RAPIDLY COMPUTING SPARSE LEGENDRE EXPANSIONS VIA SPARSE FOURIER TRANSFORMS

XIANFENG HU, MARK IWEN, AND HYEJIN KIM

Abstract. In this paper we propose a general strategy for rapidly computing sparse Legendre expansions. The resulting methods yield a new class of fast algorithms capable of approximating a given function $f : [-1, 1] \to \mathbb{R}$ with a near-optimal linear combination of s Legendre polynomials of degree $\leq N$ in just $(s \log N)^{\mathcal{O}(1)}$-time. When $s \ll N$ these algorithms exhibit sublinear runtime complexities in N, as opposed to traditional $\Omega(N \log N)$-time methods for computing all of the first N Legendre coefficients of f. Theoretical as well as numerical results demonstrate the effectiveness of the proposed methods.

1. Introduction

In this paper we consider *Legendre-compressible functions* which can be well approximated by a linear combination of a small number of unknown, and potentially high-degree, Legendre polynomials. Given such a function our objective is to quickly learn the best basis of Legendre polynomials with which to approximate it, and then to compute their coefficients. Let $f : [-1, 1] \to \mathbb{R}$ be a degree N polynomial, and $L_n(x)$ denote the Legendre polynomial of degree n. We aim to rapidly and accurately compute f's Legendre coefficients, $\tilde{f}(n) \in \mathbb{R}$ for $n \in \{0, \ldots, N\}$ with

$$f(x) = \sum_{n=0}^{N} \tilde{f}(n) L_n(x), \tag{1}$$

whenever $\tilde{f}(n) \approx 0$ for all but $s \ll N$ *initially unknown* values of n. We will call any numerical method with this objective a *sparse Legendre expansion algorithm.*

Note that solving this problem is straightforward if one is willing to sample f at $N + 1$ points in $[-1, 1]$, and then compute all $N + 1$ of its Legendre coefficients. However, any such approach will necessarily require $\Omega(N)$-operations, which can become overwhelming when the maximal degree, N, of f is large. Our objective here is to select the best basis of $s \ll N$ Legendre polynomials of degree $\leq N$ for f, and then estimate their coefficients, in $(s \log N)^{\mathcal{O}(1)}$-time. When s is significantly smaller than N, these methods will be faster than any traditional approach which computes all N Legendre coefficients of f.

Fast sparse Legendre expansion algorithms of this kind are a natural first step toward the development of computationally tractable algorithms for approximating functions of many variables with respect to tensorized Legendre polynomial bases. In such multivariate problems the maximal degree, N, grows exponentially in the number of variables, rapidly rendering even $\mathcal{O}(N)$-time methods impractical. Given this, extending $(s \log N)^{\mathcal{O}(1)}$-time sparse expansion methods for functions of one variable to the multivariate setting, once they are properly understood, would be of value in many big data era computational applications including, e.g., uncertainty quantification [22] and the computation of polynomial chaos expansions [6, 7]. In addition, it is worth pointing

Xianfeng (Janice) Hu: Institute for Mathematics and its Applications, University of Minnesota (xhu@umn.edu).

M.A. Iwen: Department of Mathematics and Department of ECE, Michigan State University (markiwen@math.msu.edu). M.A. Iwen was supported in part by NSF DMS-1416752.

Hyejin Kim: Department of Mathematics and Statistics, University of Michigan – Dearborn (khyejin@umich.edu).

out that the fast sparse expansion techniques developed herein for sparse Legendre and Chebyshev polynomials are also valuable as means for accelerating standard compressive sensing algorithms for solving sparse approximation problems in these bases [8, 30] when N is large. In particular, one could always begin solving (1) by first utilizing a fast $(s \log N)^{\mathcal{O}(1)}$-time sparse expansion method of the kind developed herein in order to rapidly approximate a best basis of $s \ll N$ Legendre polynomials for f. This basis could then be used to inform a good initial approximation of f in order to help accelerate a given $N^{\mathcal{O}(1)}$-time compressive sensing method such as Basis Pursuit.

The majority of previously proposed sparse Legendre expansion methods are based on Prony-like approaches. Examples include results by Peter et. al. [26] who develop a method based on a more general approach from [25] which needs only $\mathcal{O}(s)$ samples from (various derivatives of) f in order to recover its s-sparse Legendre expansion. More recently, Potts and Tasche [29] used approximation techniques to adapt previous Prony-like methods for the recovery of Chebyshev-sparse functions [28] to the Legendre-sparse setting. However, no theoretical results are proven in [29] that demonstrate the methods therein can extend to functions with compressible (as opposed to exactly s-sparse) Legendre expansions. In contrast, herein we provide a theoretical support recovery guarantee which proves that our techniques can indeed locate the principle support of a relatively large class of Legendre-compressible functions (see, e.g., Theorem 5). When combined with coefficient estimation methods based on techniques from compressive sensing (see, e.g., Lemma 4) these support recovery guarantees allow one to prove a variety of general sublinear-time recovery guarantees for functions with compressible Legendre expansions.

Other sparse Legendre expansion methods include those based on compressive sensing approaches [8]. In particular, Rauhut and Ward [30] demonstrate that $\mathcal{O}(s \cdot \log^4 N)$ samples from f suffice in order to accurately and stably approximate f with a near-optimal sparse Legendre expansion. The associate reconstruction algorithms are $\Omega(N)$-time, however. As opposed to these previous sparse Legendre expansion methods, we propose a new approach motivated by a recently proposed FFT-based method for computing all $N+1$ Legendre coefficients of a given function f in $\mathcal{O}(N \log N)$-time [16]. This method works by (i) implicitly mapping the Legendre coefficients of f to the Fourier coefficients of a related function, f_r, which remains easily to sample, followed by (ii) computing the Fourier coefficients of f_r with an FFT, and then (iii) using the computed Fourier coefficients of f_r in order to recover the Legendre coefficients of the original function f (i.e., by inverting the map from (i)). The sparse Legendre expansion methods proposed herein are based on this same type of approach. More specifically, we demonstrate that one may successfully replace the use of a standard FFT in the three step procedure above with the use of any Sparse Fourier Transform algorithm one desires (see, e.g., [12, 15, 21, 32]) in order to rapidly approximate Legendre-compressible functions.

1.1. **Sparse Fourier Transforms.** Sparse Fourier Transforms (SFTs) are algorithms for quickly computing near-optimal sparse approximations to the Fourier series of a given periodic function $f : [-\pi, \pi]^D \to \mathbb{C}$. Suppose that f is a trigonometric polynomial of degree N in every variable so that the Fourier series of f is effectively $\hat{f} \in \mathbb{C}^{N^D}$. An optimal s-term trigonometric approximation to f is given by

$$f_s^{\text{opt}}(\mathbf{x}) := \sum_{j=1}^{s} \hat{f}\left(\boldsymbol{\omega}_j\right) \mathbb{e}^{\mathbb{i}\boldsymbol{\omega}_j \cdot \mathbf{x}}$$

where $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_{N^D} \in (-N/2, N/2]^D \cap \mathbb{Z}^D$ are ordered by the magnitudes of their Fourier coefficients, $\hat{f}\left(\boldsymbol{\omega}_j\right) \in \mathbb{C}$, so that

$$\left|\hat{f}(\boldsymbol{\omega}_1)\right| \geq \left|\hat{f}(\boldsymbol{\omega}_2)\right| \geq \cdots \geq \left|\hat{f}(\boldsymbol{\omega}_{N^D})\right|. \tag{2}$$

The optimal s-term approximation error is then $\|f - f_s^{\text{opt}}\|_2 = \|\hat{f} - \hat{f}_s^{\text{opt}}\|_2$. In this setting, any discrete Fourier method will take function evaluations of f as input, and then output an approximate

f_s^{opt} for some value of $s \in \{0, \ldots, N^D\}$. A standard FFT always uses $s = N^D$, and so recovers trigonometric polynomials exactly. Sparse FFTs allow s to be chosen independently of N^D, and exactly recover all trigonometric polynomials consisting of at most s nonzero terms.

The first sparse Fourier methods were essentially approximate Hadamard transforms that were developed by researchers in the machine learning community for quickly learning boolean functions of many variables (see, e.g., [20, 3, 23] and [14, 13]). These techniques were later adapted to produce randomized SFTs for approximating trigonometric polynomials as rapidly as possible [24, 10, 1, 12]. These subsequent SFTs all take random samples of a given periodic function f as input, and then output a trigonometric polynomial, y, of degree N which satisfies $\|f - y\|_2 \approx \|f - f_s^{\text{opt}}\|_2$ with high probability. The fastest of these SFTs [12] uses only $s \cdot \log^{\mathcal{O}(1)} N$ operations. As a result, it is faster than the FFT for accurately approximating periodic functions which are dominated by $s \ll N$ of their largest magnitude Fourier coefficients [19].

More recently, entirely deterministic SFTs [17, 18] have been developed that are guaranteed to *always* return a near-optimal sparse trigonometric polynomial, $y_s : [-\pi, \pi]^D \to \mathbb{C}$, with $\|f - y_s\|_2 \approx \|f - f_s^{\text{opt}}\|_2$. More specifically, the following theorem was proven in [18].

Theorem 1. *Suppose $f : [-\pi, \pi]^D \to \mathbb{C}$ has $\hat{f}(\omega_1, \ldots, \omega_D) = 0$ if $(\omega_1, \ldots, \omega_D) \notin \left([-\frac{N}{2}, \frac{N}{2}] \cap \mathbb{Z}\right)^D$. Let $s, \epsilon^{-1} \in \mathbb{N} \setminus \{1\}$ with $(s/\epsilon)^2 \geq 4$. Then, there exists a simple deterministic algorithm that is guaranteed to output a trigonometric polynomial, $y_s : [-\pi, \pi]^D \to \mathbb{C}$, satisfying*

$$\|f - y_s\|_2 \leq \left\| \hat{f} - \hat{f}_s{}^{\text{opt}} \right\|_2 + \frac{22\epsilon \cdot \left\| \hat{f} - \hat{f}_{(s/\epsilon)}^{\text{opt}} \right\|_1}{\sqrt{s}}. \tag{3}$$

The algorithm's operation count is

$$\mathcal{O}\left(\frac{s^2 \cdot D^4 \cdot \log^4(ND)}{\log\left(\frac{s}{\epsilon}\right) \cdot \epsilon^2} \right). \tag{4}$$

If succeeding with probability $(1-\delta) \in [2/3, 1)$ is sufficient, and $(s/\epsilon) \geq 2$, a Monte Carlo variant of the deterministic algorithm may be used. This Monte Carlo variant will output a trigonometric polynomial, $y_s : [-\pi, \pi]^D \to \mathbb{C}$, that satisfies Equation 3 with probability at least $1 - \delta$. Its operation count will be

$$\mathcal{O}\left(\frac{s \cdot D^4}{\epsilon} \cdot \log^3(ND) \cdot \log\left(\frac{ND}{\delta} \right) \right). \tag{5}$$

The Fourier algorithms referred to by Theorem 1 are able to accurately approximate the discrete Fourier transform of a given function much more quickly than standard Fast Fourier Transform (FFT) methods [5] whenever the sorted magnitudes of the Fourier coefficients (2) go to zero quickly enough [32]. More specifically, the developed Fourier approximation algorithms have operation counts that scale *polynomially* in D and $\log N$, as opposed to standard FFT methods whose operation counts scale *exponentially* in D and $\log N$. We direct the reader to [11] for a recent survey of SFT techniques, as well as for an easy introduction to their design and implementation.

1.2. **A Simple Example: Recovering Sparse Chebyshev Expansions via SFTs.** In this section we briefly consider the use of SFTs for recovering Chebyshev-sparse functions of the form

$$g(x) = \sum_{n \in \mathcal{S}} a_n T_n(x), \ \mathcal{S} \subset \{0, \ldots, N\}, \ |\mathcal{S}| = s \ll N,$$

where $T_n(x)$ denotes the degree n Chebyshev polynomial of the first kind. In this setting it is beneficial to consider the standard transformation $h = g(\cos x)$. It is well known that this transformation implicitly maps the n^{th} Chebyshev coefficient of g to the n^{th} Fourier cosine series coefficient

of h (see, e.g., [5]). In particular, we have that

$$h(x) = \sum_{n\in\mathcal{S}} a_n T_n(\cos x) = \sum_{n\in\mathcal{S}} a_n \cos(nx) = \sum_{n\in\mathcal{S}} \frac{a_n}{2}\left(\mathbb{e}^{\mathbb{i}nx} + \mathbb{e}^{-\mathbb{i}nx}\right).$$

It is now straightforward to see that creating h by resampling g according to the cosine function implicitly produces a sparsity-preserving linear map from the Chebyshev coefficients of g, a_n, to the non-negative Fourier coefficients of h, $\hat{h}(\omega)$ as per (8) for $\omega = 0, 1, \ldots, N$, that is given by

$$\begin{pmatrix} \hat{h}(0) \\ \vdots \\ \hat{h}(N) \end{pmatrix} = \begin{pmatrix} 1 & 0 & \ldots & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \ldots & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ 0 & \ldots & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \ldots & 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_N \end{pmatrix}. \tag{6}$$

Note that the map above (6) has several important properties: It is exactly sparsity preserving, well conditioned, and trivially invertible. As a consequence, one can easily compute the sparse Chebyshev expansion of g using SFTs. One simply applies the SFT of their choice to h and then reconstructs the Chebyshev coefficients of g from the result using their knowledge of (6). One goal of the research initiated here is to find a good sparsity-preserving map similar to (6) for use with Legendre polynomials, if possible. In this paper we make a first attempt toward this goal by analyzing the maps proposed by Iserles in [16].

2. Notation and Background

We will denote the Legendre polynomial of degree n by L_n. The n^{th} Legendre coefficient of $f : [-1, 1] \to \mathbb{R}$ is then

$$\tilde{f}(n) := \left(n + \frac{1}{2}\right) \int_{-1}^{1} f(x) L_n(x)\, dx \tag{7}$$

for each $n \in \mathbb{N}$. The Fourier series coefficients of a function $f : [-\pi, \pi] \to \mathbb{C}$ will be denoted by

$$\hat{f}(\omega) := \frac{1}{2\pi} \int_{-\pi}^{\pi} f(x) \mathbb{e}^{-\mathbb{i}\omega x}\, dx \tag{8}$$

for all $\omega \in \mathbb{Z}$. The sequence of Legendre or Fourier coefficients of an appropriate function f will be called $\tilde{f}$ or $\hat{f}$, respectively.

For any matrix $X \in \mathbb{C}^{N\times n}$ we will denote the j^{th} column of X by $\mathbf{X}_j \in \mathbb{C}^N$. The adjoint of a matrix, $X \in \mathbb{C}^{N\times n}$, will be denoted by $X^* \in \mathbb{C}^{n\times N}$, and the singular values of any matrix $X \in \mathbb{C}^{N\times n}$ will always be ordered as $\sigma_1(X) \geq \sigma_2(X) \geq \cdots \geq \sigma_{\min(N,n)}(X) \geq 0$. Also, the condition number of the matrix X will denoted by $\kappa(X) := \sigma_1(X)/\sigma_{\min(N,n)}(X)$. We will use the notation $[N] := \{0, \ldots, N\} \subset \mathbb{N}$ for any $N \in \mathbb{N}$. For any matrix $X \in \mathbb{C}^{N\times(n+1)}$ and set $\mathcal{S} \subset [n]$ the matrix $X_{\mathcal{S}} \in \mathbb{C}^{N\times|\mathcal{S}|}$ will be the submatrix of X formed by selecting the columns of X indexed by $\mathcal{S}$. Similarly, for any vector $\mathbf{x} \in \mathbb{C}^N$ and set $\mathcal{S} \subset [n]$ the vector $\mathbf{x}_S \in \mathbb{C}^N$ will have entries

$$(x_S)_j = \begin{cases} 0, & \text{if } j \notin S \\ x_j, & \text{if } j \in S \end{cases}.$$

Finally, given any $\mathbf{x} \in \mathbb{C}^N$, the vector $\mathbf{x}_s{}^{\text{opt}} \in \mathbb{C}^N$ will always denote an optimal s-sparse approximation to $\mathbf{x}$. That is, $\mathbf{x}_s{}^{\text{opt}}$ will always (i) have at most $s \in [N]$ nonzero entries, and (ii) satisfy

$$\|\mathbf{x} - \mathbf{x}_s{}^{\text{opt}}\|_p = \inf_{\mathbf{z}\in\mathbb{C}^d, \|\mathbf{z}\|_0 \leq s} \|\mathbf{x} - \mathbf{z}\|_p \tag{9}$$

for all $p \geq 1$.

2.1. **Bounded Orthonormal Systems.** Let $\mathcal{D} \subset \mathbb{R}^n$ be endowed with a probability measure μ. Further, let $\Psi = \{\psi_0, \ldots, \psi_N\}$ be an orthonormal set of real-valued functions on $\mathcal{D}$ so that

$$\int_{\mathcal{D}} \psi_i (\mathbf{x}) \psi_j (\mathbf{x}) \, d\mu (\mathbf{x}) \; = \; \delta_{i,j}.$$

We will refer to any such Ψ as an *orthonormal system*. More specifically, we will utilize a particular type of orthonormal system.

Definition 1. *We call $\Psi = \{\psi_0, \ldots, \psi_N\}$ a bounded orthonormal system with constant $K \in \mathbb{R}^+$ if*

$$\| \psi_k \|_\infty := \sup_{\mathbf{x} \, \in \, \mathcal{D}} | \psi (\mathbf{x}) | \; \leq \; K \text{ for all } k \in [N].$$

For any orthonormal system, Ψ, on $\mathcal{D} \subset \mathbb{R}^n$ with probability measure μ, we may create an associated *random sampling matrix*, $R \in \mathbb{R}^{(m+1) \times (N+1)}$, as follows: First, select $m+1$ points $\mathbf{x}_0, \ldots, \mathbf{x}_m \in \mathcal{D}$ independently at random according to μ.[1] Then, form the matrix R by setting $R_{i,j} := \psi_j (\mathbf{x}_i)$ for each $i \in [m]$ and $j \in [N]$. The following theorem concerning random sampling matrices created from bounded orthonormal systems in this fashion is proven in [8].[2]

Theorem 2. *Let $R \in \mathbb{R}^{(m+1) \times (N+1)}$ be a random sampling matrix created from a bounded orthonormal system with constant $K \geq 1$. Let $\epsilon \in (0,1)$, $s \in [N] \setminus \{0\}$, and set $\widetilde{R} = \frac{1}{\sqrt{m+1}} R$. If $m \geq C K^2 \epsilon^{-2} s \ln^4(N)$, then with probability at least $1 - N^{-\ln^3(N)}$*

$$\sqrt{1-\epsilon} \; \leq \; \sigma_{|\mathcal{S}|} \left(\widetilde{R}_{\mathcal{S}} \right) \; \leq \; \sigma_1 \left(\widetilde{R}_{\mathcal{S}} \right) \; \leq \; \sqrt{1+\epsilon} \tag{10}$$

will hold simultaneously for all nonempty subsets $\mathcal{S} \subset [N]$ having $|S| \leq s$. Here the constant $C > 0$ is fixed and universal.

As pointed out in [30], the reweighted Legendre polynomials

$$Q_m(x) := \left(\frac{\pi}{2} \right)^{1/2} (1-x^2)^{1/4} L_m(x) \tag{11}$$

form a bounded orthonormal system with constant $K \leq \sqrt{3}$ with respect to the Chebyshev probability measure $d\mu(x) = \pi^{-1}(1-x^2)^{-1/2} dx$ on $\mathcal{D} = [-1,1]$. As a result, one easily obtains the following corollary of Theorem 2.

Corollary 1. *Let $R \in \mathbb{R}^{(m+1) \times (N+1)}$ be a random sampling matrix created from $\{Q_0, \ldots, Q_N\}$ in (11) via $m+1$ points $x_0, \ldots, x_m \in [-1,1]$ drawn independently according to the Chebyshev measure. Let $\epsilon \in (0,1)$, $s \in [N] \setminus \{0\}$, and set $\widetilde{R} = \frac{1}{\sqrt{m+1}} R$. If $m \geq 3C\epsilon^{-2} s \ln^4(N)$, then with probability at least $1 - N^{-\ln^3(N)}$*

$$\kappa \left(\widetilde{R}_{\mathcal{S}}^* \widetilde{R}_{\mathcal{S}} \right) \; \leq \; \frac{1+\epsilon}{1-\epsilon}$$

will hold simultaneously for all nonempty subsets $\mathcal{S} \subset [N]$ having $|S| \leq s$. The constant $C > 0$ is as in Theorem 2.

Corollary 1 guarantees that we can select a set of $m+1$ points from $[-1,1]$ *once for each value of* N which will lead to a random sampling matrix for (11), $R \in \mathbb{R}^{(m+1) \times (N+1)}$, all of whose associated

[1] So that $\mathbb{P} [\mathbf{x}_j \in \mathcal{S}] = \mu(\mathcal{S})$ for all measurable $\mathcal{S} \subseteq \mathcal{D}$ and $j \in [m]$.

[2] See Theorem 12.31 in [8].

submatrices $\widetilde{R}_{\mathcal{S}}$ are nearly isometric maps from $\mathbb{R}^{|S|}$ into $\mathbb{R}^m$. Furthermore, R can be formed in $\mathcal{O}(mN)$-time by using the standard recurrence relation for Legendre polynomials [36]

$$(m+1)L_{m+1}(x) = (2m+1)xL_m(x) - mL_{m-1}(x)$$

in order to generate each row. This is a one-time computational cost for each choice of $m, N \in \mathbb{N}$. Alternatively, in a low memory setting, one may use fast methods based asymptotic expansions in order to quickly generate any desired submatrix of $\widetilde{R}$ from Corollary 1, $\widetilde{R}_{\mathcal{S}} \in \mathbb{R}^{(m+1)\times s}$, on the fly. This can be accomplished in $\mathcal{O}(m \cdot s)$-time for any particular such submatrix of $\widetilde{R}$ as needed [4].

2.2. **Iserles' Map from Fourier to Legendre Coefficients.** For a given analytic function f and $r \in (0,1]$, define $f_r : \mathbb{C} \to \mathbb{C}$ by

$$f_r(x) := \left(1 - r^2 \mathrm{e}^{2\mathbb{i}x}\right) f\left(\frac{1}{2}\left(r^{-1}\mathrm{e}^{-\mathbb{i}x} + r\mathrm{e}^{\mathbb{i}x}\right)\right). \tag{12}$$

Here, when $r < 1$, f is evaluated on a Bernstein ellipse in the complex plane; when $r = 1$, f is composed with $\cos(x)$ as per Chebyshev interpolation. Note that f_r will be both analytic (since f is), as well as 2π-periodic on $\mathbb{R}$. Hence, both the Legendre coefficients of $f : [-1,1] \to \mathbb{R}$ and the Fourier series coefficients of $f_r : [-\pi, \pi] \to \mathbb{C}$ will decay exponentially (see, e.g., [34, 35]). In any such setting the following map may be constructed from $\widehat{f_r}$ to $\tilde{f}$ (see [16] for details).

Let $(a)_j$ be defined recursively for all $a \in \mathbb{R}$ and $j \in \mathbb{N}$ by

$$(a)_j := (a)_{j-1}\,(a+j-1)\,, \tag{13}$$

where $(a)_0 := 1$, and set

$$\tilde{g}_{i,j} := \frac{2^{2i}(i!)^2(i+1)_j(\frac{1}{2})_j}{(2i)!j!(i+\frac{3}{2})_j} \cdot r^{i+2j}, \tag{14}$$

for all $i, j \in \mathbb{N}$. Then, we have

$$\tilde{f}(i) = \sum_{j=0}^{\infty} \tilde{g}_{i,j}\ \widehat{f_r}\,(-i-2j) \tag{15}$$

for all $i \in \mathbb{N}$. Given the rapid decay of both $\tilde{g}_{i,j}$ and $\widehat{f_r}$ when $r \in (0,1)$, one may truncate the sum in order to approximate the first N Legendre coefficients using

$$\tilde{f}(i) \approx \sum_{j=0}^{M} \tilde{g}_{i,j}\ \widehat{f_r}\,(-i-2j)\,, \tag{16}$$

for a modest $M \sim \log_{1/r}(N)$, after approximating the Fourier coefficients $\widehat{f_r}\,(0), \ldots, \widehat{f_r}\,(-N-2M)$ using an FFT. For $r = 1$ (or close to 1) a modest M can still be chosen based solely on the decay of $\widehat{f_r}$. The resulting numerical method requires $\mathcal{O}(N \log N)$ floating point operations in order to approximate $\tilde{f}(i)$ for all $i \in [N]$.

Herein we are primarily concerned with the setting where $f(x)$ is a polynomial of degree at most N (recall (1)). In this case it is easy to verify that $\widehat{f_r}(\omega) = 0$ for all $\omega < -N$ and $r \in (0,1]$. Thus, the map (15) reduces to the finite linear system

$$\tilde{\mathbf{f}} := \begin{pmatrix} \tilde{f}(0) \\ \vdots \\ \tilde{f}(N) \end{pmatrix} = \tilde{G}_r \begin{pmatrix} \widehat{f_r}\,(0) \\ \vdots \\ \widehat{f_r}\,(-N) \end{pmatrix} \tag{17}$$

where $\tilde{G}_r \in \mathbb{R}^{(N+1)\times(N+1)}$ is the upper triangular matrix with entires

$$(\tilde{G}_r)_{i,j} := \begin{cases} \tilde{g}_{i,(j-i)/2} & \text{if } i \leq j, \text{ and } i \equiv j \bmod 2 \\ 0 & \text{else} \end{cases}. \tag{18}$$

We are now prepared to describe our method for rapidly and accurately computing sparse Legendre coefficient expansions.

3. A Simple SFT-based approach for Reconstructing Sparse Legendre Expansions

We propose a two stage method for approximating functions, $f : [-1, 1] \to \mathbb{R}$, with sparse/ compressible Legendre coefficient expansions as per (1). During the first stage, we use Iserles' map from §2.2 in order to help us identify Legendre polynomials whose coefficients are large in magnitude in f. We accomplish this by sampling f according to its modified form, f_r from (12), in order to take advantage of the fact that $\widehat{f_r}(-j) \approx r^{-j}\tilde{f}(j)/\sqrt{\pi j}$ holds for all $j \in [N]$ (see, e.g., Lemma 2 together with Theorem 4 in §4). This fact guarantees that the Fourier coefficients of f_r will be compressible whenever the Legendre coefficients of f, $\tilde{\mathbf{f}}$ from (17), are sparse (see, e.g., Lemma 3 in §4 for details). Hence, we may utilize SFT techniques from §1.1 in order to rapidly identify the largest magnitude Fourier coefficients of f_r which, in turn, immediately reveal the largest magnitude Legendre coefficients of f via (17).

We are happy: using Iserles' map with an SFT is good enough to guarantee that one can rapidly identify large magnitude Fourier coefficients of f_r which generally correspond to large magnitude Legendre coefficients of f! Unfortunately, this SFT-based technique does not appear to allow us to quickly compute the Legendre coefficients with much accuracy. SFTs can get us fast estimates that are "in the right ball park" – accurate enough to tell us that a coefficient is large – but getting more than a few digits of accuracy this way appears elusive. Of course, one can always force a good SFT method to supply Iserles' method (16) with enough Fourier coefficients of f_r to make it produce accurate estimates. However, this appears to be far too slow an approach to be terribly interesting for the values of r that produce decently conditioned maps $\tilde{G}_r$ (i.e., for $r \approx 1$).

Thankfully, bounded orthonormal system results for reweighted Legendre polynomials (recall §2.1) can easily solve the Legendre coefficient estimation problem for us *once we know which coefficients to compute*. Given access to a well-conditioned random sampling matrix R (recall Corollary 1), along with small vector $\mathbf{y} \in \mathbb{R}^{\mathcal{O}(s \log^4 N)}$ of additional reweighted samples from f taken at the points $x_j \in [-1, 1]$ used to build R,

$$y_j = \left(\widetilde{R}\tilde{\mathbf{f}}\right)_j = \frac{\sqrt{\pi}(1-x_j^2)^{1/4}}{\sqrt{2(m+1)}} f(x_j), \tag{19}$$

one can accurately estimate any given set of Legendre coefficients of f, $S \subset [N]$ with $|S| \leq s$, by quickly solving a small least-squares problem. In particular, this means that we can simply (i) identify a set of important Legendre coefficients of f with an SFT, and then (ii) use a random sampling matrix to accurately estimate the identified Legendre coefficients. See Algorithm 1 for pseudocode, and Lemma 4 in §4 for more details regarding coefficient estimation.

The runtime complexity of Algorithm 1 will generally be largely determined by the type of SFT chosen in line 1. Both randomized and deterministic algorithms exist. The randomized approaches are generally faster, but have a small (usually tunable) probability of failing to return a good answer. The deterministic approaches are slower, but are guaranteed to approximate a given function as well as is possible with a sparse representation of the chosen size. Recall Theorem 1 in §1.1 for example results.

Considering the runtime complexity of line 2, we note that we may efficiently solve the least squares problem there using a Conjugate Gradient (CG) algorithm. Suppose that the normalized

Algorithm 1 Fast Sparse Legendre Coefficient Expansion Algorithm

Input: (i) Sparsity $s \in [N]$, (ii) $r \in (0,1]$, (iii) Pointer to a Random Sampling Matrix, $R \in \mathbb{R}^{(m+1)\times(N+1)}$, as per Corollary 1, (iv) Renormalized samples from f, $\mathbf{y} \in \mathbb{R}^{m+1}$, as per (19), (v) Pointer to f from (1), and (vi) Pointer to a Sparse Fourier Transform code, SFT

Output: A Sparse Approximation of the Legendre Coefficients of f from (1), $\tilde{\mathbf{f}}'_{\mathcal{S}}$

1: Find the degrees of important Legendre polynomials present in f, $\mathcal{S} \subset [N]$ with $|\mathcal{S}| \leq s$, by running an SFT on f_r from (12) and recording the $\leq s$ most energetic frequencies it returns.

2: Approximately Solve a Least Squares Problem: $\tilde{\mathbf{f}}'_{\mathcal{S}} \approx \mathbf{z}_{\min} := \arg\min_{\mathbf{z}\in\mathbb{R}^{|\mathcal{S}|}} \left\| \widetilde{R}_{\mathcal{S}}\mathbf{z} - \mathbf{y} \right\|_2$.

random sampling matrix, $\widetilde{R} \in \mathbb{R}^{(m+1)\times(N+1)}$, passed to Algorithm 1 satisfies (10) of Theorem 2 with $\epsilon = 3/5$. In this case, a CG method will allow one to use just

$$C \log_{\frac{\sqrt{\kappa\left(\widetilde{R}_{\mathcal{S}}^{*}\widetilde{R}_{\mathcal{S}}\right)}+1}{\sqrt{\kappa\left(\widetilde{R}_{\mathcal{S}}^{*}\widetilde{R}_{\mathcal{S}}\right)}-1}} \left(\frac{\|\mathbf{y}\|_2}{\delta}\right) \leq C \log_{\frac{\sqrt{\frac{1+\epsilon}{1-\epsilon}}+1}{\sqrt{\frac{1+\epsilon}{1-\epsilon}}-1}} \left(\frac{\|\mathbf{y}\|_2}{\delta}\right) = C \log_3 \left(\frac{\|\mathbf{y}\|_2}{\delta}\right)$$

CG iterations in order to get

$$\left\| \widetilde{R}_{\mathcal{S}} \left(\mathbf{z}_{\min} - \tilde{\mathbf{f}}'_{\mathcal{S}} \right) \right\|_2 \leq \delta. \tag{20}$$

for any desired $\delta \in \mathbb{R}^{+}$ (see, e.g., Chapter 7 of [2]). Here, Corollary 1 (i.e., (10)) has been used to bound $\kappa\left(\widetilde{R}_{\mathcal{S}}^{*}\widetilde{R}_{\mathcal{S}}\right)$ under the assumption that $m = C' s \ln^4(N)$ for appropriate fixed universal constants $C, C' \in \mathbb{R}^{+}$. Each CG iteration then takes $\mathcal{O}\left(s^2 \ln^4(N)\right)$-time. Noting that $\|\mathbf{y}\|_2 \leq C'' \left\|\tilde{\mathbf{f}}\right\|_1$ will also hold, for a universal constant $C'' \in \mathbb{R}^{+}$, whenever $\widetilde{R}$ satisfies (10) (see, e.g., Exercise 6.6 in [8]), we have that we can compute an $\tilde{\mathbf{f}}'_{\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|}$ satisfying (20) in $\mathcal{O}\left(s^2 \ln^4(N) \cdot \ln\left(\frac{\|\tilde{\mathbf{f}}\|_1}{\delta}\right)\right)$-time.

4. Error Analysis and Recovery Guarantees

In this section we analyze Algorithm 1. The main results establish both that (i) the largest magnitude Legendre coefficients present in f can be rapidly identified via SFT methods (see Theorem 5), and that (ii) once identified, the largest magnitude Legendre coefficients can be both rapidly and accurately approximated (see Lemma 4). By combining these results one can establish deterministic[3] sublinear-time recovery guarantees for many different classes of Legendre-compressible functions. For example, one can easily prove sublinear-time recovery guarantees for exactly s-sparse Legendre polynomials of the form

$$f(x) = \sum_{n\in\mathcal{S}\subset[N]} \tilde{f}(n) L_n(x) \tag{21}$$

where $|\mathcal{S}| = s$, $\min \mathcal{S} = \Omega(N/s)$, and all s nonzero $\tilde{f}(n) \in \mathbb{R}$ have (roughly) the same magnitude. Doing so we may obtain, e.g., Theorem 3.

Theorem 3. *There exists a deterministic $\mathcal{O}\left(s^6 \log^5(N)\right)$-time algorithm that is guaranteed to exactly recover (up to machine precision) the Legendre coefficients of any function of type* (21).

[3]Note that we are implicitly using randomized techniques to construct the random sampling matrices, $R \in \mathbb{R}^{(m+1)\times(N+1)}$, used in line 2 of Algorithm 1. However, the related probabilistic guarantees establish results *for all sufficiently sparse signals* with high probability, and so can be viewed as establishing the existence of entirely deterministic methods.

Proof: Apply Corollary 3 followed by Lemma 4. □

Note that the runtime of the deterministic algorithm referred to by Theorem 3 is indeed *sublinear in N for sparsities $s \ll N$*. However, it is also almost certainly suboptimal – algorithmic modifications can probably be made that reduce the runtime complexity further without negatively impacting the recovery guarantee.

It is also important to point out that the current assumptions concerning f in (21) can be loosened considerably, without loosing deterministic recovery guarantees, by applying the subsequent results differently than done to get Theorem 3. However, the theoretical results thus derived suffer both aesthetically and, in other ways, technically. For this reason we will leave the proof of alternative guarantees via Theorem 5 and Lemma 4 to the interested reader.

We will now begin to prove our main theoretical results, starting with those concerning the rapid identification of the Legendre polynomials whose coefficients are largest in magnitude in f.

4.1. **Support Identification.** For the purposes of analyzing line 1 of Algorithm 1 it is crucial to understand how sparse

$$\widehat{\mathbf{f}}'_{\mathbf{r}} := \begin{pmatrix} \widehat{f}_r\,(0) \\ \vdots \\ \widehat{f}_r\,(-N) \end{pmatrix}$$

will be given that $\tilde{\mathbf{f}}$ is sparse (recall (17)). We will begin to move toward this goal by considering the matrix $\tilde{G}_r^{-1}$. Once it is properly understood, we will then be able to consider the compressibility characteristics of $\widehat{\mathbf{f}}'_{\mathbf{r}} = \tilde{G}_r^{-1}\tilde{\mathbf{f}}$ for sparse vectors $\tilde{\mathbf{f}}$. The following lemma gives the entries of $\tilde{G}_r^{-1}$.

Lemma 1. *The even rows of the inverse matrix $\tilde{G}_r^{-1}$ from* (17) *are given by*

$$\left(\tilde{G}_r^{-1}\right)_{2i,2j} = \begin{cases} 0 & j < i \\ \frac{(-1)^j}{4^j} \sum_{k=i}^{j} \frac{(-1)^k}{4^k} \frac{r^{-2i}(2j+2k)!}{(j-k)!(j+k)!(2k)!} \binom{2k}{k+i} \frac{1+2i}{k+i+1} & i \le j \le \left\lfloor \frac{N}{2} \right\rfloor \end{cases} \tag{22}$$

and $\left(\tilde{G}_r^{-1}\right)_{2i,2j+1} = 0$ *for* $i, j = 0, \ldots, \left\lfloor \frac{N}{2} \right\rfloor$. *Similarly, the odd rows of the inverse matrix $\tilde{G}_r^{-1}$ from* (17) *are given by*

$$\left(\tilde{G}_r^{-1}\right)_{2i+1,2j+1} = \begin{cases} 0 & j < i \\ \frac{(-1)^j}{2 \cdot 4^j} \sum_{k=i}^{j} \frac{(-1)^k}{2 \cdot 4^k} \frac{r^{-2i-1}(2j+2k+2)!}{(j-k)!(j+k+1)!(2k+1)!} \binom{2k+1}{k-i} \frac{2+2i}{k+i+2} & i \le j < \left\lfloor \frac{N}{2} \right\rfloor \end{cases} \tag{23}$$

and $\left(\tilde{G}_r^{-1}\right)_{2i+1,2j} = 0$ *for* $i, j = 0, \ldots, \left\lfloor \frac{N}{2} \right\rfloor$.

Proof: See Appendix A. □

The following corollary of Lemma 1 establishes simpler and more useful formulas for each entry of $\tilde{G}_r^{-1}$.

Corollary 2. *The even rows of the inverse matrix $\tilde{G}_r^{-1}$ from* (22) *may also be written more compactly as*

$$\left(\tilde{G}_r^{-1}\right)_{2i,2j} = \begin{cases} 0 & j < i \\ \left(-\frac{1}{4}\right)^{i+j} \frac{(2i+2j)!}{(i+j)!\ (i+j)!} \frac{r^{-2i}(2i+1)}{i+j+1} \frac{\Gamma\left(\frac{3}{2}\right)}{(j-i)!\ \Gamma\left(i-j+\frac{3}{2}\right)} & i \le j \le \left\lfloor \frac{N}{2} \right\rfloor \end{cases} \tag{24}$$

and $\left(\tilde{G}_r^{-1}\right)_{2i,2j+1} = 0$ *for* $i, j = 0, \ldots, \left\lfloor \frac{N}{2} \right\rfloor$. *Similarly, the odd rows of the inverse matrix* $\tilde{G}_r^{-1}$ *from* (23) *may be written as*

$$\left(\tilde{G}_r^{-1}\right)_{2i+1,2j+1} = \begin{cases} 0 & j < i \\ \frac{(-1)^{i+j}}{4^{i+j+1}} \frac{(2(i+j+1))!}{(i+j+1)!\ (i+j+1)!} \frac{r^{-2i-1}(2i+2)}{i+j+2} \frac{\Gamma\left(\frac{3}{2}\right)}{(j-i)!\ \Gamma\left(i-j+\frac{3}{2}\right)} & i \le j \le \left\lfloor \frac{N}{2} \right\rfloor \end{cases} \tag{25}$$

and $\left(\tilde{G}_r^{-1}\right)_{2i+1,2j} = 0$ *for* $i, j = 0, \ldots, \left\lfloor \frac{N}{2} \right\rfloor$.

Proof: We will consider the even and odd rows separately:

The Even Rows. From (22) in Lemma 1, the nonzero entries in the even rows of the inverse matrix $\tilde{G}_r^{-1}$ for $i \le j \le \left\lfloor \frac{N}{2} \right\rfloor$ can be rewritten as

$$\left(\tilde{G}_r^{-1}\right)_{2i,2j} = \left(-\frac{1}{4}\right)^{i+j} r^{-2i}(1+2i) \sum_{k=0}^{j-i} \left(-\frac{1}{4}\right)^k \frac{(2j+2i+2k)!}{((j-i)-k)!(j+i+k)!(2i+1+k)!k!}.$$

For every $i, j, k \ge 0$ and $m \in \mathbb{Z}_+$,

$$\begin{aligned} &(-m)_k = \tfrac{(-1)^k m!}{(m-k)!}, \quad (m+k)! = (m)_{k+1}(m-1)!, \quad \text{and} \\ &(2(i+j)+2k)! = 4^k\,(2(j+i))!(i+j+1)_k \left(i+j+\tfrac{1}{2}\right)_k. \end{aligned} \tag{26}$$

Thus, we deduce that

$$\begin{aligned} \left(\tilde{G}_r^{-1}\right)_{2i,2j} &= \left(-\frac{1}{4}\right)^{i+j} \frac{r^{-2i}(1+2i)}{(j-i)!} \sum_{k=0}^{j-i} \frac{(-(j-i))_k\,(2j+2i+2k)!}{4^k\,(j+i+k)!(2i+1+k)!k!} \\ &= \left(-\frac{1}{4}\right)^{i+j} \frac{r^{-2i}(1+2i)(2i+2j)!}{(j-i)!(j+i-1)!(2i)!} \sum_{k=0}^{j-i} \frac{(-(j-i))_k (j+i+1)_k \left(j+i+\frac{1}{2}\right)_k}{k!(j+i)_{k+1}(2i+1)_{k+1}} \\ &= \left(-\frac{1}{4}\right)^{i+j} \frac{r^{-2i}(2i+2j)!}{(j-i)!(j+i)!(2i)!} \sum_{k=0}^{j-i} \frac{(-(j-i))_k \left(j+i+\frac{1}{2}\right)_k}{k!(2i+2)_k} \end{aligned}$$

The *hypergeometric function* F can be expressed by

$$F(-m, \beta; \gamma; z) = \sum_{k=0}^{m} \frac{(-m)_k(\beta)_k}{(\gamma)_k} \frac{z^k}{k!},$$

where m is a positive integer and γ is neither zero nor a negative integer (see, e.g., [27]). Therefore,

$$\left(\tilde{G}_r^{-1}\right)_{2i,2j} = \left(-\frac{1}{4}\right)^{i+j} \frac{r^{-2i}(2i+2j)!}{(j-i)!(j+i)!(2i)!} F\left(-(j-i), \left(j+i+\frac{1}{2}\right); (2i+2); 1\right).$$

The special value of $F(\alpha,\beta;\gamma,z)$ at $z=1$ can be expressed in terms of a Gamma function [36]. That is, $F(\alpha,\beta;\gamma,1)=\left(\Gamma(\gamma)\Gamma(\gamma-\alpha-\beta)\right)/\left(\Gamma(\gamma-\alpha)\Gamma(\gamma-\beta)\right)$. Thus, we deduce that

$$\begin{aligned}\left(\tilde{G}_r^{-1}\right)_{2i,2j} &= \left(-\frac{1}{4}\right)^{i+j}\frac{r^{-2i}(2i+2j)!}{(j-i)!(j+i)!(2i)!}\frac{\Gamma(2i+2)\Gamma(\frac{3}{2})}{\Gamma(i+j+2)\Gamma(i-j+\frac{3}{2})}\\ &= \left(\left(-\frac{1}{4}\right)^{i+j}\frac{(2i+2j)!}{(j+i)!(j+i)!}\right)\left(\frac{r^{-2i}(2i+1)}{(i+j+1)}\right)\left(\frac{\Gamma(\frac{3}{2})}{(j-i)!\Gamma(i-j+\frac{3}{2})}\right). \qquad (27)\end{aligned}$$

The Odd Rows. Similarly, from (23) in Lemma 1, the nonzero entries in the odd rows of the inverse matrix $\tilde{G}_r^{-1}$ for $i\le j\le\left\lfloor\frac{N}{2}\right\rfloor$ can be rewritten as

$$\left(\tilde{G}_r^{-1}\right)_{2i+1,2j+1}=\frac{(-1)^{i+j}}{4^{i+j+1}}r^{-2i-1}(2+2i)\sum_{k=0}^{j-i}\left(-\frac{1}{4}\right)^k\frac{(2j+2i+2k+2)!}{((j-i)-k)!(j+i+1+k)!(2i+2+k)!k!}.$$

By using (26) we deduce that

$$\begin{aligned}\left(\tilde{G}_r^{-1}\right)_{2i+1,2j+1} &= \frac{(-1)^{i+j}}{4^{i+j+1}}\frac{r^{-2i-1}(2+2i)}{(j-i)!}\sum_{k=0}^{j-i}\frac{(-(j-i))_k\,(2(j+i+1)+2k)!}{4^k\,(j+i+1+k)!(2i+2+k)!k!}\\ &= \frac{(-1)^{i+j}}{4^{i+j+1}}\frac{r^{-2i-1}(2+2i)(2(j+i+1))!}{(j-i)!(j+i)!(2i+1)!}\sum_{k=0}^{j-i}\frac{(-(j-i))_k(j+i+2)_k\left(j+i+\frac{3}{2}\right)_k}{k!(j+i+1)_{k+1}(2i+2)_{k+1}}\\ &= \frac{(-1)^{i+j}}{4^{i+j+1}}\frac{r^{-2i-1}(2(j+i+1))!}{(j-i)!(j+i+1)!(2i+1)!}\sum_{k=0}^{j-i}\frac{(-(j-i))_k\left(j+i+\frac{3}{2}\right)_k}{k!(2i+3)_k}\\ &= \frac{(-1)^{i+j}}{4^{i+j+1}}\frac{r^{-2i-1}(2(j+i+1))!}{(j-i)!(j+i+1)!(2i+1)!}F\left(-(j-i),\left(j+i+\frac{3}{2}\right);(2i+3);1\right),\end{aligned}$$

where F is a hypergeometric function. Again, by using a Gamma function for the special value of F at $z=1$, we finally deduce that

$$\begin{aligned}\left(\tilde{G}_r^{-1}\right)_{2i+1,2j+1} &= \frac{(-1)^{i+j}}{4^{i+j+1}}\frac{r^{-2i-1}(2(j+i+1))!}{(j-i)!(j+i+1)!(2i+1)!}\frac{\Gamma(2i+3)\Gamma(\frac{3}{2})}{\Gamma(i+j+3)\Gamma(i-j+\frac{3}{2})}\\ &= \left(\frac{(-1)^{i+j}}{4^{i+j+1}}\frac{(2(j+i+1))!}{(j+i+1)!(j+i+1)!}\right)\left(\frac{r^{-2i-1}(2i+2)}{(i+j+2)}\right)\left(\frac{\Gamma(\frac{3}{2})}{(j-i)!\Gamma(i-j+\frac{3}{2})}\right). \qquad (28)\end{aligned}$$

□

With Corollary 2 in hand, one may now see that each row and column of $\tilde{G}_r^{-1}$ is weakly dominated by it's diagonal entry. See Theorem 4 below for an exact statement.

Theorem 4. *For nonzero entries of the inverse matrix $\tilde{G}_r^{-1}$, the decay rate of each row is given by*

$$\begin{aligned}\left|(\tilde{G}_r^{-1})_{n,n+2x}\right| &< \tfrac{\sqrt[3]{e}}{2\sqrt{\pi}}\tfrac{1}{x\sqrt{x-2}}\left|(\tilde{G}_r^{-1})_{n,n}\right| \quad \text{for } x>2,\\ \left|(\tilde{G}_r^{-1})_{n,n+2}\right| < \tfrac{1}{2}\left|(\tilde{G}_r^{-1})_{n,n}\right|,\ \text{and}\ \left|(\tilde{G}_r^{-1})_{n,n+4}\right| &< \tfrac{1}{8}\left|(\tilde{G}_r^{-1})_{n,n}\right| \quad \text{for } x=1,x=2.\end{aligned} \qquad (29)$$

The decay rate of each column is given by

$$\begin{aligned}\left|(\tilde{G}_r^{-1})_{n-2x,n}\right| &\le \tfrac{\sqrt[3]{e}}{\sqrt{2\pi}}\tfrac{r^{2x}}{x\sqrt{x-2}}\left|(\tilde{G}_r^{-1})_{n,n}\right| \quad \text{for } 2<x\le\left\lfloor\tfrac{n}{2}\right\rfloor,\ n\ge 6\\ \left|(\tilde{G}_r^{-1})_{n-2,n}\right| < \tfrac{r^2}{2}\left|(\tilde{G}_r^{-1})_{n,n}\right|,\ \text{and}\left|(\tilde{G}_r^{-1})_{n-4,n}\right| &< \tfrac{r^4}{4}\left|(\tilde{G}_r^{-1})_{n,n}\right| \quad \text{for } x=1,x=2,n\ge 2x.\end{aligned} \qquad (30)$$

The diagonal entries of $\tilde{G}_r^{-1}$ satisfy

$$\frac{r^{-n}}{\sqrt{\pi n}}\left(1-\frac{1}{8n}\right) \le \left|(\tilde{G}_r^{-1})_{n,n}\right| \le \frac{r^{-n}}{\sqrt{\pi n}} \text{ for } n>0, \text{ and } \left|(\tilde{G}_r^{-1})_{0,0}\right| = 1. \tag{31}$$

Proof: First, let's simplify the term $\Gamma\left(\frac{3}{2}\right) / \left((j-i)!\Gamma\left(i-j+\frac{3}{2}\right)\right)$. Since $\Gamma(3/2) = \sqrt{\pi}/2$, and by the reflection formula for the Gamma function (see, e.g., [36]), $\Gamma(-z) = -\pi/\left(z\Gamma(z)\sin(\pi z)\right)$ for $z \notin \mathbb{Z}$, we have

$$\frac{\Gamma\left(\frac{3}{2}\right)}{(j-i)!\Gamma\left(i-j+\frac{3}{2}\right)} = \frac{(-1)^{j-i+1}}{2\sqrt{\pi}} \frac{(j-i-\frac{3}{2})\Gamma(j-i-\frac{3}{2})}{(j-i)!}.$$

According to the half integer argument for the Gamma function, $\Gamma(n/2) = \left((n-2)!!\sqrt{\pi}\right)/2^{(n-1)/2}$ for $n \in \mathbb{Z}^+$, where $n!! = (2k)!/\left(2^k k!\right)$ when $n = 2k-1$ for $k \in \mathbb{Z}^+$. Thus,

$$\begin{aligned}\frac{\Gamma\left(\frac{3}{2}\right)}{(j-i)!\Gamma\left(i-j+\frac{3}{2}\right)} &= \frac{(-1)^{j-i+1}}{2} \frac{(j-i-\frac{3}{2})}{(j-i)!} \frac{(2j-2i-5)!!}{2^{j-i-2}} \\ &= \frac{(-1)^{j-i+1}}{2\cdot 4^{j-i-2}} \frac{(j-i-\frac{3}{2})}{(j-i)(j-i-1)} \frac{(2(j-i-2))!}{(j-i-2)!(j-i-2)!}.\end{aligned}$$

By Stirling's approximation, $n! = \sqrt{2\pi n} n^n \mathrm{e}^{-n} \mathrm{e}^{R_n}$, where $0 \le R_n \le 1/12n$ (see, e.g., [31, 8]),

$$\frac{\Gamma\left(\frac{3}{2}\right)}{(j-i)!\Gamma\left(i-j+\frac{3}{2}\right)} = \frac{(-1)^{j-i+1}}{2\sqrt{\pi}} \frac{(j-i-\frac{3}{2})}{(j-i)(j-i-1)\sqrt{j-i-2}} \mathrm{e}^{R_{2(j-i-2)}-2R_{j-i-2}}. \tag{32}$$

Now by Corollary 2 and (32), if $n = 2i$ for $i, x \in \mathbb{Z}$, $i \ge 0$, and $x > 2$, then

$$\left|(\tilde{G}_r^{-1})_{n,n+2x}\right| = \left|(\tilde{G}_r^{-1})_{n,n}\right| \cdot \left|\tfrac{1}{4^x} \tfrac{(n!)^2}{(2n)!} \tfrac{(2n+2x)!}{(n+x)!(n+x)!} \tfrac{n+1}{n+x+1} \tfrac{1}{2\sqrt{\pi}} \tfrac{x-\frac{3}{2}}{x(x-1)\sqrt{x-2}} \mathrm{e}^{R_{2(x-2)}-2R_{x-2}}\right| \tag{33}$$

Using Stirling's approximation again for $i \ge 1$ we have

$$\left|(\tilde{G}_r^{-1})_{n,n+2x}\right| = \left|(\tilde{G}_r^{-1})_{n,n}\right| \left(\tfrac{\sqrt{n}}{\sqrt{n+x}} \tfrac{n+1}{n+x+1} \tfrac{(x-\frac{3}{2})}{x-1} \tfrac{1}{x\sqrt{x-2}} \tfrac{1}{2\sqrt{\pi}} \mathrm{e}^{R^{\mathrm{a}}}\right), \tag{34}$$

where $R^{\mathrm{a}} = 2R_n - R_{2n} + R_{2n+2x} - 2R_{n+x} + R_{2(x-2)} - 2R_{x-2} \le 1/3$ for all $n, x \in \mathbb{Z}^+$ with $x \ge 3$. According to (33), when $i = 0$, it is easy to show that

$$\left|(\tilde{G}_r^{-1})_{0,2x}\right| \le \left|(\tilde{G}_r^{-1})_{0,0}\right| \cdot \frac{\sqrt[6]{\mathrm{e}}}{2\pi} \frac{1}{(x+1)} \frac{1}{x^{3/2}\sqrt{x-2}}, \text{ for } x > 2. \tag{35}$$

Similarly, if $n = 2i+1$ for $i \ge 0$ it is not difficult to verify that (34) still holds. Therefore, one can see that each row satisfies

$$\left|(\tilde{G}_r^{-1})_{n,n+2x}\right| < \frac{\sqrt[3]{\mathrm{e}}}{2\sqrt{\pi}} \frac{1}{x\sqrt{x-2}} \left|(\tilde{G}_r^{-1})_{n,n}\right|.$$

The remainder of the proof of (29) is now easily established using Corollary 2.

Similarly, we can bound the rate of decay of each column off of the diagonal. By Corollary 2, (32), and Stirling's approximation, if $n = 2i$ for $i \ge 3$, $2 < x \le \left\lfloor \frac{n}{2} \right\rfloor$, then

$$\left|(\tilde{G}_r^{-1})_{n-2x,n}\right| = \left|(\tilde{G}_r^{-1})_{n,n}\right| \cdot \tfrac{r^{2x}}{2\sqrt{\pi}} \left(\sqrt{\tfrac{n}{n-x}}\right) \tfrac{n-2x+1}{n-x+1} \left(\tfrac{x-\frac{3}{2}}{x-1}\right) \tfrac{\mathrm{e}^{\tilde{R}^{\mathrm{a}}}}{x\sqrt{x-2}}$$

where $\tilde{R}^{\mathrm{a}} = 2R_n - R_{2n} + R_{2(n-x)} - 2R_{n-x} + R_{2(x-2)} - 2R_{x-2} \le 1/3$ for all $n, x \in \mathbb{Z}^+$ with $x \ge 3$. Since $\sqrt{n/(n-x)} \le \sqrt{2}$ for $2 < x \le i$,

$$\left|(\tilde{G}_r^{-1})_{n-2x,n}\right| < \left|(\tilde{G}_r^{-1})_{n,n}\right| \cdot \tfrac{r^{2x}}{\sqrt{2\pi}} \tfrac{\sqrt[3]{\mathrm{e}}}{x\sqrt{x-2}}. \tag{36}$$

For $n = 2i + 1$ an analogous calculation reveals that (36) still holds. The remainder of the proof of (30) is now easily established using Corollary 2. Lastly, the proof of (31) is easily established using Lemma 1 together with Theorem 2.6 in [33]. □

We are now in the position to begin studying the compressibility of $\widehat{\mathbf{f}}'_{\mathbf{r}} = \tilde{G}_r^{-1}\tilde{\mathbf{f}}$ in terms of the compressibility of $\tilde{\mathbf{f}}$. Let $\sigma : [N] \to [N]$ be a permutation of $[N]$ such that

$$\left|\tilde{f}_{\sigma(j)}\right| \geq \left|\tilde{f}_{\sigma(j+1)}\right|$$

holds for all $j \in [N-1]$. Let $\rho : \mathbb{R} \to \mathbb{R}^+ \cup \{\infty\}$ be a modified ramp function with $\rho(x) = x$ for all $x \in \mathbb{R}^+$, and $\rho(x) = \infty$ for all $x < 0$. Finally, define the right-distance form $j \in [N]$ to the set $\left\{\sigma(n) \;\middle|\; n \in [s-1],\ \sigma(n) \equiv j \bmod 2\right\}$ to be

$$d_s(j) := \min\left(\left\{\frac{\rho\left(\sigma(n) - j\right)}{2} \;\middle|\; n \in [s-1],\ \sigma(n) \equiv j \bmod 2\right\} \cup \{\infty\}\right).$$

We now have sufficient notation to consider the sizes of specific entries of $\widehat{\mathbf{f}}'_{\mathbf{r}}$.

Lemma 2. *Let $s \in [N] \setminus \{0\}$. We have that*

$$\left|\left(\widehat{f}'_r\right)_j - (\tilde{G}_r^{-1})_{j,j}\tilde{f}_j\right| < \frac{31}{20}\left|(\tilde{G}_r^{-1})_{j,j}\right|\left|\tilde{f}_{\sigma(s)}\right| + \frac{\sqrt[3]{e}}{2\sqrt{\pi}}\left(\frac{\|\tilde{\mathbf{f}}\|_1}{d_s(j)\sqrt{d_s(j)-2}}\right)\left|(\tilde{G}_r^{-1})_{j,j}\right|$$

holds for all $j \in [N]$ with $d_s(j) > 2$.

Proof: Using that $\widehat{\mathbf{f}}'_{\mathbf{r}} = \tilde{G}_r^{-1}\tilde{\mathbf{f}}$ with $\tilde{G}_r^{-1}$ upper triangular as per Lemma 1, we have that

$$\left(\widehat{f}'_r\right)_j = (\tilde{G}_r^{-1})_{j,j}\tilde{f}_j + \sum_{l=1}^{\left\lfloor\frac{N-j}{2}\right\rfloor}(\tilde{G}_r^{-1})_{j,j+2l}\tilde{f}_{j+2l}.$$

Rearranging this expression we can see that

$$\begin{aligned}\left|\left(\widehat{f}'_r\right)_j - (\tilde{G}_r^{-1})_{j,j}\tilde{f}_j\right| &\leq \sum_{l=1}^{d_s(j)-1}\left|(\tilde{G}_r^{-1})_{j,j+2l}\right|\left|\tilde{f}_{j+2l}\right| + \sum_{l=d_s(j)}^{\left\lfloor\frac{N-j}{2}\right\rfloor}\left|(\tilde{G}_r^{-1})_{j,j+2l}\right|\left|\tilde{f}_{j+2l}\right| \\ &< \left|\tilde{f}_{\sigma(s)}\right|\left(\sum_{l=1}^{d_s(j)-1}\left|(\tilde{G}_r^{-1})_{j,j+2l}\right|\right) + \frac{\sqrt[3]{\mathrm{e}}}{2\sqrt{\pi}}\left(\frac{\|\tilde{\mathbf{f}}\|_1}{d_s(j)\sqrt{d_s(j)-2}}\right)\left|(\tilde{G}_r^{-1})_{j,j}\right|,\end{aligned}$$

where the second inequality follows from Hölder's Inequality, Theorem 4, and the definition of $d_s(j)$. Appealing to Theorem 4 again we can also see that

$$\sum_{l=1}^{d_s(j)-1}\left|(\tilde{G}_r^{-1})_{j,j+2l}\right| < \left|(\tilde{G}_r^{-1})_{j,j}\right|\left(\frac{1}{2} + \frac{1}{8} + \frac{\sqrt[3]{\mathrm{e}}}{6\sqrt{\pi}} + \frac{\sqrt[3]{\mathrm{e}}}{2\sqrt{\pi}}\int_3^\infty \frac{dx}{x\sqrt{x-2}}\right).$$

The remainder of the proof now follows. □

We can now begin to understand the compressibility of $\widehat{\mathbf{f}}'_{\mathbf{r}}$. In particular, we have the following result concerning the best $s \cdot k$ approximation to $\widehat{\mathbf{f}}'_{\mathbf{1}}$ for any $k \in \mathbb{Z}^+$ with $k > 5$.

Lemma 3. *Let $s, k \in [N]$ with $k > 5$, $s > 8$, and $n_{\min} := \min\left\{\sigma(n) \;\middle|\; n \in [s-1]\right\} > k$. Then,*

$$\left\|\widehat{\mathbf{f}}'_{\mathbf{1}} - \left(\widehat{\mathbf{f}}'_{\mathbf{1}}\right)^{\mathrm{opt}}_{s\cdot k}\right\|_1 < 7\sqrt{N}\left|\tilde{f}_{\sigma(s)}\right| + \frac{s\|\tilde{\mathbf{f}}\|_1}{\sqrt{n_{\min}-4}\sqrt{k-5}}.$$

Proof: We bound the sum of the magnitudes of all entries in $\widehat{\mathbf{f}}_{\mathbf{1}}'$ whose indices have right-distance $\geq k$ from $\left\{\sigma(n) \;\middle|\; n \in [s-1]\right\}$ using two cases:

Case I. We bound the magnitudes of the small entries with $j \leq n_{\min} - k$. Using Lemma 2 we can see that

$$\sum_{j=0}^{n_{\min}-k} \left|\left(\widehat{f}_1'\right)_j\right| < \sum_{j=0}^{n_{\min}-k} \left|(\tilde{G}_r^{-1})_{j,j}\right| \left(\frac{51}{20}\left|\tilde{f}_{\sigma(s)}\right| + \frac{\sqrt[3]{\mathrm{e}}}{\sqrt{\pi/2}}\left(\frac{\|\tilde{\mathbf{f}}\|_1}{(n_{\min}-j)\sqrt{n_{\min}-j-4}}\right)\right).$$

Theorem 4 now implies that

$$\begin{aligned}
\sum_{j=0}^{n_{\min}-k} \left|\left(\widehat{f}_1'\right)_j\right| &< \frac{51\sqrt{2\cdot n_{\min}}}{10\sqrt{\pi}}\left|\tilde{f}_{\sigma(s)}\right| + \frac{2\|\tilde{\mathbf{f}}\|_1}{n_{\min}-1} + \frac{\sqrt[3]{\mathrm{e}}\sqrt{2}\|\tilde{\mathbf{f}}\|_1}{\pi}\cdot\sum_{j=2}^{n_{\min}-k}\frac{1}{\sqrt{j}(n_{\min}-j-4)^{\frac{3}{2}}} \\
&< \frac{51\sqrt{2\cdot n_{\min}}}{10\sqrt{\pi}}\left|\tilde{f}_{\sigma(s)}\right| + \frac{2\|\tilde{\mathbf{f}}\|_1}{n_{\min}-1} + \frac{\sqrt[3]{\mathrm{e}}\sqrt{2}\|\tilde{\mathbf{f}}\|_1}{\pi}\cdot\int_1^{n_{\min}-k+1}\frac{dx}{\sqrt{x}\,(n_{\min}-4-x)^{3/2}} \\
&< 4.1\sqrt{n_{\min}}\left|\tilde{f}_{\sigma(s)}\right| + \frac{4\|\tilde{\mathbf{f}}\|_1}{\sqrt{n_{\min}-4}\sqrt{k-5}}.
\end{aligned}$$

Case II. Here we bound the sum of the magnitudes of all entries $\left|\left(\widehat{f}_1'\right)_j\right|$ with $j \in A := \{x \mid x > n_{\min}$ and $d_s(x) \geq k\}$. Note that there can be at most $(s-1)k$ entries in $[N] \setminus (A \cup [n_{\min}])$. Using Lemma 2, Theorem 4, and definition of A we can see that

$$\begin{aligned}
\sum_{j\in A}\left|\left(\widehat{f}_1'\right)_j\right| &< \frac{51}{20\sqrt{\pi}}\left|\tilde{f}_{\sigma(s)}\right|\left(\sum_{j=n_{\min}+1}^{N}\frac{1}{\sqrt{j}}\right) + \frac{\sqrt[3]{\mathrm{e}}(s-1)}{2\pi\sqrt{n_{\min}}}\left(\sum_{l=k}^{\infty}\frac{\|\tilde{\mathbf{f}}\|_1}{l\sqrt{l-2}}\right) \\
&< \frac{51\sqrt{N}}{10\sqrt{\pi}}\left|\tilde{f}_{\sigma(s)}\right| + \frac{\sqrt[3]{\mathrm{e}}(s-1)}{\pi\sqrt{n_{\min}}}\frac{\|\tilde{\mathbf{f}}\|_1}{\sqrt{k-3}}.
\end{aligned}$$

Combining the bounds from Cases I and II now finishes the proof. □

Note that the vector $\widehat{\mathbf{f}}_{\mathbf{r}}'$ contains only the (potentially nonzero) negative Fourier series coefficients of $f_1(x) = \left(1 - \mathbb{e}^{2\mathbb{i}x}\right) f\left(\cos(x)\right)$. Let

$$\widehat{\mathbf{f}}_{\mathbf{1}} := \begin{pmatrix} \widehat{f}_1\left(N+2\right) \\ \vdots \\ \widehat{f}_1\left(0\right) \\ \vdots \\ \widehat{f}_1\left(-N\right) \end{pmatrix} \in \mathbb{R}^{2N+3}$$

consist of all the potentially nonzero Fourier series coefficients of $f_1(x)$. Noting that $f\left(\cos(x)\right)$ is an even real-valued function, one can see that $\widehat{f}_1\left(\omega\right) = -\widehat{f}_1\left(2-\omega\right)$ holds for all $\omega \in \mathbb{Z}^+$ (with $\widehat{f}_1\left(1\right) = 0$). As a result, Lemmas 2 and 3 trivially extend to $\widehat{\mathbf{f}}_{\mathbf{1}}$. Using this fact in combination with results from [18] finally allows us to prove the main result of this section.

Theorem 5. *Let $s, k \in [N]$ with $k > 5$, $s > 8$, $\mathcal{S} := \left\{\sigma(n) \;\middle|\; n \in [s-1]\right\}$, and $n_{\min} := \min \mathcal{S} > k$. Given $j \in \mathcal{S}$, define the new right-distance from j to $\mathcal{S}$ to be*

$$d'_s(j) := \min\left(\left\{\frac{\rho\left(\sigma(n) - j\right)}{2} \;\middle|\; n \in [s-1],\ \sigma(n) \equiv j \bmod 2\right\} \setminus \{0\} \cup \{\infty\}\right) > 0 = d_s(j).$$

Then, the deterministic variant of the algorithm referred to by Theorem 1 will recover all $j \in \mathcal{S}$ satisfying both

$$\left|\tilde{f}_j\right| \geq \left(\frac{224\sqrt{\pi}N}{7s \cdot k} + \frac{31}{20}\right)\left|\tilde{f}_{\sigma(s)}\right| + \frac{17\sqrt{N}\|\tilde{\mathbf{f}}\|_1}{2k\sqrt{(n_{\min}-4)(k-5)}}, \tag{37}$$

and $d'_s(j) \geq k$. Its operation count will be

$$\mathcal{O}\left(s^2 \cdot k^2 \cdot \log^4 N\right).$$

Proof: We consider the behavior of the deterministic variant of the algorithm referred to by Theorem 1 when executed on $f_1(x) = \left(1 - \mathbb{e}^{2\mathbb{i}x}\right) f\left(\cos(x)\right)$ with sparsity parameter $2s$ and $\epsilon = 1/k$. In the course of forming its output trigonometric polynomial, y_s, this algorithm is guaranteed to identify every $j \in [-N, N+2] \cap \mathbb{Z}$ with the property that

$$\left|\widehat{f_1}\left(j\right)\right| > \frac{2\left\|\widehat{\mathbf{f_1}} - \left(\widehat{\mathbf{f_1}}\right)_{2s\cdot k}^{\text{opt}}\right\|_1}{s \cdot k}$$

by Lemma 6 in [18].

Suppose that $\tilde{f}_j$ satisfies (37) and also has $d'_s(j) \geq k$. A trivial variant of Lemma 2 then implies that

$$\left|\widehat{f_1}\left(-j\right)\right| > \left|(\tilde{G}_r^{-1})_{j,j}\right|\left|\tilde{f}_j\right| - \frac{31}{20}\left|(\tilde{G}_r^{-1})_{j,j}\right|\left|\tilde{f}_{\sigma(s)}\right| - \frac{\sqrt[3]{\mathbb{e}}}{2\sqrt{\pi}}\left(\frac{\|\tilde{\mathbf{f}}\|_1}{d'_s(j)\sqrt{d'_s(j)-2}}\right)\left|(\tilde{G}_r^{-1})_{j,j}\right|.$$

Using (37) and that $d'_s(j) \geq k$ one can now see that

$$\left|\widehat{f_1}\left(-j\right)\right| > \left|(\tilde{G}_r^{-1})_{j,j}\right|\left(\frac{28 \cdot 8\sqrt{\pi}N}{7s \cdot k}\left|\tilde{f}_{\sigma(s)}\right| + \frac{32\sqrt{\pi N}\|\tilde{\mathbf{f}}\|_1}{7k\sqrt{(n_{\min}-4)(k-5)}}\right).$$

Theorem 4 followed by Lemma 3 finally implies that

$$\begin{aligned}
\left|\widehat{f_1}\left(-j\right)\right| &> \frac{28 \cdot \sqrt{N}}{s \cdot k}\left|\tilde{f}_{\sigma(s)}\right| + \frac{4\|\tilde{\mathbf{f}}\|_1}{k\sqrt{(n_{\min}-4)(k-5)}} \\
&= \frac{2}{s \cdot k}\left(14\sqrt{N}\left|\tilde{f}_{\sigma(s)}\right| + \frac{2s\|\tilde{\mathbf{f}}\|_1}{\sqrt{n_{\min}-4}\sqrt{k-5}}\right) \\
&> \frac{2\left\|\widehat{\mathbf{f_1}} - \left(\widehat{\mathbf{f_1}}\right)_{2s\cdot k}^{\text{opt}}\right\|_1}{s \cdot k}.
\end{aligned}$$

This guarantees that j will indeed be identified as claimed. □

The final corollary of this section applies Theorem 5 to the situation of exactly s-sparse vectors $\tilde{\mathbf{f}}$. We have the following result:

Corollary 3. *Let $s \in [N]$ with $s > 8$, $\mathcal{S} := \left\{ \sigma(n) \;\middle|\; n \in [s-1] \right\}$, and $n_{\min} := \min \mathcal{S} > \frac{N}{17s/2-5} + 4 > 17s/2$. Furthermore, suppose that $\tilde{\mathbf{f}}$ is exactly s-sparse with $\left|\tilde{f}_{\sigma(0)}\right| = \left|\tilde{f}_{\sigma(s-1)}\right| > \tilde{f}_{\sigma(s)} = 0$. Then, there exists a deterministic algorithm which will recover a set of cardinality $\mathcal{O}\left(s^3\right)$ that is guaranteed to contain $\mathcal{S}$ as a subset. Its operation count will be $\mathcal{O}\left(s^4 \cdot \log^4 N\right)$.*

Proof: Apply Theorem 5 with $k = 17s/2$. The deterministic variant of the algorithm referred to by Theorem 1 will recover all $j \in \mathcal{S}$ satisfying both $d'_s(j) \geq 17s/2$, and

$$\left|\tilde{f}_j\right| \geq \frac{\sqrt{N}\|\tilde{\mathbf{f}}\|_1}{s\sqrt{(n_{\min}-4)(17s/2-5)}}, \text{ where } \frac{\sqrt{N}\|\tilde{\mathbf{f}}\|_1}{s\sqrt{(n_{\min}-4)(17s/2-5)}} < \frac{\|\tilde{\mathbf{f}}\|_1}{s} = \left|\tilde{f}_{\sigma(s-1)}\right|,$$

as a subset of a set $\mathcal{A}$ of cardinality $\mathcal{O}(s^2)$. Returning all $j \in [N]$ whose right-distance to $\mathcal{A}$ is $< 17s/2$ provides a set of cardinality $\mathcal{O}\left(s^3\right)$ that is guaranteed to contain $\mathcal{S}$ as a subset. The result follows. □

We conclude this section by mentioning several obvious facts regarding Theorem 5 and Corollary 3. First, we have no doubt that they can be improved in general. The easiest way to do this is to utilize randomized SFT methods in place of the deterministic algorithm from Theorem 1 in order to reduce the computational complexities involved. It is also clear that the assumptions in Corollary 3 can be relaxed rather easily at the price of increased computational costs. Less trivial improvements would probably revolve around developing better variants of $f_1(x)$ whose Fourier coefficients are less contaminated by Legendre coefficients that are "far away" from the associated ones (i.e., $\tilde{G}_r^{-1}$ should be "more diagonally dominant"). Alternatively, one might design efficient filtering schemes which gradually reveal the lower (generally more energetic) frequencies of $f_1(x)$ so that they do not overwhelm larger (generally less energetic) frequencies as we hunt for them. However, we will leave such considerations for future work. We are now ready to consider how accurately we can estimate the Legendre coefficients for the set of supporting polynomials, $\mathcal{S} \subset [N]$, we identify in this section.

4.2. **Coefficient Estimation.** Estimating the Legendre coefficients for the polynomials that have been identified as present in f is comparatively easy given all the previous work on bounded orthonormal systems. We have the following result:

Lemma 4. *Suppose that the normalized random sampling matrix, $\widetilde{R} \in \mathbb{R}^{(m+1)\times(N+1)}$, passed to Algorithm 1 satisfies* (10) *of Theorem 2 with $\epsilon = 3/5$. Let $\delta \in \mathbb{R}^+$, and $\mathcal{S} \subset [N]$ have cardinality $|\mathcal{S}| = s$. Then line 2 of of Algorithm 1 can produce an s-sparse vector, $\mathbf{z} \in \mathbb{R}^{N+1}$, satisfying*

$$\left\|\tilde{\mathbf{f}} - \mathbf{z}\right\|_2 \;\leq\; 5\left\|\tilde{\mathbf{f}}_{\mathcal{S}^c}\right\|_2 + \frac{4\left\|\tilde{\mathbf{f}}_{\mathcal{S}^c}\right\|_1}{\sqrt{s}} + \sqrt{\frac{5}{2}}\delta$$

in $\mathcal{O}\left(s^2 \ln^4(N) \cdot \ln\left(\frac{\|\tilde{\mathbf{f}}\|_1}{\delta}\right)\right)$-time.

Proof: Let $\tilde{\mathbf{f}}''_{\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|}$ be such that $\widetilde{R}_{\mathcal{S}}\tilde{\mathbf{f}}''_{\mathcal{S}} = \widetilde{R}\tilde{\mathbf{f}}_{\mathcal{S}}$ (i.e., let $\tilde{\mathbf{f}}''_{\mathcal{S}}$ be $\tilde{\mathbf{f}}_{\mathcal{S}}$ with all it's zero-valued entries indexed by $\mathcal{S}^{\mathrm{c}}$ removed). Let $\tilde{\mathbf{f}}'_{\mathcal{S}}, \mathbf{z}_{\min} \in \mathbb{R}^{|\mathcal{S}|}$ be defined as in line 2 of Algorithm 1. We have that

$$\begin{aligned}
\left\|\tilde{\mathbf{f}}'_{\mathcal{S}} - \tilde{\mathbf{f}}''_{\mathcal{S}}\right\|_2 &\leq \tfrac{\left\|\widetilde{R}_{\mathcal{S}}\tilde{\mathbf{f}}'_{\mathcal{S}} - \widetilde{R}_{\mathcal{S}}\tilde{\mathbf{f}}''_{\mathcal{S}}\right\|_2}{\sqrt{1-\epsilon}} && \text{Since (10) holds} \\
&\leq \tfrac{\left\|\widetilde{R}_{\mathcal{S}}\mathbf{z}_{\min} - \widetilde{R}_{\mathcal{S}}\tilde{\mathbf{f}}''_{\mathcal{S}}\right\|_2 + \delta}{\sqrt{1-\epsilon}} && \text{By CG discussion (20) in §3} \\
&\leq \tfrac{1}{\sqrt{1-\epsilon}}\left(\left\|\widetilde{R}_{\mathcal{S}}\mathbf{z}_{\min} - \mathbf{y}\right\|_2 + \left\|\widetilde{R}\tilde{\mathbf{f}}_{\mathcal{S}^{\mathrm{c}}}\right\|_2 + \delta\right) && \text{Since } \mathbf{y} = \widetilde{R}\left(\tilde{\mathbf{f}}_{\mathcal{S}} + \tilde{\mathbf{f}}_{\mathcal{S}^{\mathrm{c}}}\right) \\
&\leq \tfrac{2}{\sqrt{1-\epsilon}}\left\|\widetilde{R}\tilde{\mathbf{f}}_{\mathcal{S}^{\mathrm{c}}}\right\|_2 + \tfrac{\delta}{\sqrt{1-\epsilon}} && \text{By the definition of } \mathbf{z}_{\min} \\
&\leq \tfrac{2\sqrt{1+\epsilon}}{\sqrt{1-\epsilon}}\left(\left\|\tilde{\mathbf{f}}_{\mathcal{S}^{\mathrm{c}}}\right\|_2 + \tfrac{\left\|\tilde{\mathbf{f}}_{\mathcal{S}^{\mathrm{c}}}\right\|_1}{\sqrt{s}}\right) + \tfrac{\delta}{\sqrt{1-\epsilon}} && \text{Using Exercise 6.6 in [8]}
\end{aligned}$$

One can now (implicitly) form $\mathbf{z} = \mathbf{z}_{\mathcal{S}}$ from $\tilde{\mathbf{f}}'_{\mathcal{S}}$ in the obvious fashion. Doing so we learn that

$$\begin{aligned}
\left\|\tilde{\mathbf{f}} - \mathbf{z}\right\|_2 &\leq \left\|\mathbf{z}_{\mathcal{S}} - \tilde{\mathbf{f}}_{\mathcal{S}}\right\|_2 + \left\|\tilde{\mathbf{f}}_{\mathcal{S}^{\mathrm{c}}}\right\|_2 \\
&\leq \left(1 + \frac{2\sqrt{1+\epsilon}}{\sqrt{1-\epsilon}}\right)\left\|\tilde{\mathbf{f}}_{\mathcal{S}^{\mathrm{c}}}\right\|_2 + \frac{2\sqrt{1+\epsilon}}{\sqrt{1-\epsilon}} \cdot \frac{\left\|\tilde{\mathbf{f}}_{\mathcal{S}^{\mathrm{c}}}\right\|_1}{\sqrt{s}} + \frac{\delta}{\sqrt{1-\epsilon}}.
\end{aligned}$$

The total runtime complexity follows from the discussion regarding line 2 of Algorithm 1 in Section 3. □

We are now prepared to test a particular version of Algorithm 1 numerically. As we shall see, the experiments demonstrate (as a proof of concept) that SFTs can be used to build stable sublinear-time algorithms capable of rapidly computing Legendre coefficients whenever they exhibit compressibility.

5. Empirical Evaluation

We now present representative results demonstrating the numerical robustness and efficiency of the proposed SFT-based Legendre method. For the experiments in this section we used FFTW 3.3.4 to implement Iserles' scheme [16] for the purposes of comparison (see (16) in §2.2). FFTW3 [9] is a highly efficient implementation of the "standard" FFT algorithm – it has been systematically optimized over the course of the last two decades and remains one of the fastest freely available FFT implementations available today. The parameter r in (12) was chosen to be $1 - 10^{-8}$ for all experiments. The qualitative behavior for other values of r sufficiently close to 1 is similar. The number of terms, M, in (16) was varied differently in each experiment, as indicated below.

Algorithm 1 was implemented using AAFFT [19] as the sparse Fourier transform, followed by a conjugate gradient code[4] in order to compute the necessary Legendre coefficients. Although less optimized than some other SFT implementations, AAFFT's code is well documented, readable, and easy to modify. All code used to perform the experiments below is freely available.[5]

Every data point in the first two figures below is the result of 100 trials performed on 100 different randomly generated polynomials,

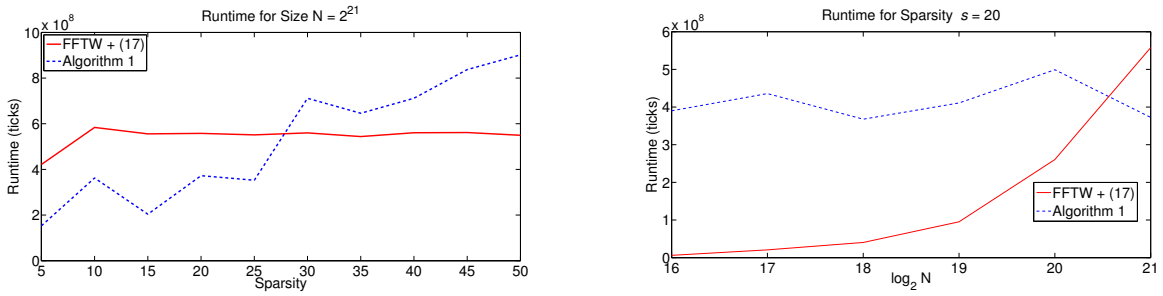
$$f(x) = \sum_{m \in S,\ |S|=s} a_m L_m(x), \tag{38}$$

where $S \subset [N]$ contains s entries independently chosen uniformly at random from $[N]$, and each $a_m \in \{-1, 1\}$ is independently chosen to be ± 1 with probability $1/2$. Figure 1 reports runtime results averaged over the 100 independent trials at each data point for Algorithm 1 versus the

[4]The conjugate gradient code is available at `http://people.sc.fsu.edu/~jburkardt/cpp_src/cg/cg.html`.

[5]All code is available at `http://www.math.msu.edu/~markiwen/Code.html`.

method outlined in §2.2. All runtimes are reported in tick counts using the "cycle.h" header included with the FFTW 3.3.4 library. Tick counts correspond closely to the number of CPU cycles used during the execution of a program and, therefore, accurately reflect each implementation's comparative computational complexity. As one can see, Algorithm 1 is faster than the method outlined in §2.2 when $s \ll N$.[6]



(A) Runtime with $N = 2^{21}$ as sparsity, s, varies.

(B) Runtime with $s = 20$ as N varies.

FIGURE 1. Runtime comparison between Algorithm 1 implemented with AAFFT as the SFT, and the approach from [16] implemented with FFTW as the FFT.

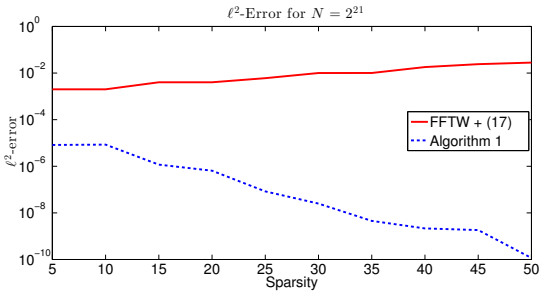
AAFFT is an implementation of a randomized Fourier method with a user-tunable probability of failure (see Theorem 1 for a similar probabilistic SFT recovery guarantee). The runtime results for Algorithm 1 in Figure 1 were produced with settings for AAFFT which guaranteed Algorithm 1 to have an ℓ^2-error of size $< 10^{-5}$ on the s true Legendre coefficients, a_m with $m \in S$, for more than 70 of the 100 trials used to generate each data point. More detailed information concerning approximation errors for these experiments is reported in Figure 2a. Although highly accurate for small values of N (see [16]), the method outlined in §2.2 has a relatively low accuracy for the large degree polynomials considered herein, achieving only two or three digits of accuracy per Legendre coefficient on average. Figure 2a compares its average ℓ^2-error on the true Legendre coefficients, a_m with $m \in S$, over all 100 trials at each data point against Algorithm 1's average ℓ^2-error over the at least 70 trials at each data point for which it correctly identified a superset of S from (38). As one can see, Algorithm 1 is generally more accurate *when it manages to identify* S. The error graphs for the other values of N considered herein were similar. For example, the method outlined in §2.2 had an average ℓ^2-error that was always less than 0.05 for the experiments reported in Figure 1b at each N, while Algorithm 1's average ℓ^2-error was always $< 10^{-5}$ for these experiments whenever it found S (for more than 70% of the trials for each data point).

Figure 3 reports the results of some additional experiments on numerical accuracy, stability, and robustness to noise. For these experiments both the maximum degree, N, and the sparsity, s, of the trial functions were fixed. In addition, the trial functions were modified so that each one was of the form

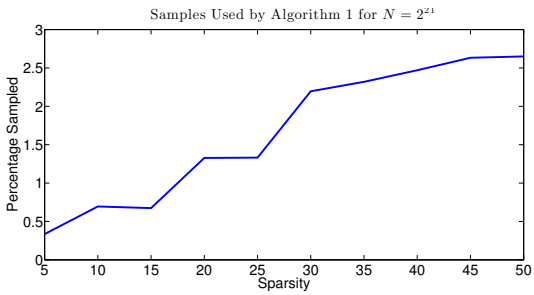
$$f(x) = \sum_{m \in S,\ |S|=s} a_m L_m(x) + \sum_{m \in [N] \setminus S} b_m L_m(x), \tag{39}$$

where (i) $S \subset [N]$ contains s entires independently chosen uniformly at random from $[N]$, (ii) each $a_m \in \{-1, 1\}$ is independently chosen to be ± 1 with probability $1/2$, and (iii) each b_m is an i.i.d mean 0 Gaussian random number generated numerically via the Box-Muller method. As above, each data point in Figure 3 is the result of 100 trials performed on 100 independently generated polynomials of this form (39).

[6]Using a faster SFT implementation would doubtlessly produce faster results for Algorithm 1 – AAFFT is the computational bottleneck here.



(A) Average ℓ^2 error for $N = 2^{21}$ runtime experiments



(B) Algorithm 1 function evaluations for $N = 2^{21}$

FIGURE 2. Additional information regarding the runtime experiments in Figure 1a. The number of terms, M, used in (16) for Figure 2a was chosen separately from the range $[1, 25]$ for each of the 100 trials in order to give the lowest possible error. The best value of M was usually 1, however. Choosing M to be much larger did not usually help. Figure 2b plots the average number of evaluations of each trial polynomial f, as a percentage of N, that Algorithm 1 used in order to produce the results plotted in Figures 2a and 1a. The raw average number of evaluations used by Algorithm 1 for each sparsity (rounded to the nearest whole number) ranges between $7,030$ and $55,575$ in Figure 2b.

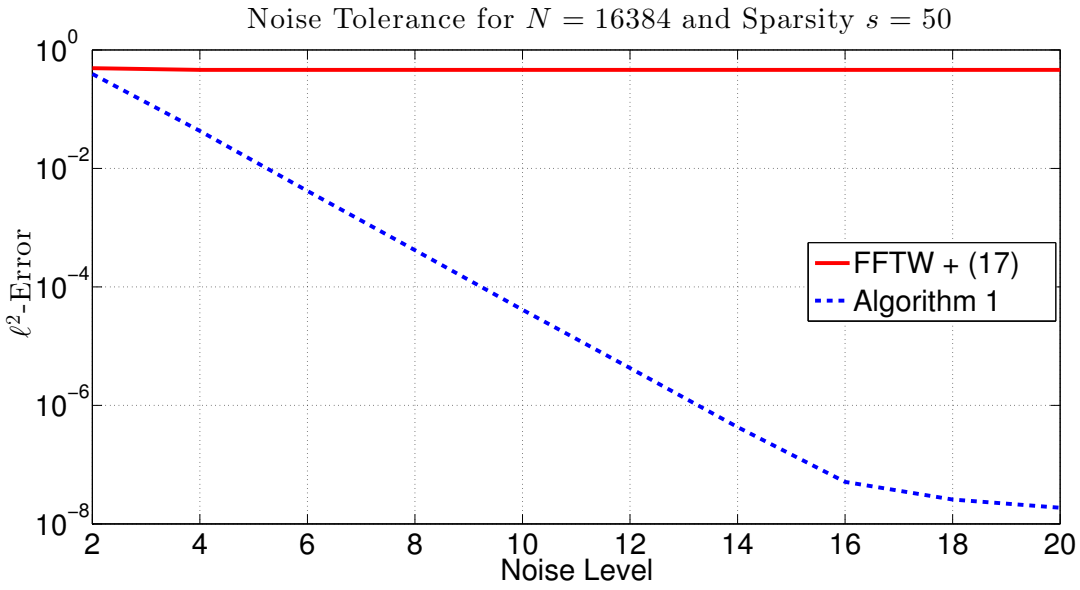


FIGURE 3. Robustness to approximate sparsity. The horizontal axis varies over ten signal-to-noise values given by $\log_{10}\left(\frac{s}{\sum_{m\in[N]\setminus S} b_m^2}\right) = \log_{10}\left(\frac{50}{\sum_{m\in[N]\setminus S} b_m^2}\right)$. The vertical axis graphs the average ℓ^2-error of each method.

For the experiments in Figure 3, Algorithm 1 had its parameters set so that it would compute all Legendre coefficients to at least 8 digits of precision for at least 90% of trials in the noiseless setting (i.e., when (38) holds). The log signal-to-noise ratio,

$$\log_{10}\left(\frac{\sum_{m\in S} a_m^2}{\sum_{m\in[N]\setminus S} b_m^2}\right) = \log_{10}\left(\frac{s}{\sum_{m\in[N]\setminus S} b_m^2}\right), \tag{40}$$

was then varied by renormalizing the i.i.d. b_m's generated for each trial. The resulting log signal-to-noise ratios appear on the horizontal axis of Figure 3. The vertical axis plots the average ℓ^2-error on the true Legendre coefficients, a_m with $m \in S$, over all trials at each noise level of the method in §2.2 against Algorithm 1's average ℓ^2-error over the at least 90 trials at each noise level for which it correctly identified a superset of $S \subset [N]$ from (39). The number of terms, M, used in (16) for the §2.2 method was again chosen separately for each of the 100 trials in order to give the lowest possible error, except that here $M \in [1, 45]$. The best value of M was, as before, usually 1, however. Increasing M still did not usually help decrease the error. The qualitative behavior for other values of s and N was similar. As above, we conclude that Algorithm 1 is generally more accurate than the original §2.2 method provided that it correctly identifies (a superset of) the support set S.

6. Conclusion

In this paper we have demonstrated that SFT techniques can be used to help rapidly approximate functions with sparse Legendre coefficient expansions. Together with the problems already relegated to future work, we believe that it would be interesting to consider extending these methods to functions which exhibit sparsity in other types of Gegenbauer polynomial expansions. Given the level of success in both the Chebyshev and Legendre settings it seems likely that SFTs can be utilized to good effect more generally.

It may also be beneficial to explore the use of slight generalizations of (12) in the Legendre setting, such as $f_{r,k}(x) := \left(1 - r^2 \mathbb{e}^{2\mathbb{i}x}\right)^k f\left(\frac{1}{2}\left(r^{-1}\mathbb{e}^{-\mathbb{i}x} + r\mathbb{e}^{\mathbb{i}x}\right)\right)$ for $r \approx 1$ and $k = 2, 3, 4, \ldots$, in order to improve numerical performance.

Acknowledgements

The authors would like to thank Aditya Viswanathan for many helpful suggestions and discussions during the writing of this paper.

References

[1] A. Akavia, S. Goldwasser, and S. Safra. Proving hard-core predicates using list decoding. In *FOCS*, volume 3, pages 146–156, 2003.

[2] A. Björck. *Numerical methods for least squares problems*. SIAM, 1996.

[3] A. Blum, M. Furst, J. Jackson, M. Kearns, Y. Mansour, and S. Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 253–262. ACM, 1994.

[4] I. Bogaert, B. Michiels, and J. Fostier. $\mathcal{O}(1)$ computation of Legendre polynomials and Gauss - Legendre nodes and weights for parallel computing. *SIAM J. Sci. Comput.*, 34(3):C83–C101, 2012.

[5] J. P. Boyd. *Chebyshev and Fourier spectral methods*. Dover Publications, Inc., 2001.

[6] A. Cohen, R. DeVore, and C. Schwab. Convergence rates of best N-term Galerkin approximations for a class of elliptic sPDEs. *Found. Comput. Math.*, 10(6):615–646, 2010.

[7] A. Cohen, R. Devore, and C. Schwab. Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDEs. *Anal. Appl. (Singap.)*, 9(01):11–47, 2011.

[8] S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Springer, 2013.

[9] M. Frigo and S. Johnson. The design and implementation of FFTW3. *Proceedings of IEEE 93 (2)*, pages 216–231, 2005.

[10] A. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, and M. Strauss. Near-optimal sparse Fourier estimation via sampling. *ACM STOC*, pages 152–161, 2002.

[11] A. Gilbert, P. Indyk, M. Iwen, and L. Schmidt. Recent developments in the sparse Fourier transform: A compressed Fourier transform for big data. *Signal Processing Magazine, IEEE*, 31(5):91–100, 2014.

[12] A. Gilbert, S. Muthukrishnan, and M. Strauss. Improved time bounds for near-optimal sparse Fourier representations. *Proceedings of SPIE Wavelets XI*, 2005.

[13] O. Goldreich. The foundations of modern cryptography. In *Modern Cryptography, Probabilistic Proofs and Pseudorandomness*, pages 1–37. Springer, 1999.

[14] O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 25–32. ACM, 1989.
[15] 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.
[16] A. Iserles. A fast and simple algorithm for the computation of Legendre coefficients. *Numer. Math.*, 117(3):529–553, 2011.
[17] M. A. Iwen. Combinatorial sublinear-time Fourier algorithms. *Found. Comput. Math.*, 10(3):303–338, 2010.
[18] M. A. Iwen. Improved approximation guarantees for sublinear-time Fourier algorithms. *Appl. Comput. Harmon. Anal.*, 34(1):5782, 2013.
[19] M. A. Iwen, A. C. Gilbert, and M. J. Strauss. Empirical evaluation of a sub-linear time sparse DFT algorithm. *Commun. Math. Sci.*, 5(4), 2007.
[20] E. Kushilevitz and Y. Mansour. Learning decision trees using the Fourier spectrum. *SIAM J. Comput.*, 22(6):1331–1348, 1993.
[21] D. Lawlor, Y. Wang, and A. Christlieb. Adaptive sub-linear time Fourier algorithms. *Adv. Adapt. Data Anal.*, 5(01), 2013.
[22] O. Le Maître and O. M. Knio. *Spectral Methods for Uncertainty Quantification*. Springer, 2010.
[23] Y. Mansour. Learning boolean functions via the Fourier transform. In *Theoretical advances in neural computation and learning*, pages 391–424. Springer, 1994.
[24] Y. Mansour. Randomized interpolation and approximation of sparse polynomials. *SIAM J. Comput.*, 24(2):357–368, 1995.
[25] T. Peter and G. Plonka. A generalized Prony method for reconstruction of sparse sums of eigenfunctions of linear operators. *Inverse Problems*, 29(2):025001, 2013.
[26] T. Peter, G. Plonka, and D. Roşca. Representation of sparse Legendre expansions. *J. Symbolic Comput.*, 50:159–169, 2013.
[27] A. D. Polyanin. *Handbook of linear partial differential equations for engineers and scientists*. CRC press, 2001.
[28] D. Potts and M. Tasche. Sparse polynomial interpolation in Chebyshev bases. *Linear Algebra Appl.*, 441:61–87, 2014.
[29] D. Potts and M. Tasche. Reconstruction of sparse Legendre and Gegenbauer expansions. *BIT Numerical Mathematics*, pages 1–25, 2015.
[30] H. Rauhut and R. Ward. Sparse Legendre expansions via ℓ_1-minimization. *J. Approx. Theory*, 164(5):517–533, 2012.
[31] H. Robbins. A remark on Stirlings formula. *Amer. Math. Monthly*, pages 26–29, 1955.
[32] B. Segal and M. Iwen. Improved sparse Fourier approximation results: faster implementations and stronger guarantees. *Numer. Algorithms*, pages 1–25, 2013.
[33] P. Stanica. Good lower and upper bounds on binomial coefficients. *Journal of Inequalities in Pure and Applied Mathematics*, 2(3):30, 2001.
[34] E. C. Titchmarsh. *The theory of functions*, volume 80. London, 1939.
[35] H. Wang and S. Xiang. On the convergence rates of Legendre approximation. *Math. Comp.*, 81(278):861–877, 2012.
[36] S. Zhang and J. Jin. *Computation of special functions*, volume 160. Wiley New York, 1996.

Appendix A. Proof of Lemma 1

For ease of reference we repeat the Lemma to be proven:

Lemma 1. *The even rows of the inverse matrix $\tilde{G}_r^{-1}$ from* (17) *are given by*

$$\left(\tilde{G}_r^{-1}\right)_{2i,2j} = \begin{cases} 0 & j < i \\ \frac{(-1)^j}{4^j} \sum_{k=i}^{j} \frac{(-1)^k}{4^k} \frac{r^{-2i}(2j+2k)!}{(j-k)!(j+k)!(2k)!} \binom{2k}{k+i} \frac{1+2i}{k+i+1} & i \le j \le \left\lfloor \frac{N}{2} \right\rfloor \end{cases}$$

and $\left(\tilde{G}_r^{-1}\right)_{2i,2j+1} = 0$ *for* $i,j = 0,\ldots,\lfloor \frac{N}{2} \rfloor$. *Similarly, the odd rows of the inverse matrix* $\tilde{G}_r^{-1}$ *from* (17) *are given by*

$$\left(\tilde{G}_r^{-1}\right)_{2i+1,2j+1} = \begin{cases} 0 & j < i \\ \frac{(-1)^j}{2\cdot 4^j} \sum_{k=i}^{j} \frac{(-1)^k}{2\cdot 4^k} \frac{r^{-2i-1}(2j+2k+2)!}{(j-k)!(j+k+1)!(2k+1)!} \binom{2k+1}{k-i} \frac{2+2i}{k+i+2} & i \le j < \lfloor \frac{N}{2} \rfloor \end{cases}$$

and $\left(\tilde{G}_r^{-1}\right)_{2i+1,2j} = 0$ *for* $i,j = 0,\ldots,\lfloor \frac{N}{2} \rfloor$.

Proof: Assuming that $f : [-1,1] \to \mathbb{R}$ is a Legendre polynomial of degree N, we can begin to compute $\widehat{f}_r$. This will, in turn, reveal the entries of $\tilde{G}_r^{-1}$ from (17). Recall that

$$\begin{aligned} f_r(x) &= \left(1 - r^2 \mathbb{e}^{2\mathbb{i}x}\right) f\left(\frac{1}{2}\left(r^{-1}\mathbb{e}^{-\mathbb{i}x} + r\mathbb{e}^{\mathbb{i}x}\right)\right) \\ &= \left(1 - r^2 \mathbb{e}^{2\mathbb{i}x}\right) \left[\sum_{j=0}^{\lfloor \frac{N}{2} \rfloor} \tilde{f}(2j) \cdot L_{2j}\left(\frac{r^{-1}\mathbb{e}^{-\mathbb{i}x} + r\mathbb{e}^{\mathbb{i}x}}{2}\right) + \sum_{j=0}^{\lceil \frac{N}{2}-1 \rceil} \tilde{f}(2j+1) \cdot L_{2j+1}\left(\frac{r^{-1}\mathbb{e}^{-\mathbb{i}x} + r\mathbb{e}^{\mathbb{i}x}}{2}\right)\right]. \end{aligned}$$

Setting $z := \frac{r^{-1}\mathbb{e}^{-\mathbb{i}x} + r\mathbb{e}^{\mathbb{i}x}}{2}$ and expanding the Legendre polynomials in terms of the canonical polynomial basis (see, e.g., [36]) we obtain

(41)
$$\begin{aligned} f_r(x) = \left(1 - r^2 \mathbb{e}^{2\mathbb{i}x}\right) \Bigg[&\sum_{j=0}^{\lfloor \frac{N}{2} \rfloor} \tilde{f}(2j) \sum_{k=0}^{j} (-1)^k \frac{(4j-2k)!}{4^j k!(2j-k)!(2j-2k)!} z^{2(j-k)} \\ &+ \sum_{j=0}^{\lceil \frac{N}{2}-1 \rceil} \tilde{f}(2j+1) \sum_{k=0}^{j} (-1)^k \frac{(4j-2k+2)!}{2\cdot 4^j k!(2j-k+1)!(2j-2k+1)!} z^{2(j-k)+1} \Bigg]. \end{aligned}$$

Note that even powers of z will only ever contain even powers of both $r^{-1}\mathbb{e}^{-\mathbb{i}x}$ and $r\mathbb{e}^{\mathbb{i}x}$. Similarly, odd powers of z will only ever contain odd powers of $r^{-1}\mathbb{e}^{-\mathbb{i}x}$ and $r\mathbb{e}^{\mathbb{i}x}$. Thus, it we can see that

$$\widehat{f}_r(\omega) = \begin{cases} \widehat{f}_e(\omega) & \text{if } \omega \equiv 0 \bmod 2 \\ \widehat{f}_o(\omega) & \text{if } \omega \equiv 1 \bmod 2 \end{cases}$$

where

$$f_e(x) = \left(1 - r^2 \mathbb{e}^{2\mathbb{i}x}\right) \left[\sum_{j=0}^{\lfloor \frac{N}{2} \rfloor} \tilde{f}(2j) \sum_{k=0}^{j} (-1)^k \frac{(4j-2k)!}{4^j k!(2j-k)!(2j-2k)!} \left(\frac{r^{-1}\mathbb{e}^{-\mathbb{i}x} + r\mathbb{e}^{\mathbb{i}x}}{2}\right)^{2(j-k)}\right]$$

$$= \left(1 - r^2 \mathbb{e}^{2\mathbb{i}x}\right) \left[\sum_{j=0}^{\lfloor \frac{N}{2} \rfloor} \tilde{f}(2j) \frac{(-1)^j}{4^j} \sum_{k=0}^{j} (-1)^k \frac{(2j+2k)!}{(j-k)!(j+k)!(2k)!} \left(\frac{r^{-1}\mathbb{e}^{-\mathbb{i}x} + r\mathbb{e}^{\mathbb{i}x}}{2}\right)^{2k}\right] \tag{42}$$

and

$$f_o(x) = \left(1 - r^2 \mathbb{e}^{2\mathbb{i}x}\right) \cdot \left[\sum_{j=0}^{\lceil \frac{N}{2}-1 \rceil} \tilde{f}(2j+1) \sum_{k=0}^{j} (-1)^k \frac{(4j-2k+2)!}{2 \cdot 4^j k!(2j-k+1)!(2j-2k+1)!} \left(\frac{r^{-1}\mathbb{e}^{-\mathbb{i}x} + r\mathbb{e}^{\mathbb{i}x}}{2}\right)^{2(j-k)+1}\right]$$

$$= \left(1 - r^2 \mathbb{e}^{2\mathbb{i}x}\right) \cdot \left[\sum_{j=0}^{\lceil \frac{N}{2}-1 \rceil} \tilde{f}(2j+1) \frac{(-1)^j}{2 \cdot 4^j} \sum_{k=0}^{j} (-1)^k \frac{(2j+2k+2)!}{(j-k)!(j+k+1)!(2k+1)!} \left(\frac{r^{-1}\mathbb{e}^{-\mathbb{i}x} + r\mathbb{e}^{\mathbb{i}x}}{2}\right)^{2k+1}\right]. \tag{43}$$

A.1. **The Even Fourier Coefficients.** Expanding $f_e(x)$ from (42) using the Binomial Theorem, reindexing, and then changing the order of summation, we obtain

$$f_e(x) = \left(1 - r^2 \mathbb{e}^{2\mathbb{i}x}\right) \left[\sum_{j=0}^{\lfloor \frac{N}{2} \rfloor} \tilde{f}(2j) \frac{(-1)^j}{4^j} \sum_{k=0}^{j} \frac{(-1)^k}{4^k} \frac{(2j+2k)!}{(j-k)!(j+k)!(2k)!} \sum_{l=-k}^{k} \binom{2k}{k+l} \left(r\mathbb{e}^{\mathbb{i}x}\right)^{2l}\right]$$

$$= \left(1 - r^2 \mathbb{e}^{2\mathbb{i}x}\right) \left[\sum_{l=-\lfloor \frac{N}{2} \rfloor}^{\lfloor \frac{N}{2} \rfloor} \left(r\mathbb{e}^{\mathbb{i}x}\right)^{2l} \sum_{j=|l|}^{\lfloor \frac{N}{2} \rfloor} \tilde{f}(2j) \frac{(-1)^j}{4^j} \sum_{k=|l|}^{j} \frac{(-1)^k}{4^k} \frac{(2j+2k)!}{(j-k)!(j+k)!(2k)!} \binom{2k}{k+l}\right].$$

Multiplying the $\left(1 - r^2 \mathbb{e}^{2\mathbb{i}x}\right)$ factor through the sum above and recombining terms now yields

$$f_e(x) = \left[\left(r\mathbb{e}^{\mathbb{i}x}\right)^{-2\lfloor \frac{N}{2} \rfloor} - \left(r\mathbb{e}^{\mathbb{i}x}\right)^{2\lfloor \frac{N}{2} \rfloor + 2}\right] \binom{4\lfloor \frac{N}{2} \rfloor}{2\lfloor \frac{N}{2} \rfloor} \frac{\tilde{f}\left(2\lfloor N/2 \rfloor\right)}{16^{\lfloor N/2 \rfloor}}$$

$$+ \sum_{l=-\lfloor \frac{N}{2} \rfloor + 1}^{\lfloor \frac{N}{2} \rfloor} \left(r\mathbb{e}^{\mathbb{i}x}\right)^{2l} \left\{\sum_{j=|l|}^{\lfloor \frac{N}{2} \rfloor} \tilde{f}(2j) \frac{(-1)^j}{4^j} \sum_{k=|l|}^{j} \frac{(-1)^k}{4^k} \frac{(2j+2k)!}{(j-k)!(j+k)!(2k)!} \binom{2k}{k+l} \right. \tag{44}$$

$$\left. - \sum_{j=|l-1|}^{\lfloor \frac{N}{2} \rfloor} \tilde{f}(2j) \frac{(-1)^j}{4^j} \sum_{k=|l-1|}^{j} \frac{(-1)^k}{4^k} \frac{(2j+2k)!}{(j-k)!(j+k)!(2k)!} \binom{2k}{k+l} \frac{k+l}{k+1-l}\right\}$$

Recalling from (17) that we are primarily concerned with $l = 0, -1, \ldots, -\lfloor \frac{N}{2} \rfloor$ in the expression above, we can now recombine terms in the bracketed sum to obtain the relevant even Fourier series coefficients of f_r from f_e (please note that, in fact, that the last line of our calculation above does not technically hold as written unless $l \leq 0$!). Doing so, we learn that

$$\widehat{f_r}\,(2l) = r^{2l} \left\{\tilde{f}(2|l|) \frac{1}{16^{|l|}} \binom{4|l|}{2|l|} + \sum_{j=|l|+1}^{\lfloor \frac{N}{2} \rfloor} \tilde{f}(2j) \frac{(-1)^j}{4^j} \left[\frac{(-1)^{|l|}}{4^{|l|}} \frac{(2j+2|l|)!}{(j-|l|)!(j+|l|)!(2|l|)!}\right.\right.$$

$$\left.\left. + \sum_{k=|l|+1}^{j} \frac{(-1)^k}{4^k} \frac{(2j+2k)!}{(j-k)!(j+k)!(2k)!} \binom{2k}{k+l} \frac{1-2l}{k-l+1}\right]\right\} \tag{45}$$

for $l = 0, -1, \ldots, -\lfloor \frac{N}{2} \rfloor + 1$. This combined with (44) gives all entries in the even rows of $\tilde{G}_r^{-1}$.

A.2. **The Odd Fourier Coefficients.** Expanding $f_o(x)$ from (43) using the Binomial Theorem, reindexing, and then changing the order of summation, we obtain

$$\begin{aligned}
f_o(x) &= \left(1 - r^2 \mathbb{e}^{2\mathbb{i}x}\right) \cdot \\
&\quad \left[\sum_{j=0}^{\lceil \frac{N}{2}-1 \rceil} \tilde{f}(2j+1) \frac{(-1)^j}{2 \cdot 4^j} \sum_{k=0}^{j} \frac{(-1)^k}{2 \cdot 4^k} \frac{(2j+2k+2)!}{(j-k)!(j+k+1)!(2k+1)!} \sum_{l=-k}^{k+1} \binom{2k+1}{k+l} \left(r\mathbb{e}^{\mathbb{i}x}\right)^{2l-1} \right] \\
&= \left(1 - r^2 \mathbb{e}^{2\mathbb{i}x}\right) \cdot \left[\sum_{l=0}^{\lceil \frac{N}{2}-1 \rceil} C_l \left(r\mathbb{e}^{\mathbb{i}x}\right)^{2l+1} + \right. \\
&\quad \left. \sum_{l=-\lceil \frac{N}{2}-1 \rceil}^{0} \left(r\mathbb{e}^{\mathbb{i}x}\right)^{2l-1} \sum_{j=|l|}^{\lceil \frac{N}{2}-1 \rceil} \tilde{f}(2j+1) \frac{(-1)^j}{2 \cdot 4^j} \sum_{k=|l|}^{j} \frac{(-1)^k}{2 \cdot 4^k} \frac{(2j+2k+2)!}{(j-k)!(j+k+1)!(2k+1)!} \binom{2k+1}{k+l} \right].
\end{aligned}$$

for some $C_0, \ldots, C_{\lceil \frac{N}{2}-1 \rceil} \in \mathbb{R}$ with which we need not concern ourselves at the moment. Multiplying the $\left(1 - r^2 \mathbb{e}^{2\mathbb{i}x}\right)$ factor through the sum above and recombining terms now yields

$$\begin{aligned}
f_o(x) &= \left(1 - r^2 \mathbb{e}^{2\mathbb{i}x}\right) \sum_{l=0}^{\lceil \frac{N}{2}-1 \rceil} C_l \left(r\mathbb{e}^{\mathbb{i}x}\right)^{2l+1} + \\
&\quad \left(r\mathbb{e}^{\mathbb{i}x}\right)^{-2\lceil \frac{N}{2}-1 \rceil - 1} \binom{4\left\lceil \frac{N}{2} - 1 \right\rceil + 2}{2\left\lceil \frac{N}{2} - 1 \right\rceil + 1} \frac{\tilde{f}\left(2\left\lceil N/2 - 1 \right\rceil + 1\right)}{4 \cdot 16^{\lceil N/2-1 \rfloor}} \\
&\quad + \sum_{l=-\lceil \frac{N}{2}-1 \rceil + 1}^{0} \left(r\mathbb{e}^{\mathbb{i}x}\right)^{2l-1} \left\{ \sum_{j=|l|}^{\lceil \frac{N}{2}-1 \rceil} \tilde{f}(2j+1) \frac{(-1)^j}{2 \cdot 4^j} \sum_{k=|l|}^{j} \frac{(-1)^k}{2 \cdot 4^k} \frac{(2j+2k+2)!}{(j-k)!(j+k+1)!(2k+1)!} \binom{2k+1}{k+l} \right. \\
&\qquad \left. - \sum_{j=|l-1|}^{\lceil \frac{N}{2}-1 \rceil} \tilde{f}(2j+1) \frac{(-1)^j}{2 \cdot 4^j} \sum_{k=|l-1|}^{j} \frac{(-1)^k}{2 \cdot 4^k} \frac{(2j+2k+2)!}{(j-k)!(j+k+1)!(2k+1)!} \binom{2k+1}{k+l} \frac{k+l}{k+2-l} \right\}.
\end{aligned} \tag{46}$$

We can now recombine terms in the bracketed sum to obtain the relevant odd Fourier series coefficients of f_r from f_o. Doing so, we learn that

$$\begin{aligned}
\widehat{f_r}\,(2l-1) &= r^{2l-1} \left\{ \tilde{f}(2|l|+1) \frac{1}{4 \cdot 16^{|l|}} \binom{4|l|+2}{2|l|+1} \right. \\
&\quad + \sum_{j=|l|+1}^{\lceil \frac{N}{2}-1 \rceil} \tilde{f}(2j+1) \frac{(-1)^j}{2 \cdot 4^j} \left[\frac{(-1)^{|l|}}{2 \cdot 4^{|l|}} \frac{(2j+2|l|+2)!}{(j-|l|)!(j+|l|+1)!(2|l|+1)!} \right. \\
&\quad \left. \left. + \sum_{k=|l|+1}^{j} \frac{(-1)^k}{2 \cdot 4^k} \frac{(2j+2k+2)!}{(j-k)!(j+k+1)!(2k+1)!} \binom{2k+1}{k+l} \frac{2-2l}{k-l+2} \right] \right\}
\end{aligned} \tag{47}$$

for $l = 0, -1, \ldots, -\left\lceil \frac{N}{2} - 1 \right\rceil + 1$. This combined with (46) gives all entries in the odd rows of $\tilde{G}_r^{-1}$. □