 307	971	0.000	1.7e-01
10	1.0e+01	5	9 310 751	834	0.000	4.1e-01
0	3.2e+01	5	13 998 685	0	0.000	1.0e+00

Table 5.5: Results for random sparse trigonometric polynomials perturbed by additive white Gaussian noise using Algorithm 4 with $s_{\text{local}} := 1\,000$.

SNR$_{\text{dB}}$	noise σ	#detect. iter. r	max. total #samples	min #freq. correct	success rate (all freq. correct)	max. rel. ℓ_2-error
80	3.2e-03	1	1 940 759	995	0.999	7.7e-02
70	1.0e-02	1	1 938 367	982	0.994	1.4e-01
60	3.2e-02	1	1 942 751	971	0.980	1.8e-01
50	1.0e-01	1	1 937 867	969	0.955	1.8e-01
40	3.2e-01	1	1 941 323	959	0.835	2.1e-01
30	1.0e+00	1	1 939 205	940	0.065	2.5e-01
80	3.2e-03	2	3 957 003	1 000	1.000	8.1e-06
70	1.0e-02	2	3 952 989	1 000	1.000	2.6e-05
60	3.2e-02	2	3 972 265	1 000	1.000	8.2e-05
50	1.0e-01	2	3 954 775	1 000	1.000	2.6e-04
40	3.2e-01	2	3 957 161	998	0.984	4.5e-02
30	1.0e+00	2	3 954 715	985	0.089	1.3e-01
50	1.0e-01	5	10 785 425	1 000	1.000	2.5e-04
40	3.2e-01	5	10 768 847	998	0.987	4.5e-02
30	1.0e+00	5	10 777 273	994	0.105	8.4e-02
20	3.2e+00	5	10 848 099	971	0.000	1.7e-01
10	1.0e+01	5	10 804 485	843	0.000	4.0e-01
0	3.2e+01	5	17 181 009	0	0.000	1.0e+00

Table 5.6: Results for random sparse trigonometric polynomials perturbed by additive white Gaussian noise using Algorithm 4 with $s_{\text{local}} := 1\,200$.

5.3 Approximation of tensor-product function by trigonometric polynomials

Next, we consider the multivariate periodic test function $f \colon \mathbb{T}^{10} \to \mathbb{R}$,

$$f(\boldsymbol{x}) := \prod_{t \in \{1,3,8\}} N_2(x_t) + \prod_{t \in \{2,5,6,10\}} N_4(x_t) + \prod_{t \in \{4,7,9\}} N_6(x_t), \qquad (5.1)$$

from [26, Section 3.3] with infinitely many non-zero Fourier coefficients, where $N_m \colon \mathbb{T} \to \mathbb{R}$ is the B-Spline of order $m \in \mathbb{N}$,

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

with a constant $C_m > 0$ such that $\|N_m|L_2(\mathbb{T})\| = 1$. We approximate the function f by multivariate trigonometric polynomials p. For this, we determine a frequency index set $I = I^{(1,\ldots,10)} \subset \Gamma := \hat{G}_N^{10}$ and we compute approximated Fourier coefficients $\tilde{\hat{p}}_{\boldsymbol{k}}$, $\boldsymbol{k} \in I$, using Algorithm 5, 3 and 4. The obtained frequency index sets I should "consist of" the union of three lower dimensional manifolds, 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

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

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

Since the number of non-zero Fourier coefficients is not finite and we use a full grid $\hat{G}_N^{10}$ for the search space, we have to limit the number of detected frequencies $\boldsymbol{k} \in I$ and Fourier coefficients $\tilde{\hat{p}}_{\boldsymbol{k}}$. For this, we use two different strategies from [26, Section 3.3]. In Section 5.3.1, we set the sparsity parameter s appropriately. Moreover, in Section 5.3.2, we use the threshold parameter δ in order to truncate smaller Fourier coefficients.

5.3.1 s-sparse

First, we use the sparsity input parameter $s \in \mathbb{N}$ to limit the number of detected Fourier coefficients, see also [26, Section 3.3.1]. In doing so, we set the threshold parameter δ to a very small value, which is $\delta := 10^{-12}$ for the following examples. In each dimension increment step t, the detected Fourier coefficients and frequencies are truncated and only those frequencies are kept which belong to the $s \ll |\Gamma| < \infty$ largest Fourier coefficients. Due to possible aliasing effects, we use a larger value s_{local} for the intermediate dimension increment steps $t \in \{2, \ldots, d-1\}$ than for the final truncation of the index set $I^{(1,\ldots,d)}$ in dimension increment step $t = d$ in Algorithm 5, 3 and 4.

Example 5.6. (*s*-sparse approximate function reconstruction). We set the expansion $N := 16, 32, 64$ and we use the full grids $\Gamma := \hat{G}_N^{10}$ as search space. Moreover, we set the number of detection iterations $r := 5$. The used sparsity input parameters s and s_{local} as well as the results of Algorithm 5 using single rank-1 lattices, of Algorithm 3 using multiple rank-1 lattices, and of Algorithm 4 using multiple rank-1 lattices are presented in Table 5.7. 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 5.1. We observe that for increasing sparsity parameter, the number of samples increases while the relative $L_2(\mathbb{T}^{10})$ approximation error decreases.

When comparing Algorithm 3 and 5, we observe that both methods yield similar errors. Furthermore, Algorithm 3 requires slightly more samples for small sparsity parameters s and about half the number of samples for large ones compared with Algorithm 5.

We observe that the number of samples could be drastically reduced when using Algorithm 4, whereas the relative $L_2(\mathbb{T}^{10})$ approximation errors are comparable for all three algorithms in most cases. For instance in the case $N = 64$ and $s = 5\,000$, the maximal total number of samples for Algorithm 5 (computed over 10 test runs) was about 2.3 billion and for Algorithm 4 only about 159 million samples, which is almost $1/15$, achieving comparable relative errors. Only in 1 test run (out of 10) for expansion $N = 64$ and sparsity $s = 1\,000$, Algorithm 4 failed to detect some of the frequencies resulting in a relative error of about $4.5 \cdot 10^{-1}$ whereas it succeeded for the remaining 9 test runs yielding a relative error of about $1.2 \cdot 10^{-2}$, which corresponds to the results of Algorithm 5.

As observed in [26, Section 3.3.1], it is not sufficient to only increase the used sparsity s but the expansion parameter N also needs to be increased. Using a large expansion N and a small target sparsity s results in the usage of distinctly more samples, e.g., about 50 million samples for $N = 16$ and sparsity $s = 1\,000$ compared to about 243 million samples for $N = 64$ for Algorithm 5 as well as about 8 million samples for $N = 16$ and sparsity $s = 1\,000$ compared to about 34 million samples for $N = 64$ for Algorithm 4. This corresponds to the almost linear factor N in the sample complexity, cf. Section 3.2. □

		single rank-1 lattices and Algorithm 5		multiple rank-1 lattices and Algorithm 3		multiple rank-1 lattices and Algorithm 4	
N	s	max. total #samples	max. rel. L_2-error	max. total #samples	max. rel. L_2-error	max. total #samples	max. rel. L_2-error
16	1 000	50 405 091	1.2e-02	62 671 623	1.2e-02	8 094 293	1.3e-02
16	2 000	128 362 707	4.3e-03	127 612 901	3.9e-03	15 449 367	5.0e-03
16	3 000	222 662 847	3.4e-03	176 257 025	3.0e-03	22 285 615	4.1e-03
32	1 000	84 241 365	1.2e-02	130 597 513	1.2e-02	16 623 363	1.3e-02
32	2 000	265 019 105	3.4e-03	258 312 555	3.4e-03	32 226 797	8.8e-03
32	3 000	565 847 035	1.7e-03	376 108 443	1.6e-03	46 948 295	2.2e-03
32	4 000	767 068 055	1.4e-03	525 215 599	1.2e-03	59 623 261	1.6e-03
64	1 000	242 940 304	1.2e-02	251 626 807	1.2e-02	34 236 011	4.5e-01
64	2 000	563 767 023	3.4e-03	547 034 957	3.4e-03	66 914 731	3.5e-03
64	3 000	946 215 129	1.6e-03	778 060 799	1.6e-03	98 926 095	1.8e-03
64	4 000	1 266 499 683	9.8e-04	1 084 767 689	9.7e-04	130 967 911	1.1e-03
64	5 000	2 285 094 003	7.1e-04	1 318 206 819	6.9e-04	158 693 803	9.6e-04
64	10 000	5 577 419 619	4.3e-04	2 639 274 025	3.8e-04	298 886 055	5.1e-04

Table 5.7: Results for function $f \colon \mathbb{T}^{10} \to \mathbb{R}$ from (5.1) when limiting the number of detected frequencies, $s_{\text{local}} := 2s$.

5.3.2 threshold-based

Additionally, we consider a variant from [26, Section 3.3.2]. Now, we use the absolute threshold parameter $\delta \in (0, 1)$ for the truncation. For the search space Γ, we use the full grid $\hat{G}_N^{10}$

for various expansions $N \in \mathbb{N}$. Moreover, we set the sparsity input parameter $s := |\Gamma|$ and we use $r := 10$ detection iterations, which is twice as high than in Section 5.3.1. However, this was required for successfully determining the relevant frequency index sets I and achieving low approximation errors.

Example 5.7. (Threshold-based approximate function reconstruction). We apply Algorithm 5 and 4 on the 10-dimensional test function (5.1). The parameters and results are shown in Table 5.8. For each parameter combination, we perform the tests 10 times. For the truncation of the one-dimensional index sets $I^{(t)}$ of frequency candidates for component t, $t \in \{1, \ldots, 10\}$, the threshold parameter $\delta :=$ "threshold"$/10$ is used as well as $\delta :=$ "threshold" for all other truncations. The column "max. $|I|$" denotes the maximal number $|I^{(1,\ldots,d)}|$ of Fourier coefficients, which were returned by Algorithm 5 and 4, respectively. The remaining columns have the same meaning as in Example 5.6. We observe that the total numbers of samples are dramatically smaller compared to the results from Section 5.3.1 while the relative $L_2(\mathbb{T}^{10})$ approximation errors are comparable in most cases for similar numbers $|I|$ of Fourier coefficients $\tilde{\hat{p}}_{\boldsymbol{k}}$ used for the approximation $\tilde{S}_I f$ of f. Moreover, the number of samples when using multiple rank-1 lattices and Algorithm 4 is distinctly smaller compared to using single rank-1 lattices and Algorithm 5. However, the number $|I^{(1,\ldots,d)}|$ of Fourier coefficients is slightly larger when using Algorithm 4 for "threshold" values above 10^{-5} and distinctly larger for 10^{-6}.

Additionally, we applied Algorithm 3 and we observe a similar behavior as in Example 5.6. Algorithm 3 and 5 yield similar errors. Moreover, Algorithm 3 requires slightly more samples for large thresholds parameters δ and about half the number of samples for small δ compared to Algorithm 5. $\qquad\square$

		single rank-1 lattices and Algorithm 5			multiple rank-1 lattices and Algorithm 4		
N	threshold	max. $\lvert I\rvert$	max. total #samples	max. rel. L_2-error	max. $\lvert I\rvert$	max. total #samples	max. rel. L_2-error
64	1.0e-02	531	388 370	4.2e-01	573	299 551	1.9e-01
64	1.0e-03	1 157	4 089 813	1.1e-02	1 327	2 172 161	1.1e-02
64	1.0e-04	3 075	22 232 568	1.6e-03	3 847	8 924 419	2.1e-03
64	1.0e-05	8 041	163 970 316	4.6e-04	12 499	42 668 463	5.1e-04
64	1.0e-06	22 019	1 087 478 506	4.1e-04	56 375	249 240 643	4.2e-04
128	1.0e-02	531	413 506	1.3e-01	567	328 595	6.1e-02
128	1.0e-03	1 163	3 196 264	1.0e-02	1 357	1 865 193	3.9e-02
128	1.0e-04	3 099	32 334 421	1.6e-03	3 733	13 946 741	1.6e-03
128	1.0e-05	8 491	250 367 814	3.0e-04	11 039	52 172 733	2.6e-03
128	1.0e-06	23 609	1 679 136 524	1.5e-04	50 571	316 908 069	1.6e-04
256	1.0e-02	531	456 113	2.0e-01	571	352 607	6.9e-02
256	1.0e-03	1 155	3 185 671	1.1e-02	1 355	1 890 573	1.1e-02
256	1.0e-04	3 085	51 682 418	1.6e-03	3 653	18 860 827	1.6e-03
256	1.0e-05	8 553	422 552 990	2.9e-04	10 863	77 359 193	2.9e-04
256	1.0e-06	25 563	3 366 629 304	6.3e-05	40 331	373 569 439	1.0e-04

Table 5.8: Results for approximation of function $f \colon \mathbb{T}^{10} \to \mathbb{R}$ from (5.1) when truncating the detected frequencies and Fourier coefficients by threshold.

6 Conclusion

In this work, we presented fast, efficient, reliable algorithms for high-dimensional sparse FFTs based on dimension-incremental projections. We applied the new algorithms to the reconstruction and approximation of high-dimensional functions, assuming that the frequencies of important Fourier coefficients are unknown.

One main ingredient was the utilization of multiple rank-1 lattices, which are newly developed sampling schemes allowing for highly efficient construction methods and corresponding FFT algorithms for known support $\operatorname{supp} \hat{p}$ in frequency domain.

Another important aspect is the modification of algorithm parameters that allow for theoretical estimates. In particular, a detailed analysis of specific realizations of the dimension-incremental approach was performed for the first time with respect to success probability. Moreover, we estimated the sample and arithmetic complexities for the case of successful detection as well as for the worst case.

Various numerical tests complement the theoretical considerations and indicate that one may obtain smaller bounds on the complexities, especially with respect to the sparsity of the signal under consideration. In Section 4.2 we discussed one improvement idea and point out the emerging open questions.

References

[1] R. Bergmann. The fast Fourier transform and fast wavelet transform for patterns on the torus. *Appl. Comput. Harmon. Anal.*, 35:39–51, 2013.

[2] G. Byrenheid, L. Kämmerer, T. Ullrich, and T. Volkmer. Tight error bounds for rank-1 lattice sampling in spaces of hybrid mixed smoothness. *Numer. Math.*, 136:993–1034, 2017.

[3] E. J. Candès. Compressive sampling. In *International Congress of Mathematicians. Vol. III*, pages 1433–1452. Eur. Math. Soc., Zürich, 2006.

[4] J. Dick, F. Y. Kuo, and I. H. Sloan. High-dimensional integration: The quasi-Monte Carlo way. *Acta Numer.*, 22:133–288, 2013.

[5] D. L. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52:1289–1306, 2006.

[6] S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing.* Applied and Numerical Harmonic Analysis. Birkhäuser/Springer, New York, 2013.

[7] M. Frigo and S. G. Johnson. FFTW, C subroutine library. `http://www.fftw.org`, 2009.

[8] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Nearly optimal sparse Fourier transform. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, pages 563–578. ACM, 2012.

[9] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Simple and practical algorithm for sparse Fourier transform. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1183–1194. SIAM, 2012.

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

[11] P. Indyk, M. Kapralov, and E. Price. (Nearly) sample-optimal sparse Fourier transform. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, pages 563–578. ACM, 2014.

[12] M. A. Iwen. Improved approximation guarantees for sublinear-time Fourier algorithms. *Appl. Comput. Harmon. Anal.*, 34:57–82, 2013.

[13] L. Kämmerer. *High Dimensional Fast Fourier Transform Based on Rank-1 Lattice Sampling*. Dissertation. Universitätsverlag Chemnitz, 2014.

[14] L. Kämmerer. Reconstructing multivariate trigonometric polynomials from samples along rank-1 lattices. In G. E. Fasshauer and L. L. Schumaker, editors, *Approximation Theory XIV: San Antonio 2013*, pages 255–271. Springer International Publishing, 2014.

[15] L. Kämmerer. Multiple rank-1 lattices as sampling schemes for multivariate trigonometric polynomials. *J. Fourier Anal. Appl.*, 2016.

[16] L. Kämmerer. Constructing spatial discretizations for sparse multivariate trigonometric polynomials that allow for a fast discrete Fourier transform. *ArXiv e-prints 1703.07230*, Nov. 2017.

[17] L. Kämmerer, S. Kunis, I. Melzer, D. Potts, and T. Volkmer. Computational Methods for the Fourier Analysis of Sparse High-Dimensional Functions. In S. Dahlke, W. Dahmen, M. Griebel, W. Hackbusch, K. Ritter, R. Schneider, C. Schwab, and H. Yserentant, editors, *Extraction of Quantifiable Information from Complex Systems*, 2014.

[18] P. Kritzer, F. Y. Kuo, D. Nuyens, and M. Ullrich. Lattice rules with random n achieve nearly the optimal $\mathcal{O}(n^{-\alpha-1/2})$ error independently of the dimension. *ArXiv e-prints 1706.04502*, June 2017.

[19] S. Kunis and H. Rauhut. Random sampling of sparse trigonometric polynomials II, Orthogonal matching pursuit versus basis pursuit. *Found. Comput. Math.*, 8:737–763, 2008.

[20] F. Y. Kuo, I. H. Sloan, and H. Woźniakowski. Lattice rules for multivariate approximation in the worst case setting. In H. Niederreiter and D. Talay, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2004*, pages 289–330. Springer Berlin Heidelberg, Berlin, 2006.

[21] F. Y. Kuo, I. H. Sloan, and H. Woźniakowski. Lattice rule algorithms for multivariate approximation in the average case setting. *J. Complexity*, 24:283–323, 2008.

[22] F. Y. Kuo, G. W. Wasilkowski, and H. Woźniakowski. Lattice algorithms for multivariate L_∞ approximation in the worst-case setting. *Constr. Approx.*, 30:475–493, 2009.

[23] M. Loève. *Probability Theory I*. Graduate Texts in Mathematics. Springer-Verlag New York, 4th edition, 1977.

[24] H. Munthe-Kaas and T. Sørevik. Multidimensional pseudo-spectral methods on lattice grids. *Appl. Numer. Math.*, 62:155–165, 2012.

[25] D. Potts and M. Tasche. Parameter estimation for multivariate exponential sums. *Electron. Trans. Numer. Anal.*, 40:204–224, 2013.

[26] D. Potts and T. Volkmer. Sparse high-dimensional FFT based on rank-1 lattice sampling. *Appl. Comput. Harmon. Anal.*, 41:713–748, 2016.

[27] I. H. Sloan and S. Joe. *Lattice methods for multiple integration*. Oxford Science Publications. The Clarendon Press Oxford University Press, New York, 1994.

[28] V. N. Temlyakov. Reconstruction of periodic functions of several variables from the values at the nodes of number-theoretic nets. *Anal. Math.*, 12:287–305, 1986. In Russian.