Appl. Comput. Harmon. Anal. 62 (2023) 310–330

Contents lists available at ScienceDirect

Applied and Computational Harmonic Analysis

journal homepage: www.elsevier.com/locate/acha

Generalization bounds for sparse random feature expansions ☆

Abolfazl Hashemi [a], Hayden Schaeffer [b,*], Robert Shi [c], Ufuk Topcu [c], Giang Tran [d], Rachel Ward [c]

[a] *Purdue University, United States of America*
[b] *Carnegie Mellon University, United States of America*
[c] *The University of Texas at Austin, United States of America*
[d] *University of Waterloo, Canada*

ARTICLE INFO

Article history:
Received 27 August 2021
Received in revised form 11 May 2022
Accepted 11 August 2022
Available online 28 August 2022
Communicated by Rayan Saab

MSC:
60B20
65D15
68Q32
46N10

Keywords:
Random features
Sparse optimization
Generalization error
Compressive sensing

ABSTRACT

Random feature methods have been successful in various machine learning tasks, are easy to compute, and come with theoretical accuracy bounds. They serve as an alternative approach to standard neural networks since they can represent similar function spaces without a costly training phase. However, for accuracy, random feature methods require more measurements than trainable parameters, limiting their use for data-scarce applications. We introduce the sparse random feature expansion to obtain parsimonious random feature models. We leverage ideas from compressive sensing to generate random feature expansions with theoretical guarantees even in the data-scarce setting. We provide generalization bounds for functions in a certain class depending on the number of samples and the distribution of features. By introducing sparse features, i.e. features with random sparse weights, we provide improved bounds for low order functions. We show that our method outperforms shallow networks in several scientific machine learning tasks.

1. Introduction

The *sparsity-of-effects* or *Pareto principle* states that most real-world systems are dominated by a small number of low-complexity interactions. This idea is at the heart of compressive sensing and sparse optimization, which computes a sparse representation for a given dataset using a large set of features. The feature spaces are often constructed using a random matrix, e.g., each element is independent and identi-

☆ Authors are listed in alphabetical order. This work was supported in part by AFOSR MURI FA9550-19-1-0005, AFOSR MURI FA9550-21-1-0084, NSERC Discovery Grant RGPIN 2018-06135, NSF DMS-1752116, and NSF DMS-1952735. The code is available on our github page https://github.com/GiangTTran/SparseRandomFeatures.

* Corresponding author.
E-mail address: hschaeff@andrew.cmu.edu (H. Schaeffer).

https://doi.org/10.1016/j.acha.2022.08.003

cally distributed from the normal distribution, or constructed using a bounded orthonormal system, e.g., Fourier or orthonormal polynomials. While completely random matrices are useful for compression, their lack of structure can limit applications to problems that require physical or meaningful constraints. On the other hand, while bounded orthonormal systems provide meaningful structure to the feature space, they often require knowledge of the sampling measure and the target functions themselves, e.g., that the target function is well-represented by polynomials.

In the high-dimensional setting, neural networks can achieve high test accuracy when there are reasonable models for the local interactions between variables. For example, a convolutional neural network imposes local spatial dependencies between pixels or nodes. In addition, neural networks can construct data-driven feature spaces that far exceed the limitations of pre-specified bases such as polynomials. However, standard neural networks often rely on back-propagation or greedy algorithms to train the weights, which is a computationally intensive procedure. Furthermore, the trained models do not provide interpretable results, i.e., they remain black-boxes. Randomized networks are a class of neural networks that randomize and fix the weights within the architecture [5,36,34,28,31]. When only the final layer is trained, the training problem becomes linear and can have a much lower cost than the non-convex optimization-based approaches. This method has motivated new algorithms and theory, for example, see [36,34,35,43,44,42,9,27]. Recently, generalization bounds for over-parameterized random features ridge regression were provided in [30], when the Tikhonov regularization parameter tends to zero. The analysis is asymptotic and is restricted to the ReLU activation function, with data and features drawn on the sphere.

In this work, we introduce a new framework for approximating high-dimensional functions in the case where measurements are *expensive* and *scarce*. We propose the *sparse random feature expansion* (SRFE), which enhances the compressive sensing approach by allowing for more flexible functional relationships between inputs, as well as a more complex feature space. The choice of basis is inspired by the random Fourier feature (RFF) method [36,34], which uses a basis comprised of simple (often trigonometric) functions with randomized parameters. In the RFF method, the model is learned using ridge regression, which leads to dense (or full) representations. By using sparsity, our approach could be viewed as a way to leverage structure in the data-scarce setting while retaining the accuracy and representation capabilities of the randomized feature methods. In addition, the use of sparsity allows for reasonable generalization bounds even in the very overcomplete setting, which is proving to be a powerful modern tool related to over-parameterized neural networks [23,17,26,3].

In terms of the approximation error, the randomized methods can achieve similar results to those associated with shallow networks. In [24,4], it was shown that if the Fourier transform of the target function f, denoted by $\hat{f}$, has finite integral $\int_{\mathbb{R}^d} |\boldsymbol{\omega}||\hat{f}(\boldsymbol{\omega})|d\boldsymbol{\omega}$ then there is a two-layer neural network with N terms that can approximate f up to an L^2 error of $\mathcal{O}(N^{-\frac{1}{2}})$. These results (and their generalizations) often require specific (greedy) algorithms to achieve. In addition, neural networks often only achieve good performance in the data-rich and over-parameterized regimes. On the other hand, the RFF method achieves uniform errors on the order of $\mathcal{O}(N^{-\frac{1}{2}})$ for functions in a certain class (associated with the choice of the basis functions) without the need for a particular algorithm or construction [34]. Generalization error bounds for random feature ridge regression from [42,57–59] also achieve the rate $\mathcal{O}(N^{-\frac{1}{2}})$, provided the number of data samples grows with N and satisfies certain statistical assumptions. Our generalization bounds for random feature expansions obtained by ℓ_1-minimization match this rate in the general setting without needing a rich training set. Specifically, we show that if the underlying function is a low-order function, admitting a decomposition into a small number of functions each of which depends on only a few variables, then *sparse* random feature expansions can achieve (at worst) generalization bounds of $\mathcal{O}(N^{-\frac{1}{2}+\frac{1}{d}})$ with constants that depend on a polynomial (and not an exponential) of the dimension, in this sense, overcoming the curse of dimensionality.

One of the most popular techniques in the area of uncertainty quantification is the Polynomial Chaos Expansion (PCE). PCE models are built up from univariate orthonormal polynomial regression; in par-

ticular, each basis term is the product of univariate orthonormal polynomials and is characterized by the multi-index of polynomial degrees in each direction. The standard PCE approach solves for the coefficients of the polynomials using the ordinary least squares method. The sparse PCE has recently gained traction, where the coefficient vector is determined through sparse regression. Many sparse regression methods used in PCE were originally developed for compressive sensing [15,7,37,16]. The success of sparse PCE is due in part to the method's ability to incorporate higher degree terms without overfitting. However, the polynomial basis must be orthogonalized with respect to the sampling measure. Moreover, good performance is limited to functions which are well-represented by moderate degree polynomials. This serves as another motivation for the use of randomized features, which may increase the richness of the approximation.

1.1. Contribution

We propose a sparse feature model (the SRFE) which improves on compressive sensing and PCE approaches by utilizing random features from the RFF model. Also, the SRFE outperforms a standard shallow neural network in the limited data regime. We incorporate sparsity in the proposed model in two ways. The first is in our approximation of the target function by using a small number of terms from a large feature space to represent the dominate behavior (this is the sparse expansion component). The second level of sparsity can be considered as side information on the variables and is incorporated by sampling random low order interactions between variables (the sparse features). Building upon these ideas, as part of our theoretical contributions, we derive sample and feature complexity bounds such that the error between the SRFE and the target function is controlled by the richness of the random features, the compressibility of the representation, and the noise on the samples (formalized in Section 3). This also shows the tractability of sparse expansions in the context of randomized feature models.

The SRFE offers additional freedom through redundancy of the basis and does not restrict the model class to low order interactions in the form of polynomials. While our main results are stated for trigonometric features, extensions and applications with ReLU and other standard activation functions can be derived in the same way. In addition, our method and analysis could be extended to include different sampling strategies such as those used in the recovery of dynamical systems [40,41].

In order to provide generalization bounds, we first characterize the approximation power of the best fit approximator; then, we bound the error between the best fit and the sparse random feature expansion. The best fit results are extensions of [36,34], but we provide the proof for completeness. The generalization bounds and the sparse approximation results are both novel. While we utilize standard coherence-based results for sparse recovery, we prove new bounds for the coherence and the sample complexity based on the randomized features (for both dense and sparse features). It is important to note that the bounds are meaningful even when the sparsity increases, which deviates from the standard compressing sensing results. In [45], a sparse random feature algorithm is proposed which iteratively adds random features by using a combination of LASSO and hard thresholding. In our work, we provide sample complexity, sparsity guarantees, and generalization bounds which did not appear in previous works. In addition, we introduce sparse feature weights within our model, which can help with the curse-of-dimensionality for approximating low order functions.

The works of [49,51,50] consider the problem of multi-task learning to learn prediction functions, for T known tasks that lead to the lowest regularized empirical risk. This differs from our algorithmic and theoretical contributions, which focus on the setting of approximating high dimensional low order functions with unknown interactions. In [46–48], the aim is to learn pairwise interactions with linear regression or logistic regression using sparsity-promoting approaches such as LASSO and group-LASSO. As a comparison, we provide generalization bounds which did not appear in previous works. It is also worth noting that our method extends to any algorithm that uses coherence-based sparsity guarantees, for example, greedy

methods such as orthogonal matching pursuit, and the alternative formulation of the basis pursuit or LASSO problem in [52].

A related direction is that of sparse learning-based additive models for kernel regression [56,53,55,54]. In [53], the authors propose the shrunk additive least square approximation (SALSA) method to utilize the interactions among the variables/features, which in some sense, is related to our aim in this paper to leverage the low-order interaction. However, our approach differs from SALSA since we consider sparse feature selection. Furthermore, [53] establish bounds on the expected generalization error while we provide high-probability generalization bounds for our proposed method. Recently, [54] considered SALSA with an ℓ_1 penalty and established high-probability generalization bounds; however, it is limited to kernel regression with exact kernels. In contrast with [54], we provide explicit sparsity guarantees along with the generalization bounds. Moreover, while [53,54] focus on exact kernels, we leverage random features [36,34,35] for efficient function approximation.

2. Approximation via Sparse Random Feature Expansion (SRFE)

Notation. Throughout this paper, we use bold letters and bold capital letters to denote column vectors and matrices, respectively (e.g., $\mathbf{x}$ and $\mathbf{A}$). Let $[N] = \{1, \dots, N\}$ for any positive integer N and $\|\mathbf{c}\|$ denote the Euclidean norm of a vector $\mathbf{c}$. Throughout the paper, f denotes functions of d variables while g denotes functions of $q \ll d$ variables. Furthermore, $\mathbb{B}^d(M)$ denotes the Euclidean ball in $\mathbb{R}^d$ of radius M. A vector $\mathbf{c} \in \mathbb{C}^N$ is said to be s-sparse if the number of nonzero components of $\mathbf{c}$ is at most s. For a vector $\mathbf{c} \in \mathbb{C}^N$, let $\kappa_{s,p}(\mathbf{c})$ denote the error of best s-term approximation to $\mathbf{c}$ in the ℓ_p sense, $\kappa_{s,p}(\mathbf{c}) := \min\{\|\mathbf{c} - \mathbf{z}\|_{\ell_p} : \mathbf{z} \text{ is } s\text{-sparse}\}$ [20]. Note in particular that $\kappa_{s,p}(\mathbf{c}) = 0$ if $\mathbf{c}$ is s-sparse, and $\kappa_{s,p}(\mathbf{c}) \leq \|\mathbf{c}\|_{\ell_p}$ always.

We are interested in identifying an unknown function $f : \mathbb{R}^d \to \mathbb{C}$, belonging to a certain class (defined in Section 3), from a set of samples. We assume that the m sampling points $\mathbf{x}_k$'s are drawn with a probability measure $\mu(\mathbf{x})$ with the corresponding output values

$$y_k = f(\mathbf{x}_k) + e_k, \quad |e_k| \leq E, \quad \forall k \in [m], \tag{1}$$

where e_k is the noise.

A fundamental approach in approximation theory relies on the assumption that f has an approximate linear representation with respect to a suitable collection of N functions $\phi_j(\mathbf{x})$, $j \in [N]$:

$$f(\mathbf{x}) \approx \sum_{j=1}^{N} c_j \phi_j(\mathbf{x}). \tag{2}$$

Important examples of such families of functions include real and complex trigonometric polynomials as well as Legendre polynomials [37,38,1,2,10].

Let $\mathbf{A} \in \mathbb{C}^{m \times N}$ be the random feature matrix with entries $a_{k,j} = \phi_j(\mathbf{x}_k)$, then approximating f in Equation (2) is equivalent to

$$\text{find} \quad \mathbf{c} \in \mathbb{C}^N \quad \text{such that} \quad \mathbf{y} \approx \mathbf{A}\mathbf{c}, \tag{3}$$

where $\mathbf{c} = [c_1, \dots, c_N]^T$ and $\mathbf{y} = [y_1, \dots, y_m]^T$. In many applications, it is often the case that f is well-approximated by a small subset of the N functions, which implies that $\mathbf{c}$ is sparse. By exploiting the sparsity, the number of samples m required to obtain an accurate approximation of f may be significantly reduced. One effective approach to learn a sparse vector $\mathbf{c}$ is to solve the basis pursuit (BP) problem:

$$\mathbf{c}^\sharp = \arg\min_{\mathbf{c}} \quad \|\mathbf{c}\|_1 \qquad \text{s.t.} \quad \|\mathbf{A}\mathbf{c} - \mathbf{y}\| \leq \eta\sqrt{m}, \tag{4}$$

where η is a parameter typically related to the measurement noise. The conditions for stable recovery of any sparse vector $\mathbf{c}^\star$ satisfying $\mathbf{y} \approx \mathbf{A}\mathbf{c}^\star$ are extensively studied in compressed sensing and statistics [8,6,20].

In order to construct a sufficiently rich family of functions, we use a randomized approach. Specifically, consider a collection of functions $\phi(\mathbf{x};\boldsymbol{\omega}) = \phi(\langle \mathbf{x}, \boldsymbol{\omega}\rangle)$ parameterized by a weight vector $\boldsymbol{\omega}$ drawn randomly from a probability distribution $\rho(\boldsymbol{\omega})$. Some popular choices for ϕ are

1. *Random Fourier features:* $\phi(\mathbf{x};\boldsymbol{\omega}) = \exp(i\langle \mathbf{x}, \boldsymbol{\omega}\rangle)$.
2. *Random trigonometric features:* $\phi(\mathbf{x};\boldsymbol{\omega}) = \cos(\langle \mathbf{x}, \boldsymbol{\omega}\rangle)$ and $\phi(\mathbf{x};\boldsymbol{\omega}) = \sin(\langle \mathbf{x}, \boldsymbol{\omega}\rangle)$.
3. *Random ReLU features:* $\phi(\mathbf{x};\boldsymbol{\omega}) = \max(\langle \mathbf{x}, \boldsymbol{\omega}\rangle, 0)$.

Based on [36,34], we call such $\phi(\cdot\,;\boldsymbol{\omega})$ the random features. Altogether, we propose the *Sparse Random Feature Expansion (SRFE)* to approximate f, which is summarized in Algorithm 1.

Algorithm 1 Sparse Random Feature Expansion (SRFE).

1: **Input:** parametric basis function $\phi(\,;\boldsymbol{\omega}) = \phi(\langle \mathbf{x}, \boldsymbol{\omega}\rangle)$, stability parameter η, number of samples m, and number of weights N.
2: Draw m data points $\mathbf{x}_k \sim \mathcal{D}_x$ and observe outputs $y_k = f(\mathbf{x}_k) + e_k$ with $|e_k| \le E$.
3: Draw N random weights $\boldsymbol{\omega}_j \sim \mathcal{D}_\omega$ (independent of the $\mathbf{x}_k$'s).
4: Construct the random feature matrix $\mathbf{A} \in \mathbb{C}^{m\times N}$ such that $a_{kj} = \phi(\mathbf{x}_k;\boldsymbol{\omega}_j)$.
5: Solve

$$\mathbf{c}^\sharp = \arg\min_{\mathbf{c}} \quad \|\mathbf{c}\|_1 \qquad \text{s.t.} \qquad \|\mathbf{A}\mathbf{c} - \mathbf{y}\| \le \eta\sqrt{m}.$$

6: (Optional) Pruning: Set $\mathcal{S}^\sharp$ to be the support set of the s largest (in magnitude) coefficients of $\mathbf{c}^\sharp$ and redefine $\mathbf{c}^\sharp$ to be zero outside of $\mathcal{S}^\sharp$.
7: **Output:** Form the approximation

$$f^\sharp(\mathbf{x}) = \sum_{j=1}^{N} \mathbf{c}_j^\sharp\, \phi(\mathbf{x};\boldsymbol{\omega}_j).$$

3. Low order functions

Often, high dimensional functions that arise from important physical systems are of low order, meaning the function is dominated by a few terms, each depending on only a subset of the input variables, say q out of the d variables where $q \ll d$ [25,14]. Low order functions also appear in other applications as a way to reduce modeling complexity. For example, in dimension reduction and surrogate modeling, sensitivity analysis is employed to determine the most influential input variables and thus to reduce the approximation onto a subset of the input space [39]. The notion of low order functions is also connected to low-dimensional structures [32,33] and active subspaces [19,11,12]. Low order additive functions and sparsely connected networks are also well-motivated in computational neuroscience for simple brain architectures [21].

Next, we formalize the notion of low order functions by extending the definition from [25].

Definition 1 *(Order-q functions).* Fix $d, q, K \in \mathbb{N}$ with $q \le d$. A function $f : \mathbb{R}^d \to \mathbb{C}$ is an order-q function of at most K terms if there exist K functions $g_1, \ldots, g_K : \mathbb{R}^q \to \mathbb{C}$ such that

$$f(x_1, \ldots, x_d) = \frac{1}{K}\sum_{j=1}^{K} g_j(x_{j_1}, \ldots, x_{j_q}) = \frac{1}{K}\sum_{j=1}^{K} g_j(\mathbf{x}|_{\mathcal{S}_j}), \tag{5}$$

where $\mathcal{S}_j = \{j_1, \ldots, j_q\}$ is a subset of the index set $[d]$, $\mathcal{S}_j \ne \mathcal{S}_{j'}$ for $j \ne j'$, and $\mathbf{x}|_{\mathcal{S}_j}$ is the restriction of $\mathbf{x}$ onto $\mathcal{S}_j$.

Note that in general, such a decomposition is not unique. Furthermore, we are interested in the smallest q to refer to the order of a function; trivially, any order-q function $f : \mathbb{R}^d \to \mathbb{C}$ is also order-d.

With this side information, we can further reduce the number of samples needed (see Theorem 3). We modify Algorithm 1 to incorporate the potential coordinate sparsity into the weights $\boldsymbol{\omega}$. Since we do not know the set of active variables, we draw a number of sparse random feature weights on every subset $\mathcal{S} \subset [d]$ of size $|\mathcal{S}| = q$. That is, for each such $\mathcal{S}$, we draw the on-support feature components randomly from the given distribution, and we set the remaining components to be zero. In particular, we have the following definition for our random features.

Definition 2 *(q-Sparse feature weights).* Let $d, q, n \in \mathbb{N}$ with $q \leq d$ and a multivariate probability density $\zeta : \mathbb{R}^q \to \mathbb{R}$. A collection of $N = n\binom{d}{q}$ weight vectors $\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_N$ is said to be a *complete set of q-sparse feature weights (drawn from density ζ)* if they are generated as follows: For each subset $\mathcal{S}_i \subset [d]$ of size $|\mathcal{S}_i| = q$, draw n random vectors $z_1, \ldots, z_n \in \mathbb{R}^q$ from ζ, independent of each other and of all previous draws. Then, use $z_1, \ldots, z_n$ to form q-sparse feature weights $\boldsymbol{\omega}_{i_1}, \ldots, \boldsymbol{\omega}_{i_n} \in \mathbb{R}^d$ by setting $\operatorname{supp}(\boldsymbol{\omega}_{i_k}) = \mathcal{S}$ and $\boldsymbol{\omega}_{i_k}\big|_{\mathcal{S}_i} = z_{i_k}$ for $k \in [n]$ where i_k denotes a reindexing of the weights.

This leads to the Sparse Random Feature Expansion with Sparse Features (*SRFE-S*) by modifying Step (3) of Algorithm 1 to "Draw a complete set of N q-sparse feature weights $\boldsymbol{\omega}_j \in \mathbb{R}^d$ sampled from density $\zeta : \mathbb{R}^q \to \mathbb{R}$". We summarize SRFE-S in Algorithm 2.

Algorithm 2 Sparse Random Feature Expansion with Sparse Feature Weights (SRFE-S).

1: **Input:** parametric basis function $\phi(\mathbf{x}; \boldsymbol{\omega}) = \phi(\langle \mathbf{x}, \boldsymbol{\omega} \rangle)$, feature sparsity level q, probability density $\zeta : \mathbb{R}^q \to \mathbb{R}$, stability parameter η, number of samples m, and number of weights N.
2: Draw m data points $\mathbf{x}_k \sim \mathcal{D}_x$ and observe outputs $y_k = f(\mathbf{x}_k) + e_k$ with $|e_k| \leq E$.
3: Draw a complete set of N q-sparse feature weights $\boldsymbol{\omega}_j \in \mathbb{R}^d$ sampled from density $\zeta : \mathbb{R}^q \to \mathbb{R}$ as defined in Definition 2 (and independent of the $\mathbf{x}_k$'s).
4: Construct a random feature matrix $\mathbf{A} \in \mathbb{C}^{m \times N}$ such that $a_{kj} = \phi(\mathbf{x}_k; \boldsymbol{\omega}_j)$.
5: Solve

$$\mathbf{c}^\sharp = \arg\min_{\mathbf{c}} \quad \|\mathbf{c}\|_1 \qquad \text{s.t.} \quad \|\mathbf{A}\mathbf{c} - \mathbf{y}\| \leq \eta\sqrt{m}.$$

6: (Optional) Pruning: Set $\mathcal{S}^\sharp$ to be the support set of the s largest (in magnitude) coefficients of $\mathbf{c}^\sharp$ and redefine $\mathbf{c}^\sharp$ to be zero outside of $\mathcal{S}^\sharp$.
7: **Output:** Form the approximation

$$f^\sharp(\mathbf{x}) = \sum_{j=1}^{N} \mathbf{c}_j^\sharp \, \phi(\mathbf{x}; \boldsymbol{\omega}_j).$$

Remark 1. Drawing a complete set of q-sparse feature weights can be slow and cumbersome. In the case where $\zeta(x_1, \ldots, x_q) = \prod_{j=1}^{q} \zeta(x_j)$ is a tensor product of univariate densities, a significantly more practical method for drawing sparse features is as follows: we randomly generate a size q subset of $[d]$ and then define the on-support values using ζ. Alternatively, one can draw sparse feature weights by the following procedure: for every $k \in [N]$, the j-th entry of $\boldsymbol{\omega}_k \in \mathbb{R}^d$, $\boldsymbol{\omega}_{k,j}$, is set to 0 with probability $\left(1 - \frac{q}{d}\right)$ and is drawn from ζ, $\boldsymbol{\omega}_{k,j} \sim \zeta$, with probability $\frac{q}{d}$. This procedure is used in the experiments found in Section 5. We further note that any side-information on the feasibility of the low order support subsets can be incorporated in the procedure outlined in Algorithm 2 to further reduce the required number of sparse features.

4. Theoretical analysis

In this section, we provide theoretical performance guarantees on the approximation given by Algorithm 1 and Algorithm 2. In particular, we derive an explicit bound on the required number of data samples for a stable approximation within a target region. Given the connections to Fourier analysis and its desirable

characteristics, we mainly focus on the case where $\phi(\mathbf{x};\boldsymbol{\omega}) = \exp(i\langle \mathbf{x},\boldsymbol{\omega}\rangle)$. The results in this section extend to other distributions and basis functions, in particular, one can show similar results for uniform or subgaussian distributions (with a change in the constants that appear in the theorems).

Before stating the main results, we recall some useful definitions. The first definition is a complex-valued extension of the class introduced in [34].

Definition 3 *(Bounded ρ-norm functions).* Fix a probability density function $\rho : \mathbb{R}^d \to \mathbb{R}$ and a function $\phi : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{C}$. A function $f : \mathbb{R}^d \to \mathbb{C}$ has finite ρ-norm with respect to $\phi(\mathbf{x};\boldsymbol{\omega})$ if it belongs to the class

$$\mathcal{F}(\phi,\rho) := \left\{ f(\mathbf{x}) = \int_{\boldsymbol{\omega}\in\mathbb{R}^d} \alpha(\boldsymbol{\omega})\phi(\mathbf{x};\boldsymbol{\omega})\, d\boldsymbol{\omega} \;:\; \|f\|_\rho := \sup_{\boldsymbol{\omega}} \left|\frac{\alpha(\boldsymbol{\omega})}{\rho(\boldsymbol{\omega})}\right| < \infty \right\}. \tag{6}$$

Note that in the above definition, if $\phi(\mathbf{x};\boldsymbol{\omega}) = \exp(i\langle \mathbf{x},\boldsymbol{\omega}\rangle)$, $\alpha : \mathbb{R}^d \to \mathbb{C}$ is the inverse Fourier transform of f.

4.1. Generalization error

We state our main results here. Recall that $\mu(\mathbf{x})$ denotes the probability measure for sampling $\mathbf{x}$.

Theorem 1 *(Generalization bound for bounded ρ-norm functions). Let $f \in \mathcal{F}(\phi,\rho)$, where $\phi(\mathbf{x};\boldsymbol{\omega}) = \phi(\langle \mathbf{x},\boldsymbol{\omega}\rangle) = \exp(i\langle \mathbf{x},\boldsymbol{\omega}\rangle)$ and $\rho(\boldsymbol{\omega})$ is the density corresponding to a spherical Gaussian with variance σ^2, $\mathcal{N}(\mathbf{0},\sigma^2\mathbf{I}_d)$. For a fixed γ, consider a set of data samples $\mathbf{x}_1,\dots,\mathbf{x}_m \sim \mathcal{N}(\mathbf{0},\gamma^2\mathbf{I}_d)$ and frequencies $\boldsymbol{\omega}_1,\dots,\boldsymbol{\omega}_N \sim \mathcal{N}(\mathbf{0},\sigma^2\mathbf{I}_d)$. The measurement noise e_k is either bounded by $E = 2\nu$ or to be drawn i.i.d. from $\mathcal{N}(0,\nu^2)$. Let $\mathbf{A} \in \mathbb{C}^{m\times N}$ denote the associated random feature matrix where $a_{k,j} = \phi(\mathbf{x}_k;\boldsymbol{\omega}_j)$. Let $f^\sharp$ be defined from Algorithm 1 and Equation (4) with $\eta = \sqrt{2(\epsilon^2\|f\|_\rho^2 + E^2)}$ and with the additional pruning step*

$$f^\sharp(\mathbf{x}) := \sum_{j\in\mathcal{S}^\sharp} \mathbf{c}_j^\sharp\, \phi(\mathbf{x};\boldsymbol{\omega}_j).$$

where $\mathcal{S}^\sharp$ is the support set of the s largest (in magnitude) coefficients of $\mathbf{c}^\sharp$.

For a given s, if the feature parameters σ and N, the confidence δ, and the accuracy ϵ are chosen so that the following conditions hold:

1. *γ-σ uncertainty*

$$\gamma^2\sigma^2 \geq \frac{1}{2}(13s)^{\frac{2}{d}}, \tag{7}$$

2. *Number of features*

$$N = \frac{4}{\epsilon^2}\left(1 + 4\gamma\sigma d\sqrt{1+\sqrt{\frac{12}{d}\log\frac{m}{\delta}}} + \sqrt{\frac{1}{2}\log\left(\frac{1}{\delta}\right)}\right)^2, \tag{8}$$

3. *Number of measurements*

$$m \geq 4(2\gamma^2\sigma^2+1)^d \log\frac{N^2}{\delta}, \tag{9}$$

Dimensionality

$$d \geq \frac{4 \log\left(\frac{N^2}{\delta}\right)}{\log\left(\frac{\gamma^2\sigma^2}{e\log(2\gamma^2\sigma^2+1)}\right)}. \tag{10}$$

Then, with probability at least $1-6\delta$ *the following error bound holds*

$$\begin{aligned}\sqrt{\int_{\mathbb{R}^d} |f(\mathbf{x}) - f^\sharp(\mathbf{x})|^2 \,\mathrm{d}\mu} \leq C'\left(1 + N^{\frac{1}{2}} s^{-\frac{1}{2}} m^{-\frac{1}{4}} \log^{1/4}\left(\frac{1}{\delta}\right)\right) \kappa_{s,1}(\mathbf{c}^\star) \\ + C\left(1 + N^{\frac{1}{2}} m^{-\frac{1}{4}} \log^{1/4}\left(\frac{1}{\delta}\right)\right) \sqrt{\epsilon^2 \|f\|_\rho^2 + 4\nu^2},\end{aligned} \tag{11}$$

where $C, C' > 0$ *are constants and* $\mathbf{c}^\star$ *is the vector*

$$\mathbf{c}^\star = \frac{1}{N}\left[\frac{\alpha(\boldsymbol{\omega}_1)}{\rho(\boldsymbol{\omega}_1)}, \cdots, \frac{\alpha(\boldsymbol{\omega}_N)}{\rho(\boldsymbol{\omega}_N)}\right]^T. \tag{12}$$

The constants in Theorem 1 are chosen for simplicity, where more precise bounds can be found in the Appendix. For example, expression (7) is a simplification of (B.33). In Theorem 1, we see that the γ-σ uncertainty principle becomes less severe in the high-dimensional setting.

Remark 2. Although the bounds include a factor of $N^{\frac{1}{2}}$, the error decreases with N in many settings. For example, let's consider the noise-free case $E = 0$ and set $s = N$ i.e. the upper bound for the sparsity. The first term becomes zero and the remaining term simplifies to

$$\begin{aligned}&\sqrt{\int_{\mathbb{R}^d} |f(\mathbf{x}) - f^\#(\mathbf{x})|^2 \,\mathrm{d}\mu} \\ &\quad\leq C\left(N^{-\frac{1}{2}} + m^{-\frac{1}{4}} \log^{1/4}\left(\frac{1}{\delta}\right)\right)\left(1 + 4\gamma\sigma d\sqrt{1 + \sqrt{\frac{12}{d}\log\frac{m}{\delta}}} + \sqrt{\frac{1}{2}\log\left(\frac{1}{\delta}\right)}\right)\|f\|_\rho, \\ &\quad\leq \tilde{C}\, N^{-\frac{1}{2}}\left(1 + \frac{\log^{1/4}\left(\frac{1}{\delta}\right)}{\log^{1/4}\left(\frac{N^2}{\delta}\right)}\right)\left(1 + 4\gamma\sigma d\sqrt{1 + \sqrt{\frac{12}{d}\log\frac{m}{\delta}}} + \sqrt{\frac{1}{2}\log\left(\frac{1}{\delta}\right)}\right)\|f\|_\rho.\end{aligned} \tag{13}$$

where we used the complexity bounds on m:

$$m = \mathcal{O}\left(N^2 \log\frac{N^2}{\delta}\right).$$

Therefore, up to log terms, our generalization bound is $\mathcal{O}(\gamma\sigma N^{-\frac{1}{2}}) = \mathcal{O}(N^{-\frac{1}{2}+\frac{1}{d}})$.

Interestingly, we observe the appearance of a Heisenberg-type uncertainty principle between "frequency-domain" and "space-domain" variances, σ^2 and γ^2 in Theorem 1 [22]. In Theorem 1, the product of the variances is bounded below by an $\mathcal{O}(s^{\frac{2}{d}})$ term.

Next, we state an informal version of Theorem 1 by simplifying the conditions in the theorem to the leading orders, i.e. ignoring the constants and slower log terms. In particular, each condition in Theorem 2 depends on the known or controllable parameters: the dimension d, the sparsity s, the accuracy ϵ, and the confidence δ.

Theorem 2 *(Informal statement: generalization bound for bounded ρ-norm functions). Given the setup from Theorem 1 but with $E = 0$, for a given s, if the feature parameters σ and N, the confidence δ, and the accuracy ϵ are chosen so that the following conditions hold:*

1. *γ-σ uncertainty*

$$\gamma^2\sigma^2 = \mathcal{O}\left(s^{\frac{2}{d}}\right), \tag{14}$$

2. *Number of features*

$$N = \mathcal{O}\left(\epsilon^{-2}s^{\frac{2}{d}}d^2\right), \tag{15}$$

3. *Number of measurements*

$$m = \mathcal{O}\left(s^2\log\frac{N^2}{\delta}\right). \tag{16}$$

4. *Dimensionality*

$$d = \mathcal{O}\left(\log\left(\frac{N^2}{\delta}\right)\right) \tag{17}$$

Then, with probability at least $1 - \mathcal{O}(\delta)$ the following noise-free error bound holds

$$\sqrt{\int_{\mathbb{R}^d} |f(\mathbf{x}) - f^\sharp(\mathbf{x})|^2 \,\mathrm{d}\mu} = \mathcal{O}\left(\left(1 + N^{\frac{1}{2}}s^{-1}\right)\,\kappa_{s,1}(\mathbf{c}^\star) + N^{-\frac{1}{2}}s^{\frac{1}{d}} + m^{-\frac{1}{4}}s^{\frac{1}{d}}\right), \tag{18}$$

in terms of the scaling with N, m, and s only, where $\mathbf{c}^\star$ is the vector

$$\mathbf{c}^\star = \frac{1}{N}\left[\frac{\alpha(\boldsymbol{\omega}_1)}{\rho(\boldsymbol{\omega}_1)}, \cdots, \frac{\alpha(\boldsymbol{\omega}_N)}{\rho(\boldsymbol{\omega}_N)}\right]^T. \tag{19}$$

Note that the dimensionality condition (10) simplifies to (17) as follows. Using the γ-σ uncertainty principle, we pick a scaling $c > 1$ (following the big-O notation) such that

$$d \simeq \frac{4\log\left(\frac{N^2}{\delta}\right)}{\log\left(\frac{1}{2e}\gamma^2\sigma^2\right)} \simeq \frac{4\log\left(\frac{N^2}{\delta}\right)}{\log\left(cs^{\frac{2}{d}}\right)}$$

which implies

$$d \simeq \frac{4d\log\left(\frac{N^2}{\delta}\right)}{d\log(c) + 2\log(s)} \quad \text{or} \quad d = \mathcal{O}\left(\log\left(\frac{N^2}{\delta}\right)\right).$$

Remark 3. Consider a function $f \in \mathcal{F}(\phi, \rho)$ whose Fourier transform is supported within a compact set $\Omega \subset \mathbb{R}^d$ such that $\int_\Omega \rho(\boldsymbol{\omega})\,d\boldsymbol{\omega} =: \beta < 1$. Then the vector $\mathbf{c}^\star$ will be sparse with high probability, as its expected sparsity scales like $s = \beta N$. Thus, functions with compactly clustered spectral energy are well-approximated by the SRFE method.

Theorem 1 shows that the generalization bound consists of several terms. The first term depends on the quality of the best s-term approximation of f with respect to the random feature basis. Since $\kappa_{s,1}(\mathbf{c}^\star)$ is bounded by $\frac{N-s}{N}\|f\|_\rho$, the first error term is related to the complexity of the function class. Part of the second term is controlled by the strength of the random features in representing f. By decreasing ϵ, thereby increasing N, we can increase the power of our representation and thus reduce this error term. The other component of the second term is proportional to the level of noise on the samples and, in general, cannot be reduced arbitrarily. However, in the high-noise case, the bound shows that taking larger m will improve the error bounds with respect to the noise.

When more information is known about the target function, the rates and complexity bounds improve (especially with respect to the dimension). This helps mitigate issues with the approximation of functions in high-dimensions. This result is detailed below.

Theorem 3 *(Generalization bounds for order-q functions). Let f be an order-q function of at most K terms as defined in Definition 1, such that each term g_ℓ, $\ell = 1, 2, \dots, K$, belongs to $\mathcal{F}(\phi, \rho)$ with $\phi(\mathbf{x}; \boldsymbol{\omega}) = \phi(\langle \mathbf{x}, \boldsymbol{\omega}\rangle) = \exp(i\langle \mathbf{x}, \boldsymbol{\omega}\rangle)$, and $\rho : \mathbb{R}^q \to \mathbb{R}$ the density for a spherical Gaussian with variance σ^2, $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}_q)$. Let $\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_N$ be a complete set of q-sparse feature weights drawn from density ρ. Fix γ and draw i.i.d. sampling points $\mathbf{x}_1, \dots, \mathbf{x}_m \sim \mathcal{N}(\mathbf{0}, \gamma^2\mathbf{I}_d)$. The measurement noise e_k is either bounded by $E = 2\nu$ or to be drawn i.i.d. from $\mathcal{N}(0, \nu^2)$. Let $\mathbf{A} \in \mathbb{C}^{m\times N}$ denote the associated random feature matrix where $a_{k,j} = \phi(\langle \mathbf{x}_k, \boldsymbol{\omega}_j\rangle)$ and $f^\sharp$ be defined from Algorithm 2 and Equation (4) with the additional pruning step and with $\eta = \sqrt{2\epsilon^2\binom{d}{q}|||f|||^2 + 2E^2}$, where $|||f||| = \frac{1}{K}\sum_{j=1}^{K}\|g_j\|_\rho$.*

For a given s, suppose the feature parameters σ and N, the confidence δ, and the accuracy ϵ are chosen so that the following conditions hold:

1. *γ-σ uncertainty principle*

$$\gamma^2\sigma^2 \geq \frac{1}{2}(13s)^{\frac{2}{q}}, \tag{20}$$

2. *Number of features*

$$N = n\binom{d}{q} = \frac{4}{\epsilon^2}\left(1 + 4\gamma\sigma d\sqrt{1 + \sqrt{\frac{12}{d}\log\frac{m}{\delta}}} + \sqrt{\frac{q}{2}\log\left(\frac{d}{\delta}\right)}\right)^2, \tag{21}$$

3. *Number of measurements*

$$m \geq 4(2\gamma^2\sigma^2 + 1)^{\max\{2q-d,0\}}(\gamma^2\sigma^2 + 1)^{\min\{2q,2d-2q\}}\log\frac{N^2}{\delta}, \tag{22}$$

4. *Dimensionality*

$$q \geq \frac{4\log\left(\frac{N^2}{\delta}\right)}{\log\left(\frac{\gamma^2\sigma^2}{e\log(2\gamma^2\sigma^2+1)}\right)} \tag{23}$$

Then, with probability at least $1-6\delta$ *the following error bound holds*

$$\begin{aligned}\sqrt{\int_{\mathbb{R}^d} |f(\mathbf{x}) - f^\sharp(\mathbf{x})|^2 \,\mathrm{d}\mu} \leq C' \left(1 + N^{\frac{1}{2}} s^{-\frac{1}{2}} m^{-\frac{1}{4}} \log^{1/4}\left(\frac{1}{\delta}\right)\right) \kappa_{s,1}(\tilde{\mathbf{c}}^\star) \\ + C\left(1 + N^{\frac{1}{2}} m^{-\frac{1}{4}} \log^{1/4}\left(\frac{1}{\delta}\right)\right) \sqrt{\epsilon^2 \binom{d}{q} |\!|\!| f |\!|\!|^2 + E^2},\end{aligned} \tag{24}$$

where $C, C' > 0$ *are constants and the vector* $\tilde{\mathbf{c}}^* = [\tilde{\mathbf{c}}_1^*, \ldots, \tilde{\mathbf{c}}_N^*]^T \in \mathbb{C}^N$ *is defined as follows*

$$\tilde{\mathbf{c}}_j^\star := \frac{1}{K} \sum_{\ell=1}^{K} \tilde{c}_{\ell,j}^\star, \quad \text{with} \quad \tilde{c}_{\ell,j}^\star = \begin{cases} \frac{\alpha_\ell(\boldsymbol{\omega}_j)}{n\,\rho(\boldsymbol{\omega}_j)}, & \text{if } \operatorname{supp}(\boldsymbol{\omega}_j) = \mathcal{S}_\ell \\ 0, & \text{otherwise.} \end{cases} \tag{25}$$

The function $\alpha_\ell(\boldsymbol{\omega})$ *is the transform of* g_ℓ *using Definition 3 and Definition 1.*

We state an informal version of Theorem 3 by simplifying the conditions in the theorem to the leading order terms analogous to Theorem 2.

Theorem 4 *(Informal statement: generalization bounds for order-q functions). Given the setup from Theorem 3 but with* $E = 0$ *and* $q < \frac{d}{2}$*, for a given* s*, suppose the feature parameters* σ *and* N*, the confidence* δ*, and the accuracy* ϵ *are chosen so that the following conditions hold:*

1. *γ-σ uncertainty principle*

$$\gamma^2 \sigma^2 = \mathcal{O}\left(s^{\frac{2}{q}}\right) \tag{26}$$

2. *Number of features*

$$N = n\binom{d}{q} = \mathcal{O}\left(\epsilon^{-2} s^{\frac{2}{q}} d^2\right), \tag{27}$$

3. *Number of measurements*

$$m = \mathcal{O}\left(s^4 \log \frac{N^2}{\delta}\right), \tag{28}$$

4. *Dimensionality*

$$q = \mathcal{O}\left(\log\left(\frac{N^2}{\delta}\right)\right) \tag{29}$$

Then, with probability at least $1 - \mathcal{O}(\delta)$ *the following error bound holds*

$$\sqrt{\int_{\mathbb{R}^d} |f(\mathbf{x}) - f^\sharp(\mathbf{x})|^2 \,\mathrm{d}\mu} = \mathcal{O}\left(\left(1 + N^{\frac{1}{2}} s^{-\frac{3}{2}}\right) \kappa_{s,1}(\mathbf{c}^\star) + N^{-\frac{1}{2}} s^{\frac{1}{q}} + m^{-\frac{1}{4}} s^{\frac{1}{q}}\right), \tag{30}$$

in terms of the scaling with N*,* m*, and* s *only, where the vector* $\tilde{\mathbf{c}}^* = [\tilde{\mathbf{c}}_1^*, \ldots, \tilde{\mathbf{c}}_N^*]^T \in \mathbb{C}^N$ *is defined as follows*

$$\tilde{\mathbf{c}}_j^\star := \frac{1}{K}\sum_{\ell=1}^{K} \tilde{c}^\star_{\ell,j}, \quad \textit{with} \quad \tilde{c}^\star_{\ell,j} = \begin{cases} \frac{\alpha_\ell(\boldsymbol{\omega}_j)}{n\,\rho(\boldsymbol{\omega}_j)}, & \textit{if } \operatorname{supp}(\boldsymbol{\omega}_j) = \mathcal{S}_\ell \\ 0, & \textit{otherwise.} \end{cases} \tag{31}$$

The function $\alpha_\ell(\boldsymbol{\omega})$ is the transform of g_ℓ using Definition 3 and Definition 1.

Remark 4. From the proof, the bound for N is

$$N = \frac{4}{\epsilon^2}\left(1 + 4\gamma\sigma d\sqrt{1+\sqrt{\frac{12}{d}\log\frac{m}{\delta}}} + \sqrt{\frac{1}{2}\log\left(\frac{K}{\delta}\right)}\right)^2$$

and we obtain Equation (21) by noting that $K \le \binom{d}{q} \le \left(\frac{ed}{q}\right)^q$ (and redefining ϵ).

Remark 5. Note that in the bound for the number of measurements, the term $(\gamma^2\sigma^2+1)^2$ is in the range

$$2\gamma^2\sigma^2 + 1 \le (\gamma^2\sigma^2+1)^2 \le (2\gamma^2\sigma^2+1)^2$$

and thus, if we choose the variances so that uncertainty principle holds with equality, which from the proof is $(2\gamma^2\sigma^2+1)^{\frac{q}{2}} = \mathcal{O}(s)$, then we see that m scales between s^4 for $q \le \frac{d}{2}$ and s^2 for $q = d$.

4.2. Discussion on low-order functions

For low-order functions, Theorems 3 and 4 indicate a significant reduction in terms of the dimension d. In particular, for small q, the term $\binom{d}{q}|||f|||$ (which includes a dimensional scale of $\binom{d}{q}^{\frac{1}{2}}$) should grow slower than the norm $\|f\|_{\rho'}$ where ρ' is the probability density in the ambient space of dimension d (assuming all terms exist). For a simple example, let f be an order-q function with $K = 1$ and let $\alpha(\boldsymbol{\omega}) = 1$ be compactly supported on the square defined by $(\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_q) \in [-1,1]^q$. If we applied Algorithm 1 in the ambient dimension d with ρ' defined as the uniform probability distribution over the square in dimension d, then $\|f\|_{\rho'} = 2^d$. Using sparse features with ρ defined as the uniform probability distribution over the square in dimension q, we have $|||f||| \le \binom{d}{q}^{\frac{1}{2}} 2^q \le d^{\frac{q}{2}} 2^q$. For small q relative to d, we see that $|||f|||$ will grow slower than $\|f\|_{\rho'}$ with respect to d (in this example). This indicates one of the potential theoretical benefits; however, a full characterization is difficult to quantify because the theoretical dependency of the number of measurements m scales like s^4 in the low-order setting while it scales like s^2 in the standard setting. A more complete picture will be investigated in future work.

4.3. Proof of Theorem 1

In this section, we discuss our main technical arguments, which lead to Theorem 1. Note that the generalization error can be written as

$$\sqrt{\int_{\mathbb{R}^d} |f(\mathbf{x}) - f^\sharp(\mathbf{x})|^2 \,\mathrm{d}\mu} \le \sqrt{\int_{\mathbb{R}^d} |f(\mathbf{x}) - f^\star(\mathbf{x})|^2 \,\mathrm{d}\mu} + \sqrt{\int_{\mathbb{R}^d} |f^\star(\mathbf{x}) - f^\sharp(\mathbf{x})|^2 \,\mathrm{d}\mu}, \tag{32}$$

where

$$f^\star(\mathbf{x}) = \sum_{j=1}^{N} c_j^\star \exp(i\langle \mathbf{x}, \boldsymbol{\omega}_j\rangle), \quad c_j^\star := \frac{\alpha(\boldsymbol{\omega}_j)}{N\rho(\boldsymbol{\omega}_j)}. \tag{33}$$

We then aim to study these two sources of error in the following lemmata.

4.3.1. Bounding the first error term

We first extend an argument from [34,35] to derive a bound on how well a function in $\mathcal{F}(\phi,\rho)$ can be approximated by SRFE and characterize the approximation power of $f^\star$, the best ϕ-based approximation to f.

Lemma 1 *(Generalization error, Term 1). Fix the confidence parameter $\delta > 0$ and accuracy parameter $\epsilon > 0$. Recall the setting of Algorithm 1 and suppose $f \in \mathcal{F}(\phi,\rho)$ where $\phi(\mathbf{x};\boldsymbol{\omega}) = \exp(i\langle \mathbf{x},\boldsymbol{\omega}\rangle)$. The data samples $\mathbf{x}_k$ have probability measure $\mu(\mathbf{x})$ and weights $\boldsymbol{\omega}_j$ are sampled using the probability density $\rho(\boldsymbol{\omega})$. Consider the random feature approximation*

$$f^\star(\mathbf{x}) := \sum_{j=1}^{N} c_j^\star \exp(i\langle \mathbf{x},\boldsymbol{\omega}_j\rangle), \quad \text{where} \quad c_j^\star := \frac{\alpha(\boldsymbol{\omega}_j)}{N\rho(\boldsymbol{\omega}_j)}. \tag{34}$$

If the number of features N satisfies the bound

$$N \geq \frac{1}{\epsilon^2}\left(1+\sqrt{2\log\left(\frac{1}{\delta}\right)}\right)^2, \tag{35}$$

then, with probability at least $1-\delta$ with respect to the draw of the weights $\boldsymbol{\omega}_j$ the following holds

$$\sqrt{\int_{\mathbb{R}^d} |f(\mathbf{x}) - f^\star(\mathbf{x})|^2 \,\mathrm{d}\mu} \leq \epsilon \|f\|_\rho. \tag{36}$$

The proof of Lemma 1 is similar to the result of [35]. The result in Lemma 1 is not constructive since $\mathbf{c}^\star$ depends on the unknown function $\alpha(\boldsymbol{\omega})$. Nonetheless, Lemma 1 establishes a useful bound on the first source of error in (32).

4.3.2. Bounding the second error term

The next lemma controls the second source of error.

Lemma 2 *(Generalization error, Term 2). Let $f \in \mathcal{F}(\phi,\rho)$, where the basis function is $\phi(\mathbf{x};\boldsymbol{\omega}) = \exp(i\langle \mathbf{x},\boldsymbol{\omega}\rangle)$. For a fixed γ and q, consider a set of data samples $\mathbf{x}_1,\dots,\mathbf{x}_m \sim \mathcal{N}(\mathbf{0},\gamma^2\mathbf{I}_d)$ with $\mu(\mathbf{x})$ denoting the associated probability measure and weights $\boldsymbol{\omega}_1,\dots,\boldsymbol{\omega}_N$ drawn from $\mathcal{N}(\mathbf{0},\sigma^2\mathbf{I}_d)$. Assume that the noise is bounded by $E = 2\nu$ or that the noise terms e_j are drawn i.i.d. from $\mathcal{N}(0,\nu^2)$. Let $\mathbf{A} \in \mathbb{C}^{m\times N}$ denote the associated random feature matrix where $a_{k,j} = \phi(\mathbf{x}_k;\boldsymbol{\omega}_j)$. Let $f^\sharp$ be defined from Algorithm 1 and Equation (4) with $\eta = \sqrt{2(\epsilon^2\|f\|_\rho^2 + E^2)}$ and with the additional pruning step*

$$f^\sharp(\mathbf{x}) := \sum_{j\in\mathcal{S}^\sharp} \mathbf{c}_j^\sharp \,\phi(\mathbf{x};\boldsymbol{\omega}_j),$$

where $\mathcal{S}^\sharp$ is the support set of the s largest (in magnitude) coefficients of $\mathbf{c}^\sharp$. Let the random feature approximation $f^\star$ be defined as

$$f^\star(\mathbf{x}) := \sum_{j=1}^{N} \mathbf{c}_j^\star \exp(i\langle \mathbf{x},\boldsymbol{\omega}_j\rangle), \tag{37}$$

where

$$\mathbf{c}^\star = \left[\frac{\alpha(\boldsymbol{\omega}_1)}{N\,\rho(\boldsymbol{\omega}_1)}, \cdots, \frac{\alpha(\boldsymbol{\omega}_N)}{N\,\rho(\boldsymbol{\omega}_N)}\right]^T. \tag{38}$$

For a given s, if the feature parameters σ and N, the confidence δ, and the accuracy ϵ are chosen so that the following conditions hold:

$$\begin{aligned}
&\gamma^2\sigma^2 \geq \frac{1}{2}(13s)^{\frac{2}{d}},\\
&N = \frac{4}{\epsilon^2}\left(1 + 4\gamma\sigma d\sqrt{1+\sqrt{\frac{12}{d}\log\frac{m}{\delta}}} + \sqrt{\frac{1}{2}\log\left(\frac{1}{\delta}\right)}\right)\\
&m \geq 4(2\gamma^2\sigma^2+1)^d \log\frac{N^2}{\delta},\\
&d \geq \frac{4\log\left(\frac{N^2}{\delta}\right)}{\log\left(\frac{\gamma^2\sigma^2}{e\log(2\gamma^2\sigma^2+1)}\right)},
\end{aligned}$$

then, with probability at least $1-5\delta$ the following error bound holds:

$$\begin{aligned}
\sqrt{\int_{\mathbb{R}^d} |f^{\#}(\mathbf{x}) - f^\star(\mathbf{x})|^2\,\mathrm{d}\mu} \leq\; & C'\left(1 + N^{\frac{1}{2}}\,s^{-\frac{1}{2}}\,m^{-\frac{1}{4}}\log^{1/4}\left(\frac{1}{\delta}\right)\right)\kappa_{s,1}(\mathbf{c}^\star)\\
&+ C\left(1 + N^{\frac{1}{2}}m^{-\frac{1}{4}}\,\log^{1/4}\left(\frac{1}{\delta}\right)\right)\sqrt{\epsilon^2\,\|f\|_\rho^2 + 4\nu^2}.
\end{aligned} \tag{39}$$

where $C, C' > 0$ are constants.

The proof of this lemma (see Appendix C) relies on demonstrating that given the assumptions on the data samples $\mathbf{x}_k$ and random weights $\boldsymbol{\omega}_j$, the corresponding random feature matrix $\mathbf{A}$ (see Step 4 in Algorithm 1) has a small mutual coherence $\mu_{\mathbf{A}}$, which we recall below.

Definition 4 *(Mutual coherence [20]).* Let $\mathbf{A} \in \mathbb{C}^{m\times N}$ be a matrix with columns $\mathbf{a}_1, \ldots, \mathbf{a}_N$. The mutual coherence of $\mathbf{A}$ is defined as

$$\mu_{\mathbf{A}} = \sup_{\ell\neq j}\left\{|\mu_{j\ell}|,\ \mu_{j\ell} := \frac{\langle \mathbf{a}_j, \mathbf{a}_\ell\rangle}{\|\mathbf{a}_j\|\|\mathbf{a}_\ell\|}\right\}. \tag{40}$$

To establish Lemma 2, we argue that a small mutual coherence $\mu_{\mathbf{A}}$ is itself a consequence of the *bounded separation* of the randomly drawn weights. That is, consider a collection of random weights $\{\boldsymbol{\omega}_j\}_{j=1}^N$ in $\mathbb{R}^d$. For $\gamma > 0$ and a function $\psi : \mathbb{R}^d \to \mathbb{R}$, we define the quantities

$$\Gamma_{j\ell} := \psi\left(\gamma(\boldsymbol{\omega}_j - \boldsymbol{\omega}_\ell)\right), \quad \Gamma_{min} := \min_{j\neq\ell}\Gamma_{j\ell}, \quad \Gamma_{max} := \max_{j\neq\ell}\Gamma_{j\ell}. \tag{41}$$

We can quantify its separation with respect to ψ by bounding Γ_{max} and Γ_{min} by values depending on N and other dimensional constants. In the setting of Theorem 1 where the sampling points $\mathbf{x}_i$'s are i.i.d. Gaussian, the bounded separations hold for $\psi\left(\gamma(\boldsymbol{\omega}_j - \boldsymbol{\omega}_\ell)\right) = \exp\left(-2\gamma^2\pi^2\|\boldsymbol{\omega}_j - \boldsymbol{\omega}_\ell\|^2\right)$. Consequently, by utilizing the fact that the weights $\boldsymbol{\omega}$'s are normally distributed, we show that the collection $\{\boldsymbol{\omega}_j\}_{j=1}^N$ has bounded separation by establishing bounds on Γ_{max} and Γ_{min} depending on N.

Given the bounds on Γ_{max} and Γ_{min}, by employing the Bernstein's inequality, we then establish that $\mu_{\mathbf{A}} = \mathcal{O}\left(\Gamma_{max}\right)$ with high probability, as long as $m \geq \frac{4}{\Gamma_{min}^2}\log\frac{N^2}{\delta}$. Consequently, we utilize a result from

compressive sensing regarding the stability of the BP formulation (see, e.g. [20]) to complete the proof of Lemma 2.

5. Experimental results

In this section, we test the SRFE approaches (both Algorithm 1 and 2) on benchmark synthetic examples and on two applications (data-driven approximations for the NACA airfoil and HyShot 30 datasets). We compare our method to the RFF method, a two-layer neural network, and the PCE algorithm. The examples and comparisons show that the SRFE approaches provide a consistent result over several hyperparameters and outperform the other methods in some of the more realistic settings (i.e. limited data, no prior information on the sampling distribution, etc.).

Throughout Section 5, we set $\phi(\mathbf{x};\boldsymbol{\omega},p) = \sin(\langle \mathbf{x},\boldsymbol{\omega}\rangle + p)$, unless otherwise specified. The parameterized functions now include an offset term p, referred to as the *bias*, which allows for the addition of a phase to capture both the sine and cosine basis terms. For each experiment, the hyperparameters for the SRFE approach will be specified, which include: the number of random features N, the number of data samples m, the dimension of the data d, the feature sparsity q (for Algorithm 2), the distributions for the data and weights, and the basis pursuit parameter η. To measure the error, we use the relative ℓ^2 error on the test set (i.e. the relative testing error) defined as:

$$\text{Error} = \sqrt{\frac{\sum_{k\in\text{Test}} |f(\mathbf{x}_k) - f^\sharp(\mathbf{x}_k)|^2}{\sum_{k\in\text{Test}} |f(\mathbf{x}_k)|^2}},$$

where f is the target function and $f^\sharp$ denotes the solution from either Algorithm 1, Algorithm 2, or the comparison algorithms. If we are randomly drawing the data, then we construct a test set using 5000 random samples (distinct from the training set). Otherwise, the discussion for each example specifies the training-testing data split percentages. For consistency between experiments, we do not use the optional pruning step. In fact, the output from Algorithms 1 and 2 are sparse in these examples, although this is not guaranteed by the theory. We use the SPGL1 algorithm [60] to solve the sparse basis pursuit step in Algorithms 1 and 2.

5.1. Comparison with two layer neural network

In the first example, we show that Algorithm 2 outperforms a shallow neural network on the approximation of an order-2 function:

$$f(x_1,\dots,x_{10}) = \frac{1}{10}\sum_{\ell=1}^{9} \frac{\exp(-x_\ell^2)}{1+x_{\ell+1}^2}$$

in the data-scarce regime. For Algorithm 2, we set $\eta = 0.01$, $q = 2$ or $q = 10$, $m = 250$, $\mathbf{x} \sim \mathcal{U}[-1,1]^{10}$, $\boldsymbol{\omega} \sim \mathcal{N}(0,1)$, and $p \sim \mathcal{U}[0,2\pi]$.

In Fig. 1, we compare the SRFE (with $N = 5000$) to a standard two-layer fully connected ReLU network with 500 and 5000 trainable parameters. The ReLU network is trained using gradient descent. The ReLU network with 500 trainable parameters is included so as to match the number of active parameters in the SRFE. The SRFE with $q = d = 10$ is more accurate than the shallow network in this data regime. When $q = 2$, the error of SRFE-S is smaller than that of the SRFE results with $q = d$ and is one order of magnitude smaller than the neural network.

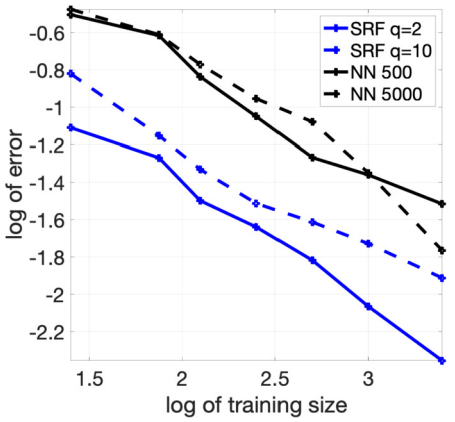

Fig. 1. Function Approximation: Comparison of relative testing error versus the size of the training set for the sparse random feature model with $q = 2$ and $q = 10$ and for the two-layer ReLU network using 500 and 5000 trainable parameters. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

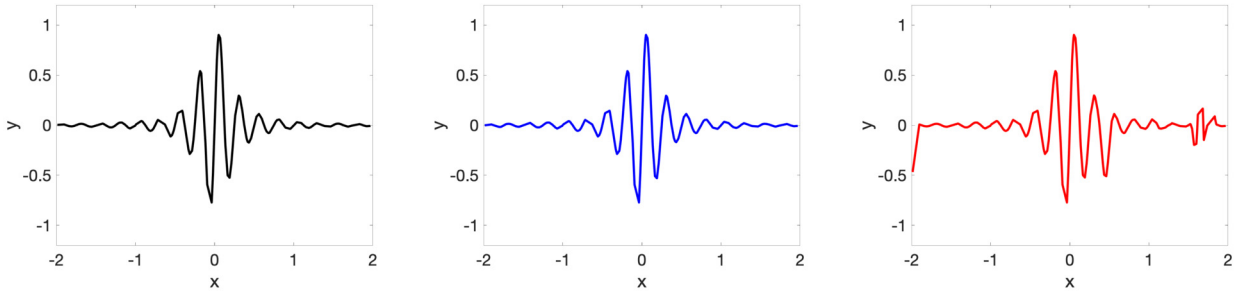

Fig. 2. Comparison, Overfitting: The first figure is the target function, the second and third figures are the approximations via the SRFE and the OLS methods respectively with the same $m = 200$ randomly sampled points.

5.2. Overfitting and noise

In this example, we consider the interpolation problem and provide a visual comparison of the recovery of one-dimensional functions using the SRFE algorithm and the ordinary least squares (OLS) approach. In this case, OLS refers to the min-norm interpolator, i.e. the solution to

$$\arg\min_{\mathbf{c}} \quad \|\mathbf{c}\|_2 \qquad \text{s.t.} \qquad \mathbf{A}\mathbf{c} = \mathbf{y}.$$

The first plot of Fig. 2 is the target function (a sine packet), the second and third plots are the approximations using the SRFE and the OLS methods respectively using the same training set of $m = 200$ randomly sampled points sampled from $\mathcal{U}[-2, 2]$ and the same feature matrix with $N = 2500$, $\boldsymbol{\omega} \sim \mathcal{N}(0, 4\pi^2)$, and $p \sim \mathcal{U}[-\pi, \pi]$. The basis pursuit parameter is set to $\eta = 10^{-3}$. Since we are comparing interpolators, the OLS approximation leads to the appearance of high-frequency aliasing. We observed that when the min-norm interpolator is replaced by a ridge regression training problem, the SRFE approach produces better test errors in the low-data limit.

In Fig. 3, noisy one dimensional data is considered. The first column includes the Runge function (top) and a triangle function (bottom) each with 5% relative noise. The second and third columns are the approximations using the SRFE and the OLS (i.e. the min-norm interpolator) approaches respectively with the same $m = 200$ randomly sampled points. The first row uses $\boldsymbol{\omega} \sim \mathcal{N}(0, \pi^2)$ and the second row uses $\boldsymbol{\omega} \sim \mathcal{N}(0, 4\pi^2)$. Both the SRFE and OLS approaches use the same feature matrix with $N = 2500$ and $p \sim \mathcal{U}[-\pi, \pi]$. The basis pursuit parameter is set to $\eta = 10^{-3}$. The results using the SRFE are more accurate (in terms of interpolation and testing error) and contain less noise artifacts. Note that since the basis

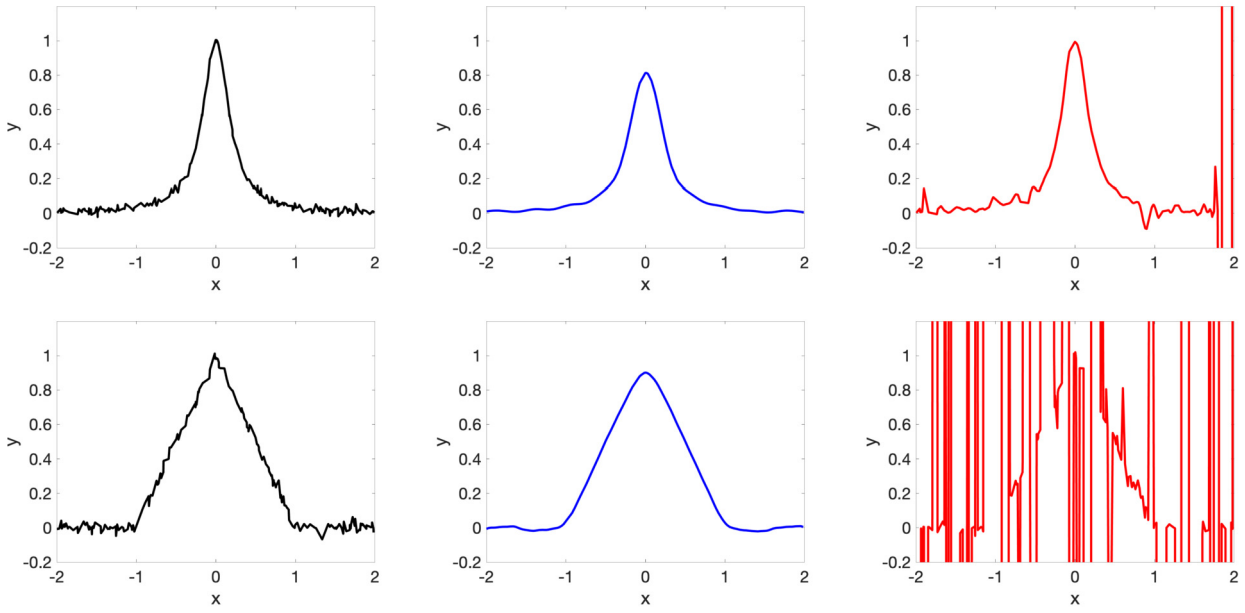

Fig. 3. Comparison, Noise: The first column includes the Runge function (top) and a triangle function (bottom) each with 5% relative noise. The second and third columns are the approximations via the SRFE and the OLS methods respectively with the same 200 randomly sampled points.

Table 1
Low Order Examples The table contains the relative test error (as a percentage) for approximating various functions using different q values. The purple values represent the order of the function. We fix $m = 1000$ and $N = 10000$ with random sine features. We draw $\mathbf{x} \sim \mathcal{U}[-1,1]^d$ and the nonzero values of $\boldsymbol{\omega}$ are drawn from $\mathcal{N}(\mathbf{0}, \sigma^2)$.

$f(\mathbf{x})$	σ	d	$q=1$	$q=2$	$q=3$	$q=5$
$\left(\sum_{i=1}^d x_j\right)^2$	0.1	1	0.82	5.71×10^{-6}	6.92×10^{-5}	8.3×10^{-4}
$(1+\|\mathbf{x}\|_2^2)^{-1/2}$	1	5	3.27	1.60	1.95	1.72
$\sqrt{1+\|\mathbf{x}\|_2^2}$	1	5	1.02	0.73	0.80	1.10
$\mathrm{sinc}(x_1)\mathrm{sinc}(x_3)^3 + \mathrm{sinc}(x_2)$	π	5	12.90	1.19	1.13	3.51
$\frac{x_1 x_2}{1+x_3^6}$	1	5	100.30	.53	4.95	5.06
$\sum_{i=1}^d \exp(-\|x_i\|)$	1	100	0.91	1.43	1.57	1.96

is trigonometric, the approximations are smooth. The OLS results have overfit the data, even when the feature parameter N is varied.

5.3. Low order approximations

In Table 1, we test the effect of varying q for different functions using Algorithm 2 and record the relative errors. The highlighted (purple) values represent the explicit order of the function. We set the parameters to $m = 1000$ and $N = 10000$. The data is sampled from $\mathcal{U}[-1,1]^d$ and the nonzero values of $\boldsymbol{\omega}$ are drawn from $\mathcal{N}(\mathbf{0}, \sigma^2)$, where σ and d are included in the table for each example. The basis pursuit parameter is $\eta = 0.01$.

In the second and third examples, while the functions are order $q = d$ functions, they enjoy better accuracy for $q = 2$. This could be due to several phenomena. The first is that, with fixed m and N, the error may increase as q increase (see Theorem 3). However, this should partially be mitigated since we chose $N = 10000$ large enough. Another reason is that, with respect to some expansion (i.e. Fourier or Taylor), the functions can be written as an order $q < d$ function within some level of accuracy. This motivates further

Table 2
HyShot 30 and NACA Sound Datasets: Average relative train and test errors over 10 random trials (as a percentage). For the shallow NN, we choose the hidden layer so that the total number of parameters match N.

HyShot 30	$N = 100$	$N = 200$	$N = 400$	$N = 800$
SRFE with Sine	6.95	6.23	5.76	5.64
SRFE with ReLU	1.40	1.45	1.51	1.59
Random Fourier Features	84.23	89.99	95.17	97.84
Two-layer ReLU Network	7.29	11.50	11.19	11.33
NACA Sound	$N = 250$	$N = 1500$	$N = 5000$	$N = 10000$
SRFE (Train)	3.22	2.30	2.30	2.31
SRFE (Test)	3.22	3.04	2.77	2.78
SRFE (Average Sparsity)	250	364.4	185.7	185.7
Random Fourier Features (Train)	3.22	0.25	0.20	0.19
Random Fourier Features (Test)	7.45	2.13×10^8	1.69×10^8	1.48×10^8

investigations in future work. The other examples show a clear transition when the correct range for q is obtained.

5.4. HyShot 30 data

In Table 2, we apply the SRFE on the HyShot dataset (Hypersonics Flow Data [13]) and measure the relative testing error as a function of N (the number of random features). The input space is $d = 7$ dimensional and the dataset includes 52 total samples (which we split into 26-26, i.e. $m = 26$). We set $\eta = 0.01$, $\boldsymbol{\omega} \sim \mathcal{N}(\mathbf{0}, 4\pi^2)$, $p \sim \mathcal{U}[0, 1]$, and $q = 7$ (no coordinate sparsity is assumed). In this setting, we have $N \gg m$, which causes the RFF model and the two-layer fully connected ReLU network to overfit on the data (the training loss is small). The NN is trained using the gradient descent algorithm and no differences were observed when using other standard optimizers.

When using $\phi(\mathbf{x}; \boldsymbol{\omega}, p) = \sin(\langle \mathbf{x}, \boldsymbol{\omega} \rangle + p)$, the SRFE produces consistent testing error which decreases as N increases. On the other hand, when $\phi(\mathbf{x}; \boldsymbol{\omega}, p) = \mathrm{ReLU}(\langle \mathbf{x}, \boldsymbol{\omega} \rangle + p)$, the results using SRFE achieve a smaller overall testing error but do not improve with N. Table 2 shows that unlike the SRFE, no gains are made from increasing the number of trainable parameters in the shallow NN model.

5.5. NACA sound dataset

We comparing the SRFE and the RFF models without coordinate sparsity (i.e. $q = d$) on the National Advisory Committee for Aeronautics (NACA) sound dataset [18] and measure the relative training and testing error as a function of N. The input space is $d = 5$ dimensional, the total number of samples is 1503, the train-test split $80 - 20$ (i.e. $m = 1202$), $\eta = 0.01$, $\boldsymbol{\omega} \sim \mathcal{N}(\mathbf{0}, 1)$, and $p \sim \mathcal{U}[0, 1]$. The relative testing errors in Table 2 indicate an overall consistent result, in terms of the coefficient sparsity and the errors, when using the SRFE approach. The RFF model overfits as N increases beyond the size of the training set.

5.6. Comparison with sparse PCE

In Fig. 4, we compare the SRFE-S approach with the Sparse PCE approach [29] using various random sampling methods on the Ishigami example $f(x_1, x_2, x_3) = \sin(x_1) + 7\sin^2(x_2) + 0.1x_3^4 \sin(x_1)$ which is of order 2. The plots in Fig. 4 show the testing set points (y-axis) against the output prediction of a model (x-axis). The line $y = x$ on the plot indicates a perfect fit, while any deviation from the line

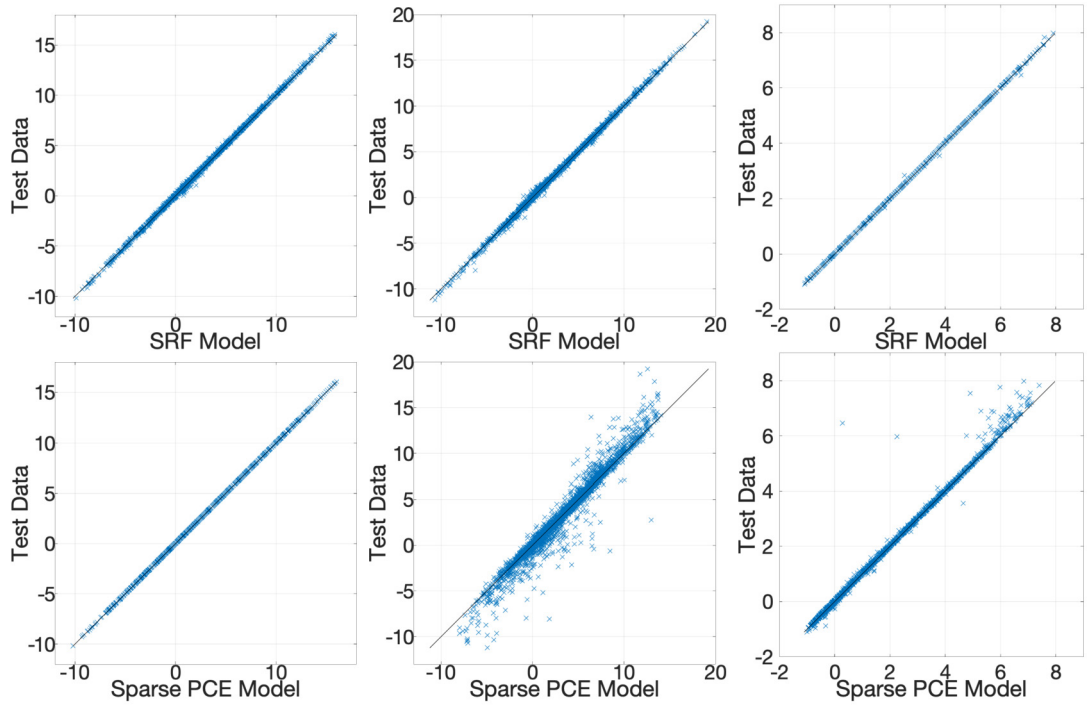

Fig. 4. Comparison with Sparse PCE. Each scatter plot is the model response versus the data. The first row is the SRFE and the second row is the Sparse PCE model. The first column uses i.i.d. samples from $\mathcal{U}[-\pi,\pi]^d$, the third column uses i.i.d. samples from $\mathcal{N}\left(0,\frac{1}{4}\mathbf{I}_d\right)$, and the second column uses the sum of samples from $\mathcal{N}\left(0,\frac{1}{100}\mathbf{I}_d\right)$ and $\mathcal{U}[-\pi,\pi]^d$. Each model uses $N = 3276$ features (which is equivalent to a degree-25 polynomial system in the case of the Sparse PCE approach) and (the same) $m = 200$ random samples. While the Sparse PCE performs well on the uniform distribution (first row), the SRFE produces accurate approximations in all cases.

indicates the errors. The first column of Fig. 4 uses i.i.d. samples $\mathbf{x}_k \sim \mathcal{U}[-\pi,\pi]^d$, the third column uses i.i.d. samples $\mathbf{x}_k \sim \mathcal{N}\left(0,\frac{1}{4}\mathbf{I}_d\right)$, and the second column uses a mixed distribution $\mathbf{x}_k = \mathbf{x}_{k,1} + \mathbf{x}_{k,2}$ where $\mathbf{x}_{k,1} \sim \mathcal{N}\left(0,\frac{1}{100}\mathbf{I}_d\right)$ and $\mathbf{x}_{k,1} \sim \mathcal{U}[-\pi,\pi]^d$. Each model uses $N = 3276$ features (which is equivalent to a degree-25 polynomial system in the case of the Sparse PCE approach) and (the same) $m = 200$ random samples. The hyperparameters for the SRFE-S are set to $q = 2$, $\boldsymbol{\omega} \sim \mathcal{N}\left(\mathbf{0},\frac{9}{4}\pi^2\right)$, and $p \sim \mathcal{U}[0,2\pi]$. When using uniformly random samples, the Sparse PCE approach produces lower testing error (0.24% versus 1.43%), which continues to perform well as N increases. This is due in part to the fact that the orthogonal polynomial basis (in this case, the Legendre basis) has knowledge of the input distribution. When the samples are Gaussian, the SRFE produces a more accurate solution than the Sparse PCE method (0.44% versus 6.24%). For the mixture case, the SRFE outperforms the Sparse PCE method (2.11% versus 15.05%). Note that the Sparse PCE must derive the orthogonal basis from the data (or use the Legendre basis as its default), where as, at least experimentally, our approach is applicable to a larger class of input distributions.

6. Conclusion

We proposed the sparse random features method as a new approach in function approximation. For low order functions, i.e. functions that admit a decomposition to terms depending on only a few of the independent variables, we introduce low order random features. By utilizing techniques from compressive sensing and probability, we provided generalization bounds for the proposed scheme and established sample and feature complexities. On several examples, we showed improved accuracy over other popular approximation schemes. As part of the future work, we intend to explore the avenues to incorporate additional functional structures into the proposed framework with the hope of further improving the approximation properties of the proposed scheme. In addition, by considering random features within a ridge regression approach,

[57] showed that the computational gains of random features come at the expense of learning accuracy, $N = \mathcal{O}(\sqrt{m}\log m)$ features are sufficient for $\mathcal{O}(1/\sqrt{m})$ error, where m is the number of samples. Utilizing this result in our proposed framework is an interesting direction which is left for future work.

Acknowledgments

We thank Zhijun Chen, Jiannan Jiang, Kameron Harris, Andrea Montanari, Rene Vidal, and Yuege Xie for their helpful feedback which led to significant improvements.

Appendix A. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.acha.2022.08.003.

References

[1] B. Adcock, S. Brugiapaglia, C.G. Webster, Compressed sensing approaches for polynomial approximation of high-dimensional functions, in: Compressed Sensing and its Applications, 2017, pp. 93–124.
[2] B. Adcock, Infinite-dimensional compressed sensing and function interpolation, Found. Comput. Math. 18 (3) (2018) 661–701.
[3] S. Arora, S. Du, W. Hu, Z. Li, R. Wang, Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks, in: International Conference on Machine Learning, 2019, pp. 322–332, PMLR.
[4] A.R. Barron, Universal approximation bounds for superpositions of a sigmoidal function, IEEE Trans. Inf. Theory 39 (3) (1993) 930–945.
[5] H.-D. Block, The perceptron: a model for brain functioning. I, Rev. Mod. Phys. 34 (1) (1962) 123.
[6] T.T. Cai, G. Xu, J. Zhang, On recovery of sparse signals via ℓ_1 minimization, IEEE Trans. Inf. Theory 55 (7) (2009) 3388–3397.
[7] E.J. Candes, J.K. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements, Commun. Pure Appl. Math. 59 (8) (2006) 1207–1223.
[8] E.J. Candes, T. Tao, Near-optimal signal recovery from random projections: universal encoding strategies?, IEEE Trans. Inf. Theory 52 (12) (2006) 5406–5425.
[9] R. Chitta, R. Jin, A.K. Jain, Efficient kernel clustering using random Fourier features, in: 2012 IEEE 12th International Conference on Data Mining, IEEE, 2012, pp. 161–170.
[10] A. Chkifa, N. Dexter, H. Tran, C. Webster, Polynomial approximation via compressed sensing of high-dimensional functions on lower sets, Math. Comput. 311 (87) (2018) 1415–1450.
[11] P.G. Constantine, E. Dow, Q. Wang, Active subspace methods in theory and practice: applications to Kriging surfaces, SIAM J. Sci. Comput. 36 (4) (2014) A1500–A1524.
[12] P.G. Constantine, A. Eftekhari, J. Hokanson, R.A. Ward, A near-stationary subspace for ridge approximation, Comput. Methods Appl. Mech. Eng. 326 (2017) 402–404.
[13] P.G. Constantine, M. Emory, J. Larsson, G. Iaccarino, Exploiting active subspaces to quantify uncertainty in the numerical simulation of the HyShot II scramjet, J. Comput. Phys. 302 (2015) 1–20.
[14] R. DeVore, G. Petrova, P. Wojtaszczyk, Approximation of functions of few variables in high dimensions, Constr. Approx. 33 (1) (2011) 125–143.
[15] D.L. Donoho, Compressed sensing, IEEE Trans. Inf. Theory 52 (4) (2006) 1289–1306.
[16] A. Doostan, H. Owhadi, A non-adapted sparse approximation of PDEs with stochastic inputs, J. Comput. Phys. 230 (8) (2011) 3015–3034.
[17] S.S. Du, X. Zhai, B. Poczos, A. Singh, Gradient descent provably optimizes over-parameterized neural networks, arXiv preprint, arXiv:1810.02054, 2018.
[18] D. Dua, C. Graff, UCI machine learning repository, 2017.
[19] M. Fornasier, K. Schnass, J. Vybiral, Learning functions of few arbitrary linear parameters in high dimensions, Found. Comput. Math. 12 (2) (2012) 229–262.
[20] S. Foucart, H. Rauhut, A Mathematical Introduction to Compressive Sensing, Springer, 2013.
[21] K.D. Harris, Additive function approximation in the brain, arXiv preprint, arXiv:1909.02603, 2019.
[22] W. Heisenberg, About the descriptive content of quantum theoretical kinematics and mechanics, Z. Phys. (1927) 172–198.
[23] A. Jacot, F. Gabriel, C. Hongler, Neural tangent kernel: convergence and generalization in neural networks, arXiv preprint, arXiv:1806.07572, 2018.
[24] L.K. Jones, A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training, Ann. Stat. 20 (1) (1992) 608–613.
[25] F. Kuo, I. Sloan, G. Wasilkowski, H. Woźniakowski, On decompositions of multivariate functions, Math. Comput. 79 (270) (2010) 953–966.

[26] Y. Li, Y. Liang, Learning overparameterized neural networks via stochastic gradient descent on structured data, arXiv preprint, arXiv:1808.01204, 2018.
[27] Z. Li, J.-F. Ton, D. Oglic, D. Sejdinovic, Towards a unified analysis of random Fourier features, in: International Conference on Machine Learning, 2019, pp. 3905–3914, PMLR.
[28] W. Maass, H. Markram, On the computational power of circuits of spiking neurons, J. Comput. Syst. Sci. 69 (4) (2004) 593–616.
[29] S. Marelli, B. Sudret, UQlab: a framework for uncertainty quantification in Matlab, in: Vulnerability, Uncertainty, and Risk: Quantification, Mitigation, and Management, 2014, pp. 2554–2563.
[30] S. Mei, A. Montanari, The generalization error of random features regression: precise asymptotics and double descent curve, arXiv preprint, arXiv:1908.05355, 2020.
[31] F. Moosmann, B. Triggs, F. Jurie, Randomized clustering forests for building fast and discriminative visual vocabularies, in: NIPS, 2006, NIPS.
[32] D. Potts, M. Schmischke, Approximation of high-dimensional periodic functions with Fourier-based methods, arXiv preprint, arXiv:1907.11412, 2019.
[33] D. Potts, M. Schmischke, Learning multivariate functions with low-dimensional structures using polynomial bases, arXiv preprint, arXiv:1912.03195, 2019.
[34] A. Rahimi, B. Recht, Uniform approximation of functions with random bases, in: 2008 46th Annual Allerton Conference on Communication, Control, and Computing, IEEE, 2008, pp. 555–561.
[35] A. Rahimi, B. Recht, Weighted sums of random kitchen sinks: replacing minimization with randomization in learning, Adv. Neural Inf. Process. Syst. (2008) 1313–1320.
[36] A. Rahimi, B. Recht, et al., Random features for large-scale kernel machines, in: NIPS, 2007, pp. 1–10, NIPS.
[37] H. Rauhut, R. Ward, Sparse Legendre expansions via l1-minimization, J. Approx. Theory 164 (5) (2012) 517–533.
[38] H. Rauhut, R. Ward, Interpolation via weighted l1 minimization, Appl. Comput. Harmon. Anal. 40 (2) (2016) 3–351.
[39] A. Saltelli, M. Ratto, T. Andres, F. Campolongo, J. Cariboni, D. Gatelli, M. Saisana, S. Tarantola, Global Sensitivity Analysis: the Primer, John Wiley & Sons, 2008.
[40] H. Schaeffer, G. Tran, R. Ward, Extracting sparse high-dimensional dynamics from limited data, SIAM J. Appl. Math. 78 (6) (2018) 3279–3295.
[41] H. Schaeffer, G. Tran, R. Ward, L. Zhang, Extracting structured dynamical systems using sparse optimization with very few samples, Multiscale Model. Simul. 18 (4) (2020) 1435–1461.
[42] B.K. Sriperumbudur, Z. Szabó, Optimal rates for random Fourier features, arXiv preprint, arXiv:1506.02155, 2015.
[43] D.J. Sutherland, J. Schneider, On the error of random Fourier features, arXiv preprint, arXiv:1506.02785, 2015.
[44] T. Yang, Y.-F. Li, M. Mahdavi, R. Jin, Z.-H. Zhou, Nyström method vs random Fourier features: a theoretical and empirical comparison, Adv. Neural Inf. Process. Syst. 25 (2012) 476–484.
[45] I.E.-H. Yen, T.-W. Lin, S.-D. Lin, P.K. Ravikumar, I.S. Dhillon, Sparse random feature algorithm as coordinate descent in Hilbert space, Adv. Neural Inf. Process. Syst. (2014) 2456–2464.
[46] Weiliang Shi, Kristine E. Lee, Grace Wahba, Detecting disease-causing genes by LASSO-Patternsearch algorithm, BMC Proc. (2007) 1–5.
[47] Ruixin Guo, Hongtu Zhu, Sy-Miin Chow, Joseph G. Ibrahim, Bayesian lasso for semiparametric structural equation models, Biometrics (2012) 567–577.
[48] Michael Lim, Trevor Hastie, Learning interactions via hierarchical group-lasso regularization, J. Comput. Graph. Stat. (2015) 627–654.
[49] Saharon Rosset, Grzegorz Swirszcz, Nathan Srebro, Ji Zhu, l-1 regularization in infinite dimensional feature spaces, in: International Conference on Computational Learning Theory, 2007, pp. 544–558.
[50] Alain Rakotomamonjy, Rémi Flamary, Florian Yger, Learning with infinitely many features, in: Machine Learning, 2013, pp. 43–66.
[51] Simon S. Du, Wei Hu, Sham M. Kakade, Jason D. Lee, Qi Lei, Few-shot learning via learning the representation, provably, arXiv preprint, arXiv:2002.09434, 2020, 1–30.
[52] Mário A.T. Figueiredo, Robert D. Nowak, Stephen J. Wright, Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems, IEEE J. Sel. Top. Signal Process. (2007) 586–597.
[53] Kirthevasan Kandasamy, Yaoliang Yu, Additive approximations in high dimensional nonparametric regression via the SALSA, in: International Conference on Machine Learning, 2016, pp. 69–78.
[54] Guodong Liu, Hong Chen, Heng Huang, Sparse shrunk additive models, in: International Conference on Machine Learning, 2020, pp. 6194–6204.
[55] Hong Chen, Xiaoqian Wang, Cheng Deng, Heng Huang, Group sparse additive machine, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 197–207.
[56] Pradeep Ravikumar, John Lafferty, Han Liu, Larry Wasserman, Sparse additive models, J. R. Stat. Soc., Ser. B, Stat. Methodol. (2009) 1009–1030.
[57] Alessandro Rudi, Lorenzo Rosasco, Generalization properties of learning with random features, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 35–3225.
[58] Enayat Ullah, Poorya Mianjy, Teodor Vanislavov Marinov, Raman Arora, Streaming kernel PCA with $O(\sqrt{n})$ random, in: Features Proceedings of the 32nd International Conference on Neural Information Processing System, 2018, pp. 7322–7332.
[59] Zoltán Szabó, Bharath Sriperumbudur, On kernel derivative approximation with random Fourier features, in: Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics, 2019, pp. 827–836.
[60] E. van Den Berg, M.P. Friedlander, Probing the Pareto frontier for basis pursuit solutions, SIAM J. Sci. Comput. (2009) 890–912.