SIAM REVIEW
Vol. 66, No. 3, pp. 535–571

© 2024 Society for Industrial and Applied Mathematics

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

# Operator Learning Using Random Features: A Tool for Scientific Computing*

Nicholas H. Nelsen†
Andrew M. Stuart†

**Abstract.** Supervised operator learning centers on the use of training data, in the form of input-output pairs, to estimate maps between infinite-dimensional spaces. It is emerging as a powerful tool to complement traditional scientific computing, which may often be framed in terms of operators mapping between spaces of functions. Building on the classical random features methodology for scalar regression, this paper introduces the function-valued random features method. This leads to a supervised operator learning architecture that is practical for nonlinear problems yet is structured enough to facilitate efficient training through the optimization of a convex, quadratic cost. Due to the quadratic structure, the trained model is equipped with convergence guarantees and error and complexity bounds, properties that are not readily available for most other operator learning architectures. At its core, the proposed approach builds a linear combination of random operators. This turns out to be a low-rank approximation of an operator-valued kernel ridge regression algorithm, and hence the method also has strong connections to Gaussian process regression. The paper designs function-valued random features that are tailored to the structure of two nonlinear operator learning benchmark problems arising from parametric partial differential equations. Numerical results demonstrate the scalability, discretization invariance, and transferability of the function-valued random features method.

**Key words.** scientific machine learning, operator learning, random feature, surrogate model, kernel ridge regression, parametric partial differential equation

**MSC codes.** 68T05, 65D40, 62J07, 62M45, 68W20, 35R60

**DOI.** 10.1137/24M1648703

## Contents

[*] Published electronically August 8, 2024. This paper originally appeared in *SIAM Journal on Scientific Computing*, Volume 43, Number 5, 2021, pages A3212–A3243, under the title "The Random Feature Model for Input-Output Maps between Banach Spaces."

https://doi.org/10.1137/24M1648703

**Funding:** The original work [111] was supported by the National Science Foundation (NSF) Graduate Research Fellowship Program under award DGE-1745301, NSF award DMS-1818977, Office of Naval Research (ONR) award N00014-17-1-2079, NSF award AGS-1835860, and ONR award N00014-19-1-2408. This SIGEST article is supported by the Amazon/Caltech AI4Science Fellowship held by the first author and by the Department of Defense Vannevar Bush Faculty Fellowship, under ONR award N00014-22-1-2790, held by the second author.

[†] Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA 91125 USA (nnelsen@caltech.edu, astuart@caltech.edu).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

**1. Introduction.** The increased use of machine learning for complex scientific tasks ranging from drug discovery to numerical weather prediction has led to the emergence of the new field of *scientific machine learning.* Scientific machine learning blends modern artificial intelligence techniques with time-tested scientific computing methods in a principled manner to tackle challenging science and engineering problems, even those previously considered to be out of reach due to high-dimensionality or computational cost. A common theme in these physical problems is that the data are typically modeled as infinite-dimensional quantities like velocity or pressure fields. Such continuum objects are spatially and temporally varying functions that have intrinsic smoothness properties and long-range correlations.

Recognizing the need for new mathematical development of learning algorithms that are tailor-made for continuum problems, researchers established the *operator learning* paradigm to build data-driven models that map between infinite-dimensional input and output spaces. An operator is an input-output relationship such that each input and corresponding output is infinite-dimensional. For example, the mapping from the current temperature in a room to the temperature one hour later is an operator. This is because temperature at a fixed time is a function characterized by its values at an uncountably infinite number of spatial locations. More generally, one may consider the semigroup generated by a time-dependent partial differential equation (PDE) mapping the initial condition to the solution at a later time. A more concrete example is the mapping $F^\dagger\colon (a,f)\mapsto u$ from coefficient function $a$ and source term $f$ to solution function $u$ governed by the elliptic PDE $-\nabla\cdot(a\nabla u)=f$, equipped with appropriate boundary conditions. The paper returns to this example later on.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

The paper focuses exclusively on supervised operator learning. This is concerned with learning models to fit infinite-dimensional input-output pairs of (what is then known as labeled) training data. However, the operator learning framework is quite general and encompasses continuum problems that involve potentially diverse sources of data, going beyond the supervised learning setting. In the unsupervised setting, only unlabeled data is available. One example of this is the estimation of a covariance operator: the dataset comprises random functions drawn from the probability measure whose covariance is to be estimated. Alternatively, the observed data might only consist of indirect or sparse measurements of a system, often also corrupted by noise, as is common in inverse problems; blind deconvolution is an important example.

Infinite-dimensional quantities must always be discretized when represented on a computer or in experiments. What distinguishes supervised operator learning from traditional supervised learning architectures that operate on high-dimensional discretized vectors is that in the continuum limit of infinite resolution, operator learning architectures have a well-defined and consistent meaning. They capture the underlying continuum structure of the problem and not artifacts due to the particular choice of discretization. Indeed, for a fixed set of trainable parameters, operator learning methods by construction produce consistent results given any finite-dimensional discretization of the conceptually infinite-dimensional data. That is, they are inherently dimension- and discretization-independent. Practically, this means that once learned at one resolution, the operator can be transferred to any other resolution without the need for retraining. Growing empirical evidence suggests that operator learning exhibits excellent performance as a tool to accelerate model-centric tasks in science and engineering or to discover unknown physical laws from experimental data. However, the mathematical theory of operator learning is far from complete, which limits its impact.

The goal of this paper is to develop an operator learning methodology with strong theoretical foundations that is also especially well suited for the task of speeding up otherwise prohibitively expensive many-query problems. The need for repeated evaluations of a complex, costly, and slow forward model for different configurations of a system parameter occurs in various science and engineering domains. The true model is often a PDE and the parameter, serving as input to the PDE model, is often a continuum quantity. For instance, in the heat equation, the input is its initial condition, and in the preceding elliptic PDE example, the input is its coefficient and forcing functions. In contrast to the big data regime that dominates computer vision and other technological fields, only a relatively small amount of high resolution labeled data can be generated from computer simulations or physical experiments in scientific applications. Fast approximate surrogates built from this limited available data that can efficiently and accurately emulate the full order model would be highly advantageous in downstream outer loop applications.

The present work demonstrates that the *random feature model* (RFM) has considerable potential for such a purpose when formulated as a map between function spaces. In contrast to more complicated deep learning approaches, the function-valued random features algorithm involves learning the coefficients of a linear expansion composed of random maps. For a suitable training objective function, this is a finite-dimensional convex, quadratic optimization problem. Equivalently, the paper shows that this supervised training procedure is equivalent to ridge regression over a reproducing kernel Hilbert space (RKHS) of operators. As a consequence of the careful construction of the method as a mapping between infinite-dimensional Banach spaces, the resulting

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

RFM surrogate enjoys rigorous convergence guarantees and scales favorably with respect to (w.r.t.) the high input and output dimensions arising in practical, discretized applications. Numerically, the method achieves a small test error for learning a semigroup and the solution operator of a parametric elliptic PDE.

This section continues with a literature review and then a summary of the main contributions of the paper.

**1.1. Literature Review.** Two different lines of research have emerged that address PDE approximation problems with scientific machine learning techniques. The first perspective takes a more traditional approach akin to point collocation methods from the field of numerical analysis. Here, the goal is to use a deep neural network (NN) [125] or other function class [30] to solve a prescribed initial boundary value problem with as high an accuracy as possible. Given a point cloud in a possibly high-dimensional spatiotemporal domain $\mathcal{D}$ as input data, the prevailing approach first directly parametrizes the PDE solution field as an NN and then optimizes the NN parameters by minimizing the PDE residual w.r.t. some loss functional using variants of stochastic gradient descent (see [78, 125, 134, 49] and the references therein). To clarify, the object approximated with this approach is a function $\mathcal{D} \to \mathbb{R}$ between finite-dimensional spaces. While mesh-free by definition, the method is highly intrusive as it requires full knowledge of the specified PDE. Any change to the original formulation of the initial boundary value problem or related PDE problem parameters necessitates an expensive retraining of the NN approximate solution. We do not explore this first approach any further in this paper.

The second direction takes an operator learning perspective and is arguably more ambitious: use an NN to emulate the infinite-dimensional mapping between an input parameter and the PDE solution itself [22] or a functional of the solution, i.e., a quantity of interest [72]; the latter is widely prevalent in inverse problems [6, 138], optimization under uncertainty [103], and optimal experimental design [4]. For an approximation-theoretic treatment of parametric PDEs, we mention the paper [34]. We emphasize that the object approximated in this setting, unlike in the first approach mentioned in the previous paragraph, is an operator $\mathcal{X} \to \mathcal{Y}$, i.e., the PDE solution operator, where $\mathcal{X}$ and $\mathcal{Y}$ are infinite-dimensional Banach spaces; this map is generally nonlinear. It is this second line of research that inspires our work. We now highlight several subtopics relevant to surrogate modeling and operator learning. Summaries of the state of the art for operator learning may be found in two mathematically oriented review articles on the subject [22, 86].

**Model Reduction.** There are many approaches to surrogate modeling that do not explicitly involve machine learning ideas [120]. The reduced basis method (see [10, 15, 41] and the references therein) is a classical idea based on constructing an empirical basis from continuum or high-dimensional data snapshots and solving a cheaper variational problem; it is still widely used in practice due to computationally efficient offline-online decompositions that eliminate dependence on the full order degrees of freedom. Machine learning extensions to the reduced basis methodology, of both intrusive (e.g., projection-based reduced order models) and nonintrusive (e.g., model-free data only) types, have further improved the applicability of these methods [9, 32, 58, 59, 70, 94, 121, 131]. However, the input-output maps considered in these works involve high dimension in only one of the input space or the output space, not both. A line of research aiming to more closely align model reduction with operator learning is the work on deep learning–based reduced order models (ROMs) [25, 55, 56, 57]; some of these studies also derive approximation guarantees

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

for the ROMs. Other popular surrogate modeling techniques include Gaussian processes [148], polynomial chaos expansions [135], and radial basis functions [146], yet these are only practically suitable for scalar-valued maps with input space of low to moderate dimension, unless strong assumptions are placed on the problem. Classical numerical methods for PDEs may also represent the discretized forward model as a map $\mathbb{R}^K \to \mathbb{R}^K$, where $K$ is the resolution, albeit implicitly in the form of a computer code (e.g., finite element, finite difference, finite volume methods). However, the approximation error is sensitive to $K$ and repeated evaluations of this forward model often become cost prohibitive due to poor scaling with input dimension $K$.

**Operator Learning.** Many earlier attempts to build cheap-to-evaluate surrogate models for PDEs display sensitivity to discretization. There is a suite of work on data-driven discretizations of PDEs that allows for identification of the governing system [8, 20, 100, 119, 137, 142]. However, we note that only the operators appearing in the underlying equation itself are approximated with these approaches, not the solution operator of the PDE; the focus in these works is mostly on model discovery rather than model acceleration. More in line with the theme of the present paper, architectures based on deep convolutional NNs have proven to be quite successful for learning elliptic PDE solution operators. For example, see [55, 143, 149, 152], which take an image-to-image regression approach. Other NNs have been used in similar elliptic problems for quantity of interest prediction [72, 81], error estimation [29], or unsupervised learning [95], and for parametric PDEs more generally [60, 88, 117, 133]. Yet in most of the preceding approaches, the architectures and resulting error are dependent on the mesh resolution. To circumvent this issue, the surrogate map must be well defined on function space and independent of any finite-dimensional realization of the map that arises from discretization. This is not a new idea (see [31, 106, 128] or, for functional data analysis, [77, 107, 126]). The aforementioned reduced basis method is an example, as is the method of [33, 34], which approximates the solution map with sparse Taylor polynomials and achieves optimal convergence rates in idealized settings. Early work in the use of NNs to learn the solution operator, or vector field, defining ODEs and time-dependent PDEs may be found from the 1990s [31, 64, 106, 127]. However, only recently have *practical* machine learning methods been designed to directly operate in infinite dimensions.

Several implementable operator learning architectures were developed concurrently [1, 19, 97, 101, 111, 113, 150]. These include the DeepONet [101], which generalizes and makes practical the main idea in [31], PCA-Net [19], and the RFM from the original version of the present paper [111]. These were followed by neural operators [87, 97] and, in particular, the Fourier neural operator [96]. Details for and comparisons among these architectures are given in [86, sect. 3]. Apart from the RFM, what these methods—which we collectively call "neural operators"—share is a deep learning backbone. The approximation theory of such neural operators is fairly well developed [69, 72, 83, 85, 87, 89, 90, 91, 93]. It includes qualitative universal approximation, i.e., density, results as well as quantitative parameter complexity bounds, that is, the number of NN parameters required to achieve accuracy $\varepsilon$. The paper [93] reveals a "curse of parametric complexity" in which the parameter complexity is shown to be exponentially large in powers of $\varepsilon^{-1}$ to approximate general Lipschitz continuous operators. This aligns with the findings of older work [106] and suggests that efficient neural operator learning is not possible without further assumptions. It turns out that the curse is lifted if enough regularity is assumed. For example, for linear or holomorphic target operators, efficient algebraic approximation rates may be

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

established [2, 69]. However, what rates are possible for sets of operators "in between" holomorphic and Lipschitz operators is still an open question.

One of the simplest classes of operators is that of linear operators. There is a substantial body of work in this setting ranging from the learning of general linear operators [39, 76, 109, 136] to estimating the Green's function of specific linear PDEs [62, 21, 23, 132] and Koopman operators [84]. The linear setting allows for very thorough and sharp statistical analysis that leads to deep insights about the data efficiency of operator learning in terms of problem structure [21, 39, 72]. Some sample complexity results have been obtained for nonlinear functionals and operators, which give the training dataset size needed to obtain $\varepsilon$ accuracy. Most of this theory depends on kernels, either in an RKHS framework [27, 92] or via local averaging (e.g., kernel smoothers) [53, 114]. Error bounds are obtained for encoder–decoder neural operators such as DeepONet and PCA-Net in [99]. These results imply a "curse of sample complexity," i.e., exponentially large sample sizes, for learning Lipschitz operators. Similar to the parameter complexity case, with enough regularity assumed on the operators of interest, as expressed through weighted tensor product structure, operator holomorphy, or analyticity, for example, minimax lower bounds can return to better behaved algebraic rates in the sample size [3, 27, 73, 74]. Moreover, there exist both constructive and nonconstructive estimators that achieve these algebraic convergence rates for operator learning [2, 27, 92].

**Random Features.** The RFM as a mapping between finite-dimensional spaces was formalized in the series of papers [122, 123, 124], building on earlier work in [11, 110, 147]. The RFM is in some sense the simplest possible machine learning model; it may be viewed as an ensemble average of randomly parametrized functions: an expansion in a randomized basis with trainable coefficients. The method of random Fourier features is the most mainstream instantiation of the approach [122]. Here, the RFM is used to approximate popular translation-invariant kernels by averages of sinusoidal functions with random frequencies. This approximation is then used downstream for kernel regression tasks [67]. An equivalent viewpoint is that the RFM approximates the Gaussian process prior distribution in a Gaussian process regression method [148]. However, the choice of random feature map can be much more general than just random sines and cosines. These random features could be defined, for example, by randomizing the internal parameters of an NN. Many papers take this viewpoint [63, 105, 123, 124]. Compared to NN emulators with enormous learnable parameter counts (e.g., $O(10^5)$ to $O(10^7)$; see [51, 52, 95]) and methods that are intrusive or lead to nontrivial implementations [33, 94, 131], the RFM is one of the simplest models to formulate and train. Often $O(10^4)$ or fewer linear expansion coefficients—which are the only free parameters in the RFM—suffice.

The theory of the RFM for real-valued outputs is well developed, partly due to its close connection to kernel methods [7, 26, 75, 122, 146] and Gaussian processes [110, 147], and it includes generalization bounds and dimension-free rates [92, 98, 48, 123, 129, 139, 140]. A quadrature viewpoint on the RFM provides further insight and leads to Monte Carlo sampling ideas [7]. As in modern deep learning practice, for some problem classes the RFM has been shown to perform well even when overparametrized [14, 48, 105]. However, overparametrization is not necessary for good performance; state-of-the-art fast rates are established in the underparametrized regime by [98, 129]. The paper [63] derives similar bounds for random NN approximation of functionals with a random feature-based training strategy.

For the supervised operator learning setting in which inputs and outputs are both

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

infinite-dimensional, kernel [27, 77] and Gaussian process methods [12]—and hence random features—are less explored. The paper [115] performs nonlinear operator learning in the encoder–decoder paradigm, where the input and output spaces are represented by truncated orthonormal bases and the finite-dimensional coefficient-to-coefficient mapping is performed with a kernel smoother. The kernel smoother is then approximated with random Fourier features [122]. A similar idea uses a Gaussian process perspective [12]. For high-dimensional input parameter spaces, the authors of [65, 80] analyze nonparametric kernel regression for parametric PDEs with real-valued solution map outputs. However, the preceding methods have poor computational scalability w.r.t. data dimension and sample size. The RFM alleviates these issues with randomization and efficient convex optimization. The specific random Fourier feature approach of Rahimi and Recht [122] was generalized in [24] to the finite-dimensional matrix-valued kernel setting with vector-valued random Fourier features and to the operator-valued kernel setting in [108]. However, canonical operator-valued kernels are hard to define and the preceding works require explicit knowledge of these kernels. Our viewpoint in the current paper is to develop function-valued random features and work directly with them as a standalone supervised learning method, choosing them for their properties and noting that they implicitly define a kernel, but not working directly with that kernel. An additional benefit of our approach is that it avoids the nonconvex training routines that plague more sophisticated neural operator architectures and, in particular, hinder the development of uncertainty quantification and comprehensive complexity bounds. The key idea underlying our methodology is to formulate the proposed operator random features algorithm on infinite-dimensional space and only discretize it at implementation time. This philosophy in algorithm development has been instructive in a number of areas in scientific computing, as we describe next.

**Other Continuum Algorithms.** The general philosophy of designing algorithms at the continuum level has been hugely successful across disciplines. In PDE-constrained optimization, there is the "optimize-then-discretize" principle [71]. In applied probability, there are Markov chain Monte Carlo algorithms for sampling probability distributions supported on function spaces [36]. The Bayesian formulation of inverse problems on Banach spaces provides another example [138]. Work along similar lines extends numerical linear algebra routines for finite-dimensional vectors and matrices to new ones for infinite-dimensional functions and linear operators [144, 145]. All such methods inherit certain dimension-independent properties that make them more robust and possibly more accurate. Operator learning brings this powerful perspective to machine learning, where it has been promoted as a way of designing and analyzing learning algorithms [66, 47, 130, 44, 45]. Our work may be understood within this general framework.

### 1.2. Contributions.

Our primary contributions in this paper are now listed.

(C1) We develop the RFM, directly formulated on the function space level, for learning nonlinear operators between Banach spaces purely from data. As a method for parametric PDEs, the methodology is nonintrusive but also has the additional advantage that it may be used in settings where only data is available and no model is known.

(C2) We show that our proposed method is more computationally tractable to both train and evaluate than standard kernel methods in infinite dimensions. Furthermore, we show that the method is equivalent to kernel ridge regression performed in a finite-dimensional space spanned by random features and

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

comes equipped with a full convergence theory.

(C3) We apply our operator learning methodology to learn the semigroup defined by the solution operator for the viscous Burgers equation and the coefficient-to-solution operator for the Darcy flow equation.

(C4) We perform numerical experiments that demonstrate two mesh-independent approximation properties that are built into the proposed methodology: invariance of relative error to mesh resolution and evaluation ability on any mesh resolution.

The remainder of this paper is structured as follows. In section 2, we communicate the mathematical framework required to work with the RFM in infinite dimensions, identify an appropriate approximation space, explain the training procedure, and review recent error bounds for the method. We introduce two instantiations of random feature maps that target physical science applications in section 3 and detail the corresponding numerical results for these applications in section 4. We conclude in section 5 with a summary and directions for future work.

## 2. Methodology.

In this work, the overarching problem of interest is the approximation of a map $F^\dagger\colon \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ and $\mathcal{Y}$ are infinite-dimensional spaces of real-valued functions defined on some bounded open subset of $\mathbb{R}^d$, and $F^\dagger$ is defined by $a \mapsto F^\dagger(a) := u$, where $u \in \mathcal{Y}$ is the solution of a (possibly time-dependent) PDE and $a \in \mathcal{X}$ is an input function required to make the problem well-posed. Our proposed approach for this approximation, constructing a surrogate map $F$ for the true map $F^\dagger$, is data-driven, nonintrusive, and based on least squares. Least squares–based methods are integral to the random feature methodology as proposed in low dimensions [122, 123] and generalized here to the infinite-dimensional setting. They have also been shown to work well in other algorithms for high-dimensional numerical approximation [18, 35, 42]. Within the broader scope of reduced order modeling techniques [15], the approach we adopt in this paper falls within the class of data-fit emulators. Essentially, our method approximates the solution manifold

$$\mathcal{M} = \left\{u \in \mathcal{Y}\colon u = F^\dagger(a) \quad \text{and} \quad a \in \mathcal{X}\right\} \tag{2.1}$$

on average. The solution map $F^\dagger$, often being the inverse of a differential operator, is usually smoothing and admits some notion of compactness. Then, the idea is that $\mathcal{M}$ should have some compact, low-dimensional structure or intrinsic dimension. However, actually finding a model $F$ that exploits this structure despite the high dimensionality of the truth map $F^\dagger$ is quite difficult. Further, the effectiveness of many model reduction techniques, such as those based on the reduced basis method, are dependent on inherent properties of the map $F^\dagger$ itself (e.g., analyticity), which in turn may influence the decay rate of the Kolmogorov width of the manifold $\mathcal{M}$ [34]. While such subtleties of approximation theory are crucial to developing rigorous theory and provably convergent algorithms [86], we choose to work in the nonintrusive setting where knowledge of the map $F^\dagger$ and its associated PDE are only obtained through measurement data, and hence detailed characterizations such as those mentioned previously are essentially unavailable. Thus, we emphasize that our proposed operator learning methodology is applicable to general continuum problems with function space data, not just to PDEs.

The remainder of this section introduces the mathematical preliminaries for our methodology. With the goal of operator approximation in mind, in subsection 2.1 we formulate a supervised learning problem in an infinite-dimensional setting. We provide the necessary background on RKHSs in subsection 2.2 and then define the

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

RFM in subsection 2.3. In subsection 2.4, we describe the optimization principle that leads to implementable algorithms for the RFM and an example problem in which $\mathcal{X}$ and $\mathcal{Y}$ are one-dimensional vector spaces. We finish by providing two convergence results for trained function-valued RFMs in subsection 2.5.

### 2.1. Problem Formulation.

Let $\mathcal{X}$ and $\mathcal{Y}$ be real Banach spaces and $F^\dagger\colon \mathcal{X} \to \mathcal{Y}$ be a (possibly nonlinear) map. It is natural to frame the approximation of $F^\dagger$ as a supervised learning problem. Suppose we are given training data in the form of input-output pairs $\{(a_i, y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$, where $a_i \sim \nu$ are independent and identically distributed (i.i.d.), $\nu$ is a probability measure supported on $\mathcal{X}$, and $y_i$ is given by $F^\dagger(a_i) \sim F^\dagger_\sharp \nu$ plus, potentially, noise. In the examples in this paper, the noise is viewed as resulting from model error (the PDE does not perfectly represent the physics) or from discretization error (in approximating the PDE); situations in which the data acquisition process is inherently noisy can also be envisioned [92] but are not explicitly studied here. We aim to build a parametric reconstruction of the true map $F^\dagger$ from the data; that is, construct a model $F\colon \mathcal{X} \times \mathcal{P} \to \mathcal{Y}$ and find $\alpha^\dagger \in \mathcal{P} \subseteq \mathbb{R}^m$ such that $F(\,\cdot\,, \alpha^\dagger) \approx F^\dagger$ are close as maps from $\mathcal{X}$ to $\mathcal{Y}$ in some suitable sense. The natural number $m$ here denotes the total number of model parameters.

The standard approach to determining parameters in supervised learning is to define a loss functional $\ell\colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ and minimize the expected risk,

$$\min_{\alpha \in \mathcal{P}} \mathbb{E}^{a \sim \nu}\Big[\ell\big(F^\dagger(a), F(a, \alpha)\big)\Big]\,. \tag{2.2}$$

With only the data $\{(a_i, y_i)\}_{i=1}^n$ at our disposal, we approximate problem (2.2) by replacing $\nu$ with the empirical measure $\nu^{(n)} := \frac{1}{n}\sum_{i=1}^n \delta_{a_i}$, which leads to the empirical risk minimization problem

$$\min_{\alpha \in \mathcal{P}} \frac{1}{n}\sum_{i=1}^n \ell\big(y_i, F(a_i, \alpha)\big)\,. \tag{2.3}$$

The hope is that given minimizer $\alpha^{(n)}$ of (2.3) and $\alpha^\dagger$ of (2.2), $F(\,\cdot\,, \alpha^{(n)})$ well approximates $F(\,\cdot\,, \alpha^\dagger)$, that is, the learned model *generalizes* well; these ideas may be made rigorous with results from statistical learning theory [68]. Solving problem (2.3) is called *training* the model $F$. Once trained, the model is then validated on a new set of i.i.d. input-output pairs previously unseen during the training process. This *testing* phase indicates how well $F$ approximates $F^\dagger$. In what follows, we assume that $(\mathcal{Y}, \langle\cdot,\cdot\rangle_{\mathcal{Y}}, \|\cdot\|_{\mathcal{Y}})$ is a real separable Hilbert space and focus on the squared loss

$$\ell(y, y') := \frac{1}{2}\|y - y'\|_{\mathcal{Y}}^2\,. \tag{2.4}$$

We stress that our entire formulation is in an infinite-dimensional setting and we will remain in this setting throughout the paper; as such, the random feature methodology we propose will inherit desirable discretization-invariant properties, to be observed in the numerical experiments of section 4.

*Notation* 2.1 (expectation). For a Borel measurable map $G\colon \mathcal{U} \to \mathcal{V}$ between two Banach spaces $\mathcal{U}$ and $\mathcal{V}$ and a probability measure $\pi$ supported on $\mathcal{U}$, we denote the expectation of $G$ under $\pi$ by

$$\mathbb{E}^{u \sim \pi}\big[G(u)\big] = \int_{\mathcal{U}} G(u)\pi(du) \in \mathcal{V} \tag{2.5}$$

in the sense of Bochner integration (see, e.g., [38, sect. A.2]).

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

### 2.2. Operator-Valued Reproducing Kernels.

The RFM is naturally formulated in an RKHS setting, as our exposition will demonstrate in subsection 2.3. However, the usual RKHS theory is concerned with real-valued functions [5, 16, 37, 146]. Our setting, with the output space $\mathcal{Y}$ a separable Hilbert space, requires several ideas that generalize the real-valued case. We now outline these ideas with a review of operator-valued kernels; parts of the presentation that follow may be found in the references [7, 28, 107, 112].

We first consider the special case $\mathcal{Y} := \mathbb{R}$ for ease of exposition. A real RKHS is a Hilbert space $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}}, \|\cdot\|_{\mathcal{H}})$ comprising real-valued functions $f\colon \mathcal{X} \to \mathbb{R}$ such that the pointwise evaluation functional $f \mapsto f(a)$ is bounded for every $a \in \mathcal{X}$. It then follows that there exists a unique, symmetric, positive definite kernel function $k\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ such that for every $a \in \mathcal{X}$, we have $k(\cdot, a) \in \mathcal{H}$ and the *reproducing kernel property* $f(a) = \langle k(\cdot, a), f\rangle_{\mathcal{H}}$ holds. These two properties are often taken as the definition of an RKHS. The converse direction is also true: every symmetric, positive definite kernel defines a unique RKHS [5].

We now introduce the necessary generalization of the reproducing property to the case of arbitrary real Hilbert spaces $\mathcal{Y}$, as this result will motivate the construction of the RFM. Kernels in this setting are now operator-valued.

Definition 2.2 (operator-valued kernel). *Let $\mathcal{X}$ be a real Banach space and $\mathcal{Y}$ a real separable Hilbert space. An* operator-valued kernel *is a map*

$$k\colon \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y}), \tag{2.6}$$

*where $\mathcal{L}(\mathcal{Y})$ denotes the Banach space of all bounded linear operators on $\mathcal{Y}$, such that its adjoint satisfies $k(a, a')^* = k(a', a)$ for all $a$ and $a'$ in $\mathcal{X}$ and, for every $N \in \mathbb{N}$,*

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\big\langle y_i, k(a_i, a_j)y_j\big\rangle_{\mathcal{Y}} \geq 0 \tag{2.7}$$

*for all pairs $\{(a_i, y_i)\}_{i=1}^N \subset \mathcal{X} \times \mathcal{Y}$.*

Paralleling the development for the real-valued case, an operator-valued kernel $k$ also uniquely (up to isomorphism) determines an associated real RKHS $\mathcal{H}_k = \mathcal{H}_k(\mathcal{X};\mathcal{Y})$ of operators mapping $\mathcal{X}$ to $\mathcal{Y}$. Now, choosing a probability measure $\nu$ supported on $\mathcal{X}$, we define a kernel integral operator $T_k$ associated to $k$ by

$$\begin{aligned} T_k\colon L^2_\nu(\mathcal{X};\mathcal{Y}) &\to L^2_\nu(\mathcal{X};\mathcal{Y}), \\ F &\mapsto T_kF := \mathbb{E}^{a'\sim\nu}\big[k(\cdot, a')F(a')\big], \end{aligned} \tag{2.8}$$

which is nonnegative, self-adjoint, and compact (provided $k(a, a) \in \mathcal{L}(\mathcal{Y})$ is compact for all $a \in \mathcal{X}$ [28]). Let us further assume that all conditions needed for $T_k^{1/2}$ to be an isometry from $L^2_\nu$ into $\mathcal{H}_k$ are satisfied, i.e., $\mathcal{H}_k = \operatorname{im}(T_k^{1/2})$. Generalizing the standard Mercer theory (see, e.g., [7, 16]), we may write the RKHS inner product as

$$\langle F, G\rangle_{\mathcal{H}_k} = \langle F, T_k^{-1}G\rangle_{L^2_\nu} \quad \text{for all} \quad F \in \mathcal{H}_k \quad \text{and} \quad G \in \mathcal{H}_k\,. \tag{2.9}$$

Note that while (2.9) appears to depend on the measure $\nu$ on $\mathcal{X}$, the set $\mathcal{H}_k$ itself is determined by the kernel without any reference to a measure [37, Chap. 3, Thm. 4]. With the inner product now explicit, it is possible to deduce the following reproducing property of the operator-valued kernel $k$ [111, sect. 2.2].

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

RESULT 2.3 (reproducing property for operator-valued kernels). *Let $F \in \mathcal{H}_k$ be given. For every $a \in \mathcal{X}$ and $y \in \mathcal{Y}$, it holds that*

$$\langle y, F(a)\rangle_{\mathcal{Y}} = \langle k(\cdot, a)y, F\rangle_{\mathcal{H}_k} . \tag{2.10}$$

The identity (2.10), paired with a special choice of operator-valued kernel $k$, is the basis of the RFM in our abstract infinite-dimensional setting.

### 2.3. Random Feature Model.

One could approach the approximation of target map $F^\dagger\colon \mathcal{X} \to \mathcal{Y}$ from the perspective of kernel methods. However, it is generally a difficult task to explicitly design operator-valued kernels of the form (2.6) since the spaces $\mathcal{X}$ and $\mathcal{Y}$ may be of different regularity, for example. Example constructions of operator-valued kernels studied in the literature include those taking value as diagonal operators, multiplication operators, or composition operators [77, 107, 116], but these all involve some simple generalization of scalar-valued kernels or strong assumptions about $\mathcal{Y}$. Instead, the RFM allows one to implicitly work with fully general operator-valued kernels through the use of a *random feature map* $\varphi\colon \mathcal{X} \times \Theta \to \mathcal{Y}$ and a probability measure $\mu$ supported on Banach space $\Theta$. The map $\varphi$ is assumed to be square integrable w.r.t. the product measure $\nu \times \mu$, i.e., $\varphi \in L^2_{\nu\times\mu}(\mathcal{X} \times \Theta; \mathcal{Y})$, where $\nu$ is the (sometimes a modeling choice at our discretion, sometimes unknown) data distribution on $\mathcal{X}$. Together, $(\varphi, \mu)$ form a *random feature pair*. With this setup in place, we now describe the connection between random features and kernels. To this end, recall the following standard notation.

*Notation* 2.4 (outer product). Given a Hilbert space $(H, \langle\cdot,\cdot\rangle, \|\cdot\|)$, the *outer product* $a \otimes b \in \mathcal{L}(H)$ is defined by $(a \otimes b)c = \langle b, c\rangle a$ for any $a$, $b$, and $c \in H$.

#### 2.3.1. An Intractable Nonparametric Model Class.

Given the pair $(\varphi, \mu)$, we begin by considering maps $k_\mu\colon \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y})$ of the form

$$k_\mu(a, a') := \mathbb{E}^{\theta\sim\mu}\bigl[\varphi(a;\theta) \otimes \varphi(a';\theta)\bigr] . \tag{2.11}$$

Such representations need not be unique; different pairs $(\varphi, \mu)$ may induce the same kernel $k = k_\mu$ in (2.11). Since $k_\mu$ may readily be shown to be an operator-valued kernel via Definition 2.2, it defines a unique real RKHS $\mathcal{H}_{k_\mu} \subset L^2_\nu(\mathcal{X}; \mathcal{Y})$. Our methodology will be based on this space and, in particular, finite-dimensional approximations thereof.

We now perform a purely formal but instructive calculation, following from application of the reproducing property (2.10) to operator-valued kernels of the form (2.11). Doing so leads to an integral representation of any $F \in \mathcal{H}_{k_\mu}$. For all $a \in \mathcal{X}$ and $y \in \mathcal{Y}$, it holds that

$$\begin{aligned}
\langle y, F(a)\rangle_{\mathcal{Y}} = \langle k_\mu(\cdot, a)y, F\rangle_{\mathcal{H}_{k_\mu}} &= \Bigl\langle \mathbb{E}^{\theta\sim\mu}\bigl[\langle \varphi(a;\theta), y\rangle_{\mathcal{Y}}\, \varphi(\,\cdot\,;\theta)\bigr], F \Bigr\rangle_{\mathcal{H}_{k_\mu}}\\
&= \mathbb{E}^{\theta\sim\mu}\Bigl[\langle \varphi(a;\theta), y\rangle_{\mathcal{Y}} \langle \varphi(\,\cdot\,;\theta), F\rangle_{\mathcal{H}_{k_\mu}}\Bigr]\\
&= \mathbb{E}^{\theta\sim\mu}\bigl[c_F(\theta)\langle y, \varphi(a;\theta)\rangle_{\mathcal{Y}}\bigr]\\
&= \Bigl\langle y, \mathbb{E}^{\theta\sim\mu}\bigl[c_F(\theta)\varphi(a;\theta)\bigr]\Bigr\rangle_{\mathcal{Y}} ,
\end{aligned}$$

where the coefficient function $c_F\colon \Theta \to \mathbb{R}$ is defined by

$$\theta \mapsto c_F(\theta) := \langle \varphi(\,\cdot\,;\theta), F\rangle_{\mathcal{H}_{k_\mu}} . \tag{2.12}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Since $\mathcal{Y}$ is Hilbert, the above holding for all $y \in \mathcal{Y}$ implies the integral representation

$$F = \mathbb{E}^{\theta\sim\mu}\big[c_F(\theta)\varphi(\,\cdot\,;\theta)\big]\,. \tag{2.13}$$

The formal expression (2.12) for $c_F(\theta)$ needs careful interpretation, which is provided in [111, App. B] of the original version of this paper. For instance, if $\varphi(\,\cdot\,;\theta)$ is chosen to be a realization of a Gaussian process (as seen later in Example 2.9), then $\varphi(\,\cdot\,;\theta) \notin \mathcal{H}_{k_\mu}$ with probability 1; indeed, in this case $c_F$ is defined only as an $L^2_\mu(\Theta;\mathbb{R})$ limit. Nonetheless, the RKHS may be completely characterized by this integral representation. Define the map

$$\begin{aligned} \mathcal{A}\colon L^2_\mu(\Theta;\mathbb{R}) &\to L^2_\nu(\mathcal{X};\mathcal{Y})\,,\\ c &\mapsto \mathcal{A}c \coloneqq \mathbb{E}^{\theta\sim\mu}\big[c(\theta)\varphi(\,\cdot\,;\theta)\big]\,. \end{aligned} \tag{2.14}$$

The map $\mathcal{A}$ may be shown to be a bounded linear operator that is a particular square root of $T_{k_\mu}$ from (2.8) [111, App. B]. We have the following result whose proof, provided in the original version of this paper [111, App. A], is a straightforward generalization of the real-valued case given in [7, sect. 2.2].

Result 2.5 (infinite-dimensional RKHS). *Under the assumption that the feature map $\varphi$ satisfies $\varphi \in L^2_{\nu\times\mu}(\mathcal{X}\times\Theta;\mathcal{Y})$, the RKHS defined by the kernel $k_\mu$ in* (2.11) *is*

$$\mathcal{H}_{k_\mu} = \operatorname{im}(\mathcal{A}) = \Big\{\mathbb{E}^{\theta\sim\mu}\big[c(\theta)\varphi(\,\cdot\,;\theta)\big] : c \in L^2_\mu(\Theta;\mathbb{R})\Big\}\,. \tag{2.15}$$

We stress that the integral representation of mappings in RKHS (2.15) is not unique, since $\mathcal{A}$ is not injective in general. However, the particular choice $c = c_F$ (2.12) in representation (2.13) does enjoy a sense of uniqueness as described in [111, App. B]. In particular, the $L^2_\mu(\Theta;\mathbb{R})$ norm of $c_F$ equals the $\mathcal{H}_{k_\mu}$ norm of $F$. The formula (2.15) suggests that $\mathcal{H}_{k_\mu}$, which is built from $(\varphi,\mu)$ and completely determined by coefficient functionals $c \in L^2_\mu(\Theta;\mathbb{R})$, is a natural nonparametric class of operators with which to perform approximation. However, the actual implementation of estimators based on the model class $\mathcal{H}_{k_\mu}$ is known to incur enormous computational cost without further assumptions on the structure of $(\varphi,\mu)$, as we discuss later in this section. Instead, we next adopt a parametric approximation to this full RKHS approach.

### 2.3.2. A Tractable Parametric Model Class.

A central role in what follows is the approximation of measure $\mu$ by the empirical measure

$$\mu^{(m)} \coloneqq \frac{1}{m}\sum_{j=1}^{m}\delta_{\theta_j}\,, \quad \text{where} \quad \theta_j \overset{\text{iid}}{\sim} \mu\,. \tag{2.16}$$

Given (2.16), define $k^{(m)} \coloneqq k_{\mu^{(m)}}$ to be the empirical approximation to $k_\mu$, that is,

$$k^{(m)}(a,a') = \mathbb{E}^{\theta\sim\mu^{(m)}}\big[\varphi(a;\theta)\otimes\varphi(a';\theta)\big] = \frac{1}{m}\sum_{j=1}^{m}\varphi(a;\theta_j)\otimes\varphi(a';\theta_j)\,. \tag{2.17}$$

We then let $\mathcal{H}_{k^{(m)}}$ be the unique RKHS induced by the kernel $k^{(m)}$; note that $k^{(m)}$ and hence $\mathcal{H}_{k^{(m)}}$ are themselves random. The following characterization of $\mathcal{H}_{k^{(m)}}$ is proved in the original version of this paper [111, App. A].

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

RESULT 2.6 (finite-dimensional RKHS). *Assume that* $\varphi \in L^2_{\nu\times\mu}(\mathcal{X}\times\Theta;\mathcal{Y})$ *and that the random features* $\{\varphi(\,\cdot\,;\theta_j)\}_{j=1}^m$ *are linearly independent in* $L^2_\nu(\mathcal{X};\mathcal{Y})$. *Then the RKHS* $\mathcal{H}_{k^{(m)}}$ *is equal to the linear span of* $\{\varphi(\,\cdot\,;\theta_j)\}_{j=1}^m$.

Applying a simple Monte Carlo sampling approach to elements in RKHS (2.15) by replacing probability measure $\mu$ by empirical measure $\mu^{(m)}$ gives the intuition that

$$\frac{1}{m}\sum_{j=1}^m c(\theta_j)\varphi(\,\cdot\,;\theta_j)\approx \mathbb{E}^{\theta\sim\mu}\bigl[c(\theta)\varphi(\,\cdot\,;\theta)\bigr] \quad \text{for} \quad c\in L^2_\mu(\Theta;\mathbb{R})\,. \tag{2.18}$$

This low-rank approximation achieves the Monte Carlo rate $O(m^{-1/2})$ in expectation and, by virtue of Result 2.6, is in $\mathcal{H}_{k^{(m)}}$. However, in the setting of this work, the Monte Carlo approach does not give rise to a practical method for learning a target map $F^\dagger\in\mathcal{H}_{k_\mu}$ because $F^\dagger$, $k_\mu$, and $\mathcal{H}_{k_\mu}$ are all unknown; only the random feature pair $(\varphi,\mu)$ is assumed to be given. Hence one cannot apply (2.12) or [111, eq. (B.2), p. A3239] to evaluate $c=c_{F^\dagger}$ in (2.18). Furthermore, in realistic settings it may be that $F^\dagger\notin\mathcal{H}_{k_\mu}$, which leads to an additional smoothness misspecification gap not accounted for by the Monte Carlo method. To sidestep these difficulties, the RFM adopts a data-driven optimization approach to determine a different estimator of $F^\dagger$, also from the space $\mathcal{H}_{k^{(m)}}$. We now define the RFM.

DEFINITION 2.7 (RFM). *Given probability spaces* $(\mathcal{X},\mathcal{B}(\mathcal{X}),\nu)$ *and* $(\Theta,\mathcal{B}(\Theta),\mu)$, *with* $\mathcal{X}$ *and* $\Theta$ *being real finite- or infinite-dimensional Banach spaces, a real separable Hilbert space* $\mathcal{Y}$, *and* $\varphi\in L^2_{\nu\times\mu}(\mathcal{X}\times\Theta;\mathcal{Y})$, *the* RFM *is the parametric map*

$$\begin{aligned} &F_m\colon \mathcal{X}\times\mathbb{R}^m\to\mathcal{Y}\,,\\ &(a;\alpha)\mapsto F_m(a;\alpha):=\frac{1}{m}\sum_{j=1}^m \alpha_j\varphi(a;\theta_j)\,,\quad \textit{where}\quad \theta_j\overset{\text{iid}}{\sim}\mu\,. \end{aligned} \tag{2.19}$$

We use the Borel $\sigma$-algebras $\mathcal{B}(\mathcal{X})$ and $\mathcal{B}(\Theta)$ to define the probability spaces in the preceding definition. Our goal with the RFM is to choose parameters $\alpha\in\mathbb{R}^m$ to approximate mappings $F^\dagger\in\mathcal{H}_{k_\mu}$ (in the well-specified setting) by mappings $F_m(\cdot;\alpha)\in\mathcal{H}_{k^{(m)}}$. The RFM is itself random and may be viewed as a *spectral method* because the randomized family $\{\varphi(\,\cdot\,;\theta)\}$ in the linear expansion (2.19) is defined $\nu$-almost everywhere on $\mathcal{X}$. Determining the coefficient vector $\alpha$ from data obviates the difficulties associated with the oracle Monte Carlo approach because the data-driven method only requires knowledge of the pair $(\varphi,\mu)$ and knowledge of sample input-output pairs from target operator $F^\dagger$.

As written, (2.19) is incredibly simple. The operator $F_m$ is nonlinear in its input $a$ but linear in its coefficient parameters $\alpha$. In practice, the linearity w.r.t. the RFM parameters is broken by also learning *hyperparameters* that appear in the pair $(\varphi,\mu)$. Moreover, similar to operator learning architectures such as neural operators [87] and Fourier neural operators [96], the RFM is a *nonlinear approximation.* This means that the output $F_m(a;\alpha)$ of the RFM belongs to a nonlinear manifold in $\mathcal{Y}$ (cf. (2.1)) instead of a fixed linear subspace of $\mathcal{Y}$. In contrast, methods such as PCA-Net [19] and DeepONet [101] are restricted to such fixed linear spaces, which may limit their approximation power for specific classes of problems. More theory is required to quantitatively separate these two classes of approximation method.

Overall, it is clear that the choice of random feature map and measure pair $(\varphi,\mu)$ will determine the quality of approximation. Most papers deploying these methods,

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

including [24, 122, 123], take a kernel-oriented perspective by first choosing a kernel and then finding a random feature map to estimate this kernel. Our perspective, more aligned with [124, 140], is the opposite in that we allow the choice of random feature map $\varphi$ and distribution $\mu$ to implicitly *define* the kernel via the formula (2.11), instead of picking the kernel first. This viewpoint also has implications for numerics: the kernel never explicitly appears in any computations, which leads to memory and other cost savings. It does, however, leave open the question of characterizing the universality [140] of such kernels and the RKHS $\mathcal{H}_{k_\mu}$ of mappings from $\mathcal{X}$ to $\mathcal{Y}$ that underlies the approximation method; this is an important avenue for future work.

**2.3.3. Connection to Neural Networks and Neural Operators.** The close connection to kernels explains the origins of the RFM in the machine learning literature. Moreover, the RFM may also be interpreted in the context of NNs [110, 140, 147, 151]. To see this explicitly, consider the setting in which $\mathcal{X}$ and $\mathcal{Y}$ are both equal to the Euclidean space $\mathbb{R}$ and choose $\varphi$ to be a family of hidden neurons of the form $\varphi_{\mathrm{NN}}(a;\theta) := \sigma(\theta^{(1)} \cdot a + \theta^{(2)})$, where $\sigma(\,\cdot\,)$ is a nonlinear activation function. A single hidden layer NN would seek to find $\{(\alpha_j, \theta_j)\}_{j=1}^m \subset \mathbb{R} \times \mathbb{R}^2$ such that

$$\frac{1}{m}\sum_{j=1}^{m} \alpha_j \varphi_{\mathrm{NN}}(\,\cdot\,;\theta_j) \tag{2.20}$$

matches the given training data $\{(a_i, y_i)\}_{i=1}^n \subset \mathcal{X}\times\mathcal{Y}$. More generally, and in arbitrary Euclidean spaces, one may allow $\varphi_{\mathrm{NN}}(\,\cdot\,;\theta)$ to be any deep NN. However, while the RFM has the same *form* as (2.20), there is a difference in the *training*: the $\theta_j$ are drawn i.i.d. from a probability measure and then fixed, and only the $\alpha_j$ are chosen to fit the training data. This idea immediately transfers to the operator learning setting in which $\mathcal{X}$ and $\mathcal{Y}$ are function spaces and the maps $\varphi_{\mathrm{NN}}(\,\cdot\,;\theta)$ are themselves randomly initialized deep neural operators or DeepONets. Given any deep NN with randomly initialized parameters, studies of the lazy training regime and neural tangent kernel [26, 75] suggest that adopting an RFM approach and only optimizing over the last layer weights $\alpha$ is quite natural. Indeed, it is observed that in this regime the internal NN parameters do not stray far from their random initialization during gradient descent, while the last layer of parameters $\{\alpha_j\}_{j=1}^m$ adapts considerably.

Once the feature parameters $\{\theta_j\}_{j=1}^m$ are sampled at random and fixed, training the RFM $F_m$ only requires optimizing over $\alpha \in \mathbb{R}^m$. Due to the linearity of $F_m$ in $\alpha$, this is a straightforward task that we now describe.

**2.4. Optimization.** One of the most attractive characteristics of the RFM is its training procedure. With the $L^2$-type loss (2.4) as in standard regression settings, optimizing the coefficients of the RFM w.r.t. the empirical risk (2.3) is a convex optimization problem, requiring only the solution of a finite-dimensional system of linear equations; the convexity also suggests the possibility of appending convex constraints (such as linear inequalities), although we do not pursue this here. Further, the kernels $k_\mu$ or $k^{(m)}$ are not required anywhere in the algorithm. We emphasize the simplicity of the underlying optimization tasks as they suggest the possibility of numerical implementation of the RFM in complicated black-box computer codes. This is in contrast with other methods such as deep neural operators, which are trained with variants of stochastic gradient descent. Such a training strategy leads to nonconvexity that is notoriously difficult to study, both computationally and theoretically.

We now show that a regularized version of the quadratic optimization problem (2.3)–(2.4) arises naturally from approximation of a nonparametric regression

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

problem defined over the RKHS $\mathcal{H}_{k_\mu}$. To this end, recall the supervised learning formulation in subsection 2.1. Given $n$ i.i.d. input-output pairs $\{(a_i, y_i)\}_{i=1}^n \subset \mathcal{X} \times \mathcal{Y}$ as data, with the $a_i$ drawn from (possibly unknown) probability measure $\nu$ on $\mathcal{X}$ and $y_i = F^\dagger(a_i)$, the objective is to find an approximation $\widehat{F}$ to the map $F^\dagger$. Let $\mathcal{H}_{k_\mu}$ be the hypothesis space and $k_\mu$ its operator-valued reproducing kernel of the form (2.11). The most straightforward learning algorithm in this RKHS setting is kernel ridge regression, also known as penalized least squares. This method produces a nonparametric model by finding a minimizer $\widehat{F}$ of

$$\min_{F \in \mathcal{H}_{k_\mu}} \left\{ \sum_{i=1}^n \frac{1}{2} \|y_i - F(a_i)\|_{\mathcal{Y}}^2 + \frac{\lambda}{2} \|F\|_{\mathcal{H}_{k_\mu}}^2 \right\}, \tag{2.21}$$

where $\lambda \geq 0$ is a penalty parameter. By the representer theorem for operator-valued kernels [107, Thms. 2 and 4], the minimizer has the form

$$\widehat{F} = \sum_{i=1}^n k_\mu(\cdot, a_i)\beta_i \tag{2.22}$$

for some functions $\{\beta_i\}_{i=1}^n \subset \mathcal{Y}$. In practice, finding these $n$ functions in the output space requires solving a block linear operator equation. For the high-dimensional PDE problems we consider in this work, solving such an equation may become prohibitively expensive due to both operation count and memory required. A few workarounds were proposed in [77] such as certain diagonalizations, but these rely on simplifying assumptions that are somewhat limiting. More fundamentally, the representation of the solution in (2.22) requires knowledge of the kernel $k_\mu$; in our setting we assume access only to the random feature pair $(\varphi, \mu)$, which defines $k_\mu$ and not $k_\mu$ itself.

We thus explain how to make progress with this problem given knowledge only of random features. Recall the empirical kernel given by (2.17), the RKHS $\mathcal{H}_{k^{(m)}}$, and Result 2.6. The following result, proved in [111, App. A], shows that an RFM hypothesis class with a penalized least squares empirical loss function in optimization problem (2.3)–(2.4) is equivalent to kernel ridge regression (2.21) restricted to $\mathcal{H}_{k^{(m)}}$.

Result 2.8 (random feature ridge regression is equivalent to a kernel method). *Assume that $\varphi \in L^2_{\nu\times\mu}(\mathcal{X} \times \Theta; \mathcal{Y})$ and that the random features $\{\varphi(\,\cdot\,; \theta_j)\}_{j=1}^m$ are linearly independent in $L^2_\nu(\mathcal{X}; \mathcal{Y})$. Fix $\lambda \geq 0$. Let $\widehat{\alpha} \in \mathbb{R}^m$ be the unique minimum norm solution of*

$$\min_{\alpha \in \mathbb{R}^m} \left\{ \sum_{i=1}^n \frac{1}{2} \left\| y_i - \frac{1}{m} \sum_{l=1}^m \alpha_l \varphi(a_i; \theta_l) \right\|_{\mathcal{Y}}^2 + \frac{\lambda}{2m} \|\alpha\|_2^2 \right\}. \tag{2.23}$$

*Then the RFM defined by this choice of $\alpha = \widehat{\alpha}$ satisfies*

$$F_m(\,\cdot\,; \widehat{\alpha}) = \operatorname*{argmin}_{F \in \mathcal{H}_{k^{(m)}}} \left\{ \sum_{i=1}^n \frac{1}{2} \|y_i - F(a_i)\|_{\mathcal{Y}}^2 + \frac{\lambda}{2} \|F\|_{\mathcal{H}_{k^{(m)}}}^2 \right\}. \tag{2.24}$$

Solving the convex problem (2.23) trains the RFM. The first order condition for a global minimizer leads to the normal equations

$$\sum_{j=1}^m \left( \frac{1}{m} \sum_{i=1}^n \langle \varphi(a_i; \theta_l), \varphi(a_i; \theta_j) \rangle_{\mathcal{Y}} + \lambda \delta_{lj} \right) \alpha_j = \sum_{i=1}^n \langle \varphi(a_i; \theta_l), y_i \rangle_{\mathcal{Y}} \tag{2.25}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

for each $l \in \{1, \ldots, m\}$, where $\delta_{lj} = 1$ if $l = j$, and equals zero otherwise. This is an $m$-by-$m$ linear system of equations for $\alpha \in \mathbb{R}^m$ that is standard to solve. In the case $\lambda = 0$, the minimum norm solution of (2.25) may be written in terms of a pseudoinverse operator (see [102, sect. 6.11]).

Equation (2.25) reveals that the trained RFM $F_m(\,\cdot\,; \widehat{\alpha})$ is a linear function of the labeled output data $\{y_i\}_{i=1}^n$. This property is undesirable from the perspective of statistical optimality. Indeed, it is known that any estimator that is linear in the output training data is minimax *suboptimal* for certain classes of problems [141, Thm. 1, sect. 4.1, p. 6]. However, any adaptation of the feature pair $(\varphi, \mu)$ to the training data will break this property and potentially restore optimality. For example, choosing $\lambda$ or hyperparameters appearing in $(\varphi, \mu)$ based on a cross-validation procedure would make the RF pair data-dependent as desired. This is typically done in practice.

*Example* 2.9 (Brownian bridge). We now provide a one-dimensional instantiation of the RFM to illustrate the methodology. Take the input space as $\mathcal{X} := (0, 1)$, output space as $\mathcal{Y} := \mathbb{R}$, input space measure $\nu := \mathsf{Unif}(0, 1)$ to be uniform, and random parameter space as $\Theta := \mathbb{R}^\infty$. Denote the input by $a = x \in \mathcal{X}$. Then, consider the random feature map $\varphi \colon (0, 1) \times \mathbb{R}^\infty \to \mathbb{R}$ defined by the *Brownian bridge*

$$\varphi(x; \theta) := \sum_{j=1}^{\infty} \theta^{(j)} (j\pi)^{-1} \sqrt{2} \sin(j\pi x)\,, \quad \text{where} \quad \theta^{(j)} \overset{\text{iid}}{\sim} N(0, 1)\,, \tag{2.26}$$

where $\theta := \{\theta^{(j)}\}_{j \in \mathbb{N}}$ and $\mu := N(0, 1) \times N(0, 1) \times \cdots$. For any realization of $\theta \sim \mu$, the function $\varphi(\,\cdot\,; \theta)$ is a Brownian motion constrained to zero at $x = 0$ and $x = 1$. The induced kernel $k_\mu \colon (0, 1) \times (0, 1) \to \mathbb{R}$ is then simply the covariance function of this stochastic process:

$$k_\mu(x, x') = \mathbb{E}^{\theta \sim \mu}\big[\varphi(x; \theta)\varphi(x'; \theta)\big] = \min\{x, x'\} - xx'\,. \tag{2.27}$$

Note that $k_\mu$ is the Green's function for the negative Laplacian on $(0, 1)$ with Dirichlet boundary conditions. Using this fact, we may explicitly characterize the associated RKHS $\mathcal{H}_{k_\mu}$ as follows. First, we have

$$T_{k_\mu} f = \int_0^1 k_\mu(\cdot, y) f(y)\, dy = \left(-\frac{d^2}{dx^2}\right)^{-1} f\,, \tag{2.28}$$

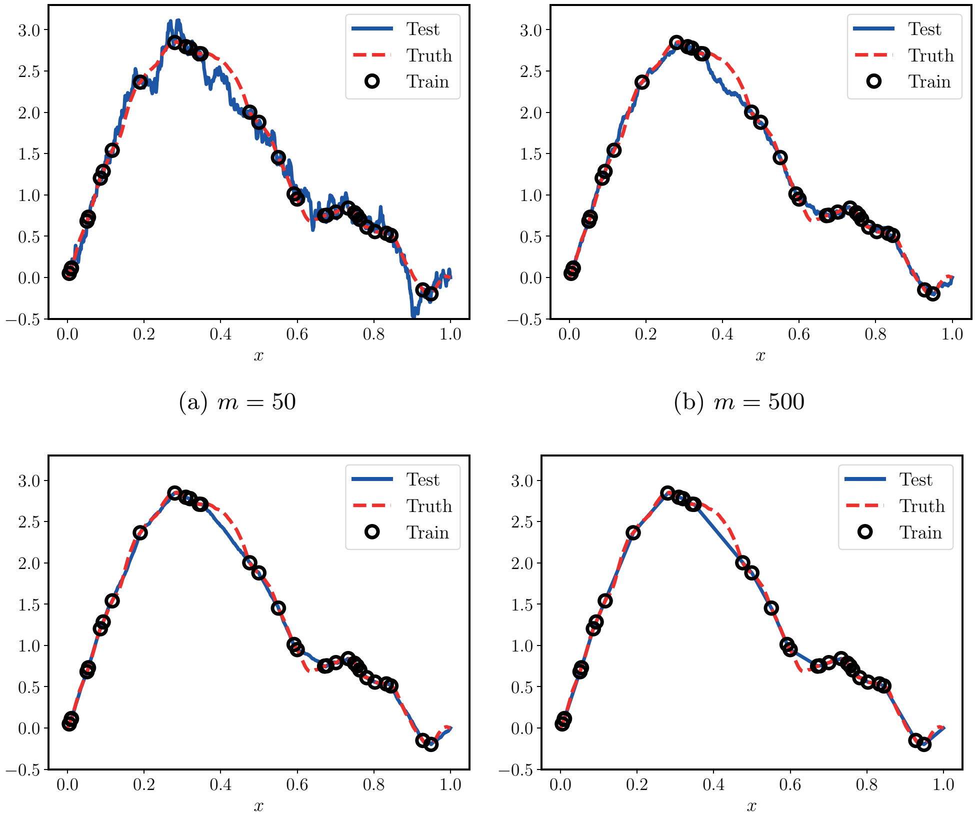
where the negative Laplacian has domain $H^2((0, 1); \mathbb{R}) \cap H_0^1((0, 1); \mathbb{R})$. Viewing $T_{k_\mu}$ as an operator from $L^2((0, 1); \mathbb{R})$ into itself, from (2.9) we conclude, upon integration by parts, that for any elements $f$ and $g$ of $\mathcal{H}_{k_\mu}$, it holds that

$$\langle f, g \rangle_{\mathcal{H}_{k_\mu}} = \langle f, T_{k_\mu}^{-1} g \rangle_{L^2} = \left\langle \frac{df}{dx}, \frac{dg}{dx} \right\rangle_{L^2} = \langle f, g \rangle_{H_0^1}\,. \tag{2.29}$$

Note that the last identity does indeed define an inner product on $H_0^1$. By this formal argument we identify the RKHS $\mathcal{H}_{k_\mu}$ as the Sobolev space $H_0^1((0, 1); \mathbb{R})$. Furthermore, the Brownian bridge may be viewed as the Gaussian measure $N(0, T_{k_\mu})$. Approximation using the RFM with the Brownian bridge random features is illustrated in Figure 1. Since $k_\mu(\cdot, x)$ is a piecewise linear function, a kernel interpolation or regression method will produce a piecewise linear approximation. Indeed, the figure indicates that the RFM with $n$ training points fixed approaches the optimal piecewise linear kernel interpolant as $m \to \infty$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy



(a) $m = 50$

(b) $m = 500$

(c) $m = 5000$

(d) $m = \infty$

Fig. 1. *Brownian bridge RFM for one-dimensional input-output spaces with $n = 32$ training points fixed and $\lambda = 0$ (Example* 2.9*): As $m \to \infty$, the RFM approaches the nonparametric interpolant given by the representer theorem* (d)*, which in this case is a piecewise linear approximation of the true function (an element of RKHS $\mathcal{H}_{k_\mu} = H_0^1$, shown in red). Blue lines denote the trained model evaluated on test data points, and black circles denote evaluation at training points.*

The Brownian bridge in Example 2.9 illuminates a more fundamental idea. For this low-dimensional problem, an expansion in a deterministic Fourier sine basis would of course be more natural. However, if we do not have a natural, computable orthonormal basis, then randomness provides a useful alternative representation; notice that the random features each include random combinations of the deterministic Fourier sine basis in this example. For the more complex problems that we study numerically in the next two sections, we lack knowledge of good, computable bases for general maps in infinite dimensions. The RFM approach exploits randomness to explore, implicitly discover the structure of, and represent such maps. Thus we now turn away from this example of real-valued maps defined on a subset of the real line and instead consider the use of random features to represent maps between spaces of functions. It turns out that theoretical guarantees are still possible to obtain in this setting.

**2.5. Error Bounds.** In this subsection, we review a recent comprehensive error analysis [92] of the random feature ridge regression problem (2.24) in the general infinite-dimensional input and output space setting. This is the sharpest available theory for misspecified problems. Owing to its tractable optimization, the RFM

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

is one of the first guaranteed convergent operator learning algorithms for nonlinear problems that is actually implementable on a computer with controlled complexity. To see this more concretely, we require the following technical assumptions.

ASSUMPTION 2.10 (data and features). *The following hold true:*

(i) *The ground truth operator $F^\dagger$ satisfies $F^\dagger \in L^\infty_\nu(\mathcal{X};\mathcal{Y})$.*

(ii) *The noise-free training data are given by $a_i \overset{\text{iid}}{\sim} \nu$ and $y_i = F^\dagger(a_i)$ for each $i$.*

(iii) *The random feature map $\varphi \in L^\infty_{\nu\times\mu}(\mathcal{X}\times\Theta;\mathcal{Y})$ is measurable and bounded.*

(iv) *The RKHS $\mathcal{H}_{k_\mu}$ corresponding to the pair $(\varphi,\mu)$ is separable.*

Our first convergence result is qualitative and follows from [92, Thm. 3.10, p. 6], which itself is a consequence of a more general error estimate [92, Thm. 3.4, pp. 4–5].[1]

THEOREM 2.11 (almost sure convergence of trained RFM). *Let Assumption* 2.10 *hold. Suppose that the integral operator $T_{k_\mu} \in \mathcal{L}(L^2_\nu(\mathcal{X};\mathcal{Y}))$ in* (2.8) *is injective. Let $\{\delta_l\}_{l\in\mathbb{N}} \subset (0,1)$ be any positive sequence with the property that $\sum_{l=1}^\infty \delta_l < \infty$. For $l \in \mathbb{N}$, denote by $\widehat{\alpha}^{(l)} \in \mathbb{R}^{m_l}$ the trained RFM coefficients corresponding to* (2.23) *with $m = m_l$ random features, $n = n_l$ training samples, and regularization parameter $\lambda = \lambda_l$. If*

$$m_l \simeq \delta_l^{-1}\log(2/\delta_l)\,, \quad n_l \simeq \delta_l^{-2}\log(2/\delta_l)\,, \quad \textit{and} \quad \lambda_l \simeq m_l\,, \tag{2.30}$$

*then the trained RFM satisfies*

$$\mathbb{P}\left\{\lim_{l\to\infty} \mathbb{E}^{a\sim\nu}\big\|F^\dagger(a) - F_{m_l}(a;\widehat{\alpha}^{(l)})\big\|^2_{\mathcal{Y}} = 0\right\} = 1\,. \tag{2.31}$$

The probability in (2.31) is w.r.t. the joint law of the data $\{a_i\}_{i=1}^n \sim \nu^{\otimes n}$ and the feature parameters $\{\theta_j\}_{j=1}^M \sim \mu^{\otimes m}$. Going beyond the existence of an accurate approximation to $F^\dagger$, Theorem 2.11 shows that the random feature ridge regression algorithm delivers a strongly consistent statistical estimator of $F^\dagger$ in the limit of large $m$, $n$, and $\lambda$. That is, the trained RFM that one actually obtains in practice converges (along a subsequence w.r.t. $n$) to the true underlying operator $F^\dagger$ with probability 1. The three quantities $m$, $n$, and $\lambda$ are linked via a summable sequence $\{\delta_l\}$, which determines how they are simultaneously sent to infinity. The conditions of the theorem are satisfied with $\delta_l = l^{-2} \to 0$, for example.

The next theorem delivers a high probability nonasymptotic error bound that includes both parameter and sample complexity contributions that depend only algebraically on the reciprocal of the error, instead of exponentially [86, sect. 5]. It is a consequence of [92, Thm. 3.7, p. 5] and controls sources of error due to regularization, finite parametrization, finite data, and optimization.

THEOREM 2.12 (complexity bounds for trained RFM). *Let $\varepsilon \in (0,1)$ be an arbitrary error tolerance. Let $\widehat{\alpha} \in \mathbb{R}^m$ denote the trained RFM coefficients from* (2.23) *with training sample size $n \in \mathbb{N}$ and regularization parameter $\lambda \in (0,n)$. Suppose that $F^\dagger$ belongs to the RKHS $\mathcal{H}_{k_\mu}$ corresponding to the random feature pair $(\varphi,\mu)$. Under Assumption* 2.10, *there exists an absolute constant $C > 0$ such that if*

$$m \geq 11\varepsilon^{-2}\,, \quad n \geq 10\varepsilon^{-4}\,, \quad \textit{and} \quad \lambda \leq 10\varepsilon^{-2}\,, \tag{2.32}$$

[1] The regularization parameter $\lambda$ in Theorem 2.11 and subsection 2.4 is equal to $n$ times the regularization parameter that is discussed in [92], which is also denoted by the same symbol.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

*then the trained RFM $F_m(\,\cdot\,;\widehat{\alpha})$ satisfies the high probability $L^2_\nu(\mathcal{X};\mathcal{Y})$ error bound*

$$\mathbb{P}\left\{\sqrt{\mathbb{E}^{a\sim\nu}\big\|F^\dagger(a)-F_m(a;\widehat{\alpha})\big\|^2_{\mathcal{Y}}}\leq\big(C\|F^\dagger\|_{\mathcal{H}_{k_\mu}}\big)\varepsilon\right\}\geq 0.999\,. \tag{2.33}$$

The takeaway from Theorem 2.12 is that, up to constant factors, an appropriately tuned regularization parameter $\lambda\simeq\sqrt{n}$ and number of random features $m\simeq\sqrt{n}$ are enough to guarantee a trained RFM generalization error of size $n^{-1/4}\simeq m^{-1/2}$ with high probability. However, this quantitative result is dependent on the well-specified condition $F^\dagger\in\mathcal{H}_{k_\mu}$, which is quite difficult to verify in practice. It would be interesting to identify concrete operators of interest that actually belong to such RKHSs. Similar questions are also open for the Barron [46] and operator Barron spaces [83] that correspond to NN models instead of RFMs.

The parameter complexity bound $m\gtrsim\varepsilon^{-2}$ in (2.32) corresponds to the standard "Monte Carlo" parametric rate of estimation. Due to the i.i.d. sampling in Definition 2.7 of the RFM, we expect this parametric rate to be sharp. However, the sample complexity bound $n\gtrsim\varepsilon^{-4}$ from (2.32) is likely not sharp for fixed $F^\dagger$. Indeed, it is a worst case bound [27] that presumably can be improved to $n\gtrsim\varepsilon^{-(2+\delta)}$ for some small $\delta>0$ under stronger assumptions; see, e.g., [129] in the $\mathcal{Y}=\mathbb{R}$ setting. Such "fast rates" are empirically observable in numerical experiments. We remark that the constants in Theorem 2.12 were not optimized and could be improved. Additional refinements to Theorems 2.11 and 2.12 that account for discretization error, noisy output data, and smoothness misspecification may be found in [92, sect. 3].

## 3. Application to PDE Solution Operators.

In this section, we design the random feature maps $\varphi\colon\mathcal{X}\times\Theta\to\mathcal{Y}$ and measures $\mu$ for the RFM approximation of two particular PDE parameter-to-solution maps: the evolution semigroup of the viscous Burgers equation in subsection 3.1 and the coefficient-to-solution operator for the Darcy problem in subsection 3.2. It is well known to kernel method practitioners that the choice of kernel (which in this work follows from the choice of $(\varphi,\mu)$) plays a central role in the quality of the function reconstruction. While our method is purely data-driven and requires no knowledge of the governing PDE, we take the view that any prior knowledge can, and should, be introduced into the design of $(\varphi,\mu)$. However, the question of how to automatically determine good random feature pairs for a particular problem or dataset, inducing data-adapted kernels, is open. The maps $\varphi$ that we choose to employ are nonlinear in both arguments. We also detail the probability measure $\nu$ on the input space $\mathcal{X}$ for both of the PDE applications; this choice is crucial because while we desire the trained RFM to transfer to arbitrary out-of-distribution inputs from $\mathcal{X}$, we can in general only expect the learned map to perform well when restricted to inputs statistically similar to those sampled from $\nu$.

### 3.1. Burgers' Equation: Formulation.

The viscous Burgers equation in one spatial dimension is representative of the advection-dominated PDE problem class in some regimes; these time-dependent equations are not conservation laws due to the presence of small dissipative terms, but nonlinear transport still plays a central role in the evolution of solutions. The initial value problem we consider is

$$\begin{cases}\dfrac{\partial u}{\partial t}+\dfrac{\partial}{\partial x}\left(\dfrac{u^2}{2}\right)-\varepsilon\dfrac{\partial^2 u}{\partial x^2}=f & \text{in}\quad(0,\infty)\times(0,1)\,,\\ u(\cdot,0)=u(\cdot,1)\,,\quad \dfrac{\partial u}{\partial x}(\cdot,0)=\dfrac{\partial u}{\partial x}(\cdot,1) & \text{in}\quad(0,\infty)\,,\\ \qquad\qquad u(0,\cdot)=a & \text{in}\quad(0,1)\,,\end{cases} \tag{3.1}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

where $\varepsilon > 0$ is the viscosity (i.e., diffusion coefficient) and we have imposed periodic boundary conditions. The initial condition $a$ serves as the input and is drawn according to a Gaussian measure defined by

$$a \sim \nu := N(0, \mathcal{C})\,, \tag{3.2}$$

with Matérn-like covariance operator [43, 104]

$$\mathcal{C} := \tau^{2\alpha - d}(-\Delta + \tau^2 \,\mathrm{Id})^{-\alpha}\,, \tag{3.3}$$

where $d = 1$ and the negative Laplacian $-\Delta$ is defined over the torus $\mathbb{T}^1 = [0,1]_{\mathrm{per}}$ and restricted to functions which integrate to zero over $\mathbb{T}^1$. The hyperparameter $\tau \geq 0$ is an inverse length scale and $\alpha > 1/2$ controls the regularity of the draw. Such $a$ are almost surely Hölder and Sobolev regular with exponent up to $\alpha - 1/2$ [38, Thm. 12, p. 338], so in particular $a \in \mathcal{X} := L^2(\mathbb{T}^1; \mathbb{R})$. Then, for all $\varepsilon > 0$, the unique global solution $u(t, \cdot)$ to (3.1) is real analytic for all $t > 0$ [82, Thm. 1.1]. Hence, setting the output space to be $\mathcal{Y} := H^s(\mathbb{T}^1; \mathbb{R})$ for any $s > 0$, we may define the solution map

$$\begin{aligned} F^\dagger \colon L^2 &\to H^s\,, \\ a &\mapsto F^\dagger(a) := \Psi_T(a) = u(T, \cdot)\,, \end{aligned} \tag{3.4}$$

where $\{\Psi_t\}_{t>0}$ forms the solution operator semigroup for (3.1) and we fix the final time $t = T > 0$. The map $F^\dagger$ is smoothing and nonlinear.

We now describe a random feature map for use in the RFM (2.19) that we call *Fourier space random features*. Let $\mathcal{F}$ denote the Fourier transform over spatial domain $\mathbb{T}^1$ and define $\varphi \colon \mathcal{X} \times \Theta \to \mathcal{Y}$ by

$$\varphi(a; \theta) := \sigma\big(\mathcal{F}^{-1}(\chi \mathcal{F} a \mathcal{F} \theta)\big)\,, \tag{3.5}$$

where $\sigma(\,\cdot\,)$, the ELU function defined below, is defined as a mapping on $\mathbb{R}$ and applied pointwise to functions. Considering $\Theta \subseteq L^2(\mathbb{T}^1; \mathbb{R})$, the randomness enters through $\theta \sim \mu := N(0, \mathcal{C}')$ with $\mathcal{C}'$ the same covariance operator as in (3.3) but with potentially different inverse length scale and regularity, and the *wavenumber filter function* $\chi \colon \mathbb{Z} \to \mathbb{R}_{\geq 0}$ is given for $k \in \mathbb{Z}$ by

$$\chi(k) := \sigma_\chi(2\pi |k| \delta)\,, \quad \text{where} \quad \sigma_\chi(r) := \max\Big(0, \min\big(2r, (r + 1/2)^{-\beta}\big)\Big)\,, \tag{3.6}$$

$\delta > 0$, and $\beta > 0$. The map $\varphi(\,\cdot\,; \theta)$ essentially performs a filtered random convolution with the initial condition. Figure 2(a) illustrates a sample input and output from $\varphi$. Although simply hand-tuned for performance and not optimized, the filter $\chi$ is designed to shuffle energy in low to medium wavenumbers and cut off high wavenumbers (see Figure 2(b)), reflecting our prior knowledge of solutions to (3.1).

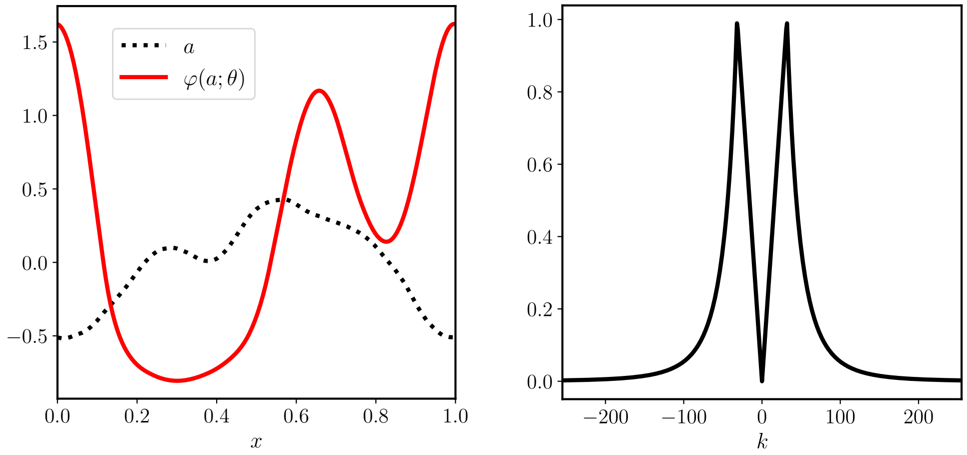
We choose the activation function $\sigma$ in (3.5) to be the exponential linear unit

$$r \mapsto \mathrm{ELU}(r) := \begin{cases} r & \text{if } r \geq 0\,, \\ e^r - 1 & \text{if } r < 0\,. \end{cases} \tag{3.7}$$

The ELU function has been used successfully as activation in other machine learning frameworks for related nonlinear PDE problems [94, 118, 119]. We also find $\mathrm{ELU}(\,\cdot\,)$ to perform better in the RFM framework than several other choices including $\mathrm{ReLU}(\,\cdot\,)$, $\tanh(\,\cdot\,)$, $\mathrm{sigmoid}(\,\cdot\,)$, $\sin(\,\cdot\,)$, $\mathrm{SELU}(\,\cdot\,)$, and $\mathrm{softplus}(\,\cdot\,)$. Note that the pointwise evaluation of the ELU function in (3.5) will be well defined, by Sobolev embedding, for $s > 1/2$ sufficiently large in the definition of $\mathcal{Y} = H^s$. Since the solution operator maps into $H^s$ for any $s > 0$, this does not constrain the method.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy



(a) Random feature

(b) Filter function

FIG. 2. *Random feature map construction for Burgers' equation:* (a) *displays a representative input-output pair for the random feature* $\varphi(\,\cdot\,;\theta)$ *with* $\theta\sim\mu$ (3.5)*, while* (b) *shows the filter* $k\mapsto\chi(k)$ *for* $\delta=0.0025$ *and* $\beta=4$ (3.6).

**3.2. Darcy Flow: Formulation.** Divergence form elliptic equations [61] arise in a variety of applications, in particular, the groundwater flow in a porous medium governed by Darcy's law [13]. This linear elliptic boundary value problem reads

$$\begin{cases}-\nabla\cdot(a\nabla u)=f & \text{in}\quad D\,,\\ \qquad\quad u=0 & \text{on}\quad \partial D\,,\end{cases}\tag{3.8}$$

where $D$ is a bounded open subset in $\mathbb{R}^d$, $f$ represents sources and sinks of fluid, $a$ the permeability of the porous medium, and $u$ is the piezometric head; all three functions map $D$ into $\mathbb{R}$ and, in addition, $a$ is strictly positive almost everywhere in $D$. We work in a setting where $f$ is fixed and consider the input-output map defined by $a\mapsto u$. The measure $\nu$ on $a$ is a high contrast level set prior constructed as the pushforward of a Gaussian measure:

$$a\sim\nu:=\psi_\sharp N(0,\mathcal{C})\,.\tag{3.9}$$

Here $\psi\colon\mathbb{R}\to\mathbb{R}$ is a threshold function defined for $r\in\mathbb{R}$ by

$$\psi(r):=a^+\mathbb{1}_{(0,\infty)}(r)+a^-\mathbb{1}_{(-\infty,0)}(r)\,,\quad\text{where}\quad 0<a^-\le a^+<\infty\,,\tag{3.10}$$

applied pointwise to functions, and the covariance operator $\mathcal{C}$ is given in (3.3) with $d=2$ and homogeneous Neumann boundary conditions on $-\Delta$. That is, the resulting coefficient $a$ almost surely takes only two values ($a^+$ or $a^-$) and, as the zero level set of a Gaussian random field, exhibits random geometry in the physical domain $D$. It follows that $a\in L^\infty(D;\mathbb{R}_{\ge0})$ almost surely. Further, the size of the contrast ratio $a^+/a^-$ measures the scale separation of this elliptic problem and hence controls the difficulty of reconstruction [17]. See Figure 3(a) for a representative sample.

Given $f\in L^2(D;\mathbb{R})$, the standard Lax–Milgram theory may be applied to show that for coefficient $a\in\mathcal{X}:=L^\infty(D;\mathbb{R}_{\ge0})$, there exists a unique weak solution $u\in\mathcal{Y}:=H_0^1(D;\mathbb{R})$ for (3.8) (see, e.g., Evans [50]). Thus, we define the ground truth

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

solution map

$$\begin{aligned} F^\dagger \colon L^\infty &\to H_0^1\,, \\ a &\mapsto F^\dagger(a) := u\,. \end{aligned} \tag{3.11}$$

Although the PDE (3.8) is linear, the solution map $F^\dagger$ is nonlinear.

We now describe the chosen random feature map for this problem, which we call *predictor-corrector random features.* Define $\varphi\colon \mathcal{X}\times\Theta\to\mathcal{Y}$ by $\varphi(a;\theta) := p_1$ such that

$$-\Delta p_0 = \frac{f}{a} + \sigma_\gamma(\theta_1)\,, \tag{3.12a}$$

$$-\Delta p_1 = \frac{f}{a} + \sigma_\gamma(\theta_2) + \nabla(\log a)\cdot\nabla p_0\,, \tag{3.12b}$$

where the boundary conditions are homogeneous Dirichlet, $\theta = (\theta_1,\theta_2)\sim\mu := \mu'\times\mu'$ are two Gaussian random fields each drawn from $\mu' := N(0,\mathcal{C}')$, $f$ is the source term in (3.8), and $\gamma = (s^+, s^-, \delta)$ are parameters for a thresholded sigmoid $\sigma_\gamma\colon\mathbb{R}\to\mathbb{R}$,

$$r \mapsto \sigma_\gamma(r) := \frac{s^+ - s^-}{1+e^{-r/\delta}} + s^-\,, \tag{3.13}$$

and extended as a Nemytskii operator when applied to $\theta_1(\,\cdot\,)$ or $\theta_2(\,\cdot\,)$. We consider $\Theta\subseteq L^2(D;\mathbb{R})\times L^2(D;\mathbb{R})$. In practice, since $\nabla a$ is not well defined when drawn from the level set measure, we replace $a$ with $a_\varepsilon$, where $a_\varepsilon := v(1,\cdot)$ is a smoothed version of $a$ obtained by evolving the following linear heat equation for one time unit:

$$\begin{cases} \dfrac{\partial v}{\partial t} = \eta\Delta v & \text{in} \quad (0,1)\times D\,, \\ \mathsf{n}\cdot\nabla v = 0 & \text{on} \quad (0,1)\times\partial D\,, \\ v(0,\cdot) = a & \text{in} \quad D\,, \end{cases} \tag{3.14}$$

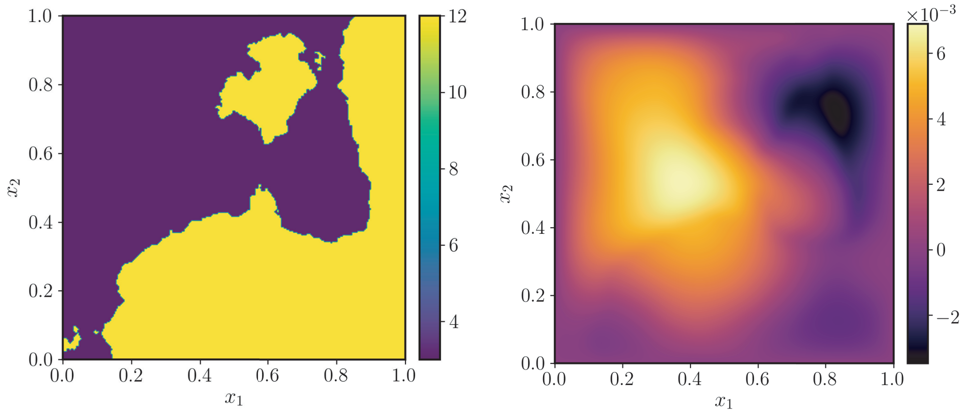
where $\mathsf{n}$ is the outward unit normal vector to $\partial D$. An example of the response $\varphi(a;\theta)$ to a piecewise constant input $a\sim\nu$ is shown in Figure 3 for some $\theta\sim\mu$.

We remark that by removing the two random terms involving $\theta_1$ and $\theta_2$ in (3.12), we obtain a remarkably accurate surrogate model for the PDE. This observation is representative of a more general iterative method, a predictor-corrector type iteration, for solving the Darcy equation (3.8), whose convergence depends on the size of $a$. The map $\varphi$ is essentially a random perturbation of a single step of this iterative method: (3.12a) makes a coarse prediction of the output, then (3.12b) improves this prediction with a correction term derived from expanding the original PDE. This choice of $\varphi$ falls within an ensemble viewpoint that the RFM may be used to improve preexisting surrogate models by taking $\varphi(\,\cdot\,;\theta)$ to be an existing emulator, but randomized in a principled way through $\theta\sim\mu$.

For this particular example, we are cognizant of the facts that the random feature map $\varphi$ requires full knowledge of the Darcy equation and a naive evaluation of $\varphi$ may be as expensive as solving the original PDE, which is itself linear; however, we believe that the ideas underlying the random features used here are intuitive and suggestive of what is possible in other application areas. For example, RFMs may be applied on larger domains with simple geometries, viewed as supersets of the physical domain of interest, enabling the use of efficient algorithms such as the fast Fourier transform (FFT) even though these may not be available for the original problem, either because the operator to be inverted is spatially inhomogeneous or because of the complicated geometry of the physical domain.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy



(a) $a \sim \nu$ (b) $\varphi(a;\theta)$, where $\theta \sim \mu$

FIG. 3. *Random feature map construction for Darcy flow:* (a) *displays a representative input draw $a$ with $\tau = 3$, $\alpha = 2$ and $a^+ = 12$, $a^- = 3$;* (b) *shows the output random feature $\varphi(a;\theta)$ (equation* (3.12)*) taking the coefficient $a$ as input. Here, $f \equiv 1$, $\tau' = 7.5$, $\alpha' = 2$, $s^+ = 1/a^+$, $s^- = -1/a^-$, and $\delta = 0.15$.*

## 4. Numerical Experiments.

We now assess the performance of our proposed methodology on the approximation of operators $F^\dagger \colon \mathcal{X} \to \mathcal{Y}$ presented in section 3. Practical implementation of the approach on a computer necessitates discretization of the input-output function spaces $\mathcal{X}$ and $\mathcal{Y}$. Hence, in the numerical experiments that follow, all infinite-dimensional objects such as the training data, evaluations of random feature maps, and random fields are discretized on an equispaced mesh with $K$ grid points to take advantage of the $O(K \log K)$ computational speed of the FFT. The simple choice of equispaced points does not limit the proposed approach, as our formulation of the RFM on function space allows the method to be implemented numerically with any choice of spatial discretization. Such a numerical discretization procedure leads to the problem of high- but finite-dimensional approximation of discretized target operators mapping $\mathbb{R}^K$ to $\mathbb{R}^K$ by similarly discretized RFMs. However, we emphasize the fact that $K$ is allowed to vary, and we study the properties of the discretized RFM as $K$ varies, noting that since the RFM is defined conceptually on function space in section 2 without reference to discretization, its discretized numerical realization has approximation quality consistent with the infinite-dimensional limit $K \to \infty$. This implies that the same trained model can be deployed across the entire hierarchy of finite-dimensional spaces $\mathbb{R}^K$ parametrized by $K \in \mathbb{N}$ without the need to be retrained, provided $K$ is sufficiently large. Thus, in this section, our notation does not make explicit the dependence of the discretized RFM or target operators on mesh size $K$. We demonstrate these claimed properties numerically.

The input functions and our chosen random feature maps (3.5) and (3.12) require i.i.d. draws of Gaussian random fields to be fully defined. We efficiently sample these fields by truncating a Karhunen–Loéve expansion and employing fast summation of the eigenfunctions with FFTs. More precisely, on a mesh of size $K$, denote by $g(\cdot)$ a numerical approximation of a Gaussian random field on domain $D = (0,1)^d$, $d = 1, 2$:

$$g = \sum_{k \in Z_K} \xi_k \sqrt{\lambda_k} \phi_k \approx \sum_{k' \in \mathbb{Z}^d_{\geq 0}} \xi_{k'} \sqrt{\lambda_{k'}} \phi_{k'} \sim N(0, \mathcal{C}) \,, \tag{4.1}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

where $\xi_j \sim N(0,1)$ i.i.d. for each $j$ and $Z_K \subset \mathbb{Z}_{\geq 0}$ is a truncated one-dimensional lattice of cardinality $K$ ordered such that $\{\lambda_j\}$ is nonincreasing. The pairs $(\lambda_{k'}, \phi_{k'})$ are found by solving the eigenvalue problem $\mathcal{C}\phi_{k'} = \lambda_{k'}\phi_{k'}$ for nonnegative, symmetric, trace-class operator $\mathcal{C}$ (3.3). Concretely, these solutions are given by

$$\phi_{k'}(x) = \begin{cases} \sqrt{2}\cos(k_1'\pi x_1)\cos(k_2'\pi x_2), & k_1' \text{ or } k_2' = 0\,, \\ 2\cos(k_1'\pi x_1)\cos(k_2'\pi x_2) & \text{otherwise}\,, \end{cases} \qquad \lambda_{k'} = \tau^{2\alpha-2}(\pi^2|k'|^2 + \tau^2)^{-\alpha}, \tag{4.2}$$

for homogeneous Neumann boundary conditions when $d = 2$, $k' = (k_1', k_2') \in \mathbb{Z}^2_{\geq 0}\backslash\{0\}$, $x = (x_1, x_2) \in (0,1)^2$. They are given by

$$\phi_{2j}(x) = \sqrt{2}\cos(2\pi j x)\,, \quad \phi_{2j-1}(x) = \sqrt{2}\sin(2\pi j x)\,, \quad \phi_0(x) = 1\,, \tag{4.3a}$$

$$\lambda_{2j} = \lambda_{2j-1} = \tau^{2\alpha-1}(4\pi^2 j^2 + \tau^2)^{-\alpha}\,, \quad \lambda_0 = \tau^{-1}, \tag{4.3b}$$

for periodic boundary conditions when $d = 1$, $j \in \mathbb{Z}_{>0}$, and $x \in (0,1)$. In both cases, we enforce that $g$ integrates to zero over $D$ by manually setting to zero the Fourier coefficient corresponding to multi-index $k' = 0$. We use such a $g$ in all experiments that follow. Additionally, the $k$ and $k'$ used in this section to denote wavenumber indices should not be confused with our previous notation for kernels.

With the discretization and data generation setup now well defined, and the pairs $(\varphi, \mu)$ given in section 3, the last algorithmic step is to train the RFM by solving (2.25) and then test its performance. For a fixed number of random features $m$, we only train and test a single realization of the RFM, viewed as a random variable itself. In each instance $m$ is varied in the experiments that follow, and the draws $\{\theta_j\}_{j=1}^m$ are resampled i.i.d. from $\mu$. To measure the distance between the trained RFM $F_m(\cdot\,;\widehat{\alpha})$ and the ground truth $F^\dagger$, we employ the *approximate expected relative test error*

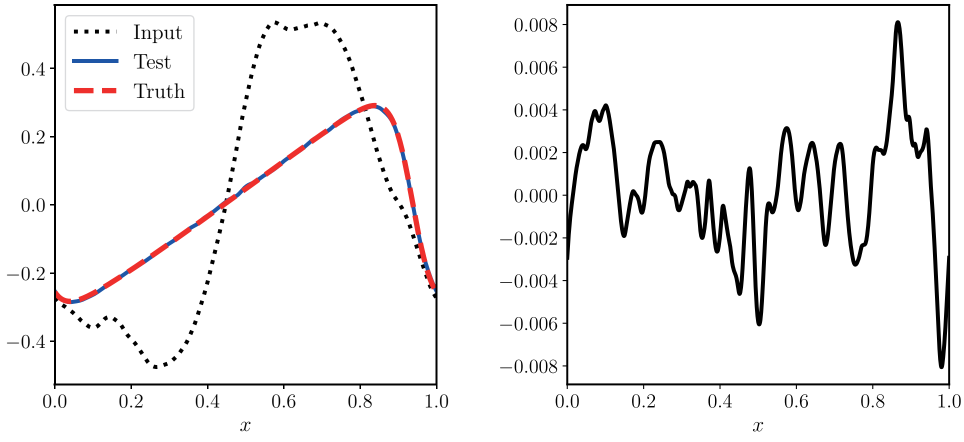
$$e_{n',m} := \frac{1}{n'}\sum_{j=1}^{n'} \frac{\|F^\dagger(a_j') - F_m(a_j';\widehat{\alpha})\|_{L^2}}{\|F^\dagger(a_j')\|_{L^2}} \approx \mathbb{E}^{a'\sim\nu}\left[\frac{\|F^\dagger(a') - F_m(a';\widehat{\alpha})\|_{L^2}}{\|F^\dagger(a')\|_{L^2}}\right], \tag{4.4}$$

where the $\{a_j'\}_{j=1}^{n'}$ are drawn i.i.d. from $\nu$ and $n'$ denotes the number of input-output pairs used for testing. All $L^2(D;\mathbb{R})$ norms on the physical domain are numerically approximated by composite trapezoid rule quadrature. Since $\mathcal{Y} \subset L^2$ for both the PDE solution operators (3.4) and (3.11), we also perform all required inner products during training in $L^2$ rather than in $\mathcal{Y}$; this results in smaller relative test error $e_{n',m}$.

**4.1. Burgers' Equation: Experiment.** We generate a high resolution dataset of input-output pairs by solving Burgers' equation (3.1) on an equispaced periodic mesh of size $K = 1025$ (identifying the first mesh point with the last) with random initial conditions sampled from $\nu = N(0, \mathcal{C})$ using (4.1), where $\mathcal{C}$ is given by (3.3) with parameter choices $\tau = 7$ and $\alpha = 2.5$. The full order solver is an FFT-based pseudospectral method for spatial discretization [54] and a fourth order Runge–Kutta integrating factor time-stepping scheme for time discretization [79]. All data represented on mesh sizes $K < 1025$ used in both training and testing phases are subsampled from this original dataset, and hence we consider numerical realizations of $F^\dagger$ (3.4) up to $\mathbb{R}^{1025} \to \mathbb{R}^{1025}$. We fix $n = 512$ training and $n' = 4000$ testing pairs unless otherwise noted and also fix the viscosity to $\varepsilon = 10^{-2}$ in all experiments. Lowering $\varepsilon$ leads to smaller length scale solutions and more difficult reconstruction; more data (higher $n$)

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy



(a) Input and output

(b) Pointwise error

FIG. 4. *Representative input-output test sample for the Burgers equation solution map $F^\dagger := \Psi_1$:* (a) *shows a sample input, output (truth), and trained RFM prediction (test), while* (b) *displays the pointwise error. The relative $L^2$ error for this single prediction is* 0.0146. *Here,* $n = 512$, $m = 1024$, *and* $K = 1025$.

and features (higher $m$) or a more expressive choice of $(\varphi, \mu)$ would be required to achieve comparable error levels due to the slow decaying Kolmogorov width of the solution map. For simplicity, we set the forcing $f \equiv 0$, although nonzero forcing could lead to other interesting solution maps such as $f \mapsto u(T, \cdot)$. It is easy to check that the solution will have zero mean for all time and a steady state of zero. Hence, we choose $T \leq 2$ to ensure that the solution is far enough away from steady state. For the random feature map (3.5), we fix the hyperparameters $\alpha' = 2$, $\tau' = 5$, $\delta = 0.0025$, and $\beta = 4$. The map itself is evaluated efficiently with the FFT and requires no other tools to be discretized. RFM hyperparameters are hand-tuned but not optimized. We find that regularization during training has a negligible effect for this problem, so the RFM is trained with $\lambda = 0$ by solving the normal equations (2.25) with the pseudoinverse to deliver the minimum norm least squares solution; we use the truncated SVD implementation in Python's `scipy.linalg.pinv2` for this purpose.

Our experiments study the RFM approximation to the viscous Burgers equation evolution operator semigroup (3.4). As a visual aid for the high-dimensional problem at hand, Figure 4 shows a representative sample input and output along with a trained RFM test prediction. To determine whether the RFM has actually learned the correct evolution operator, we test the semigroup property of the map; [150] pursues closely related work also in a Fourier space setting. Denote the $(j-1)$-fold composition of a function $G$ with itself by $G^j$. Then, with $u(0, \cdot) = a$, we have

$$(\Psi_T \circ \cdots \circ \Psi_T)(a) = \Psi_T^j(a) = \Psi_{jT}(a) = u(jT, \cdot) \tag{4.5}$$

by definition. We train the RFM on input-output pairs from the map $\Psi_T$ with $T := 0.5$ to obtain $\widehat{F} := F_m(\,\cdot\,; \widehat{\alpha})$. Then, it should follow from (4.5) that $\widehat{F}^j \approx \Psi_{jT}$; that is, each application of $\widehat{F}$ should evolve the solution $T$ time units. We test this semigroup approximation by learning the map $\widehat{F}$ and then comparing $\widehat{F}^j$ on $n' = 4000$ fixed inputs to outputs from each of the operators $\Psi_{jT}$, with $j \in \{1, 2, 3, 4\}$ (the solutions at time $T$, $2T$, $3T$, $4T$). The results are presented in Table 1 for a fixed mesh size

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

TABLE 1

*Expected relative error $e_{n',m}$ for time upscaling with the learned RFM operator semigroup for Burgers' equation. Here, $n' = 4000$, $m = 1024$, $n = 512$, and $K = 129$. The RFM is trained on data from the evolution operator $\Psi_{T=0.5}$ and then tested on input-output samples generated from $\Psi_{jT}$, where $j = 2, 3, 4$, by repeated composition of the learned model. The increase in error is small even after three compositions, reflecting excellent out-of-distribution performance.*

| Train on: | $T = 0.5$ | Test on: | $2T = 1.0$ | $3T = 1.5$ | $4T = 2.0$ |
|---|---|---|---|---|---|
| | 0.0360 | | 0.0407 | 0.0528 | 0.0788 |

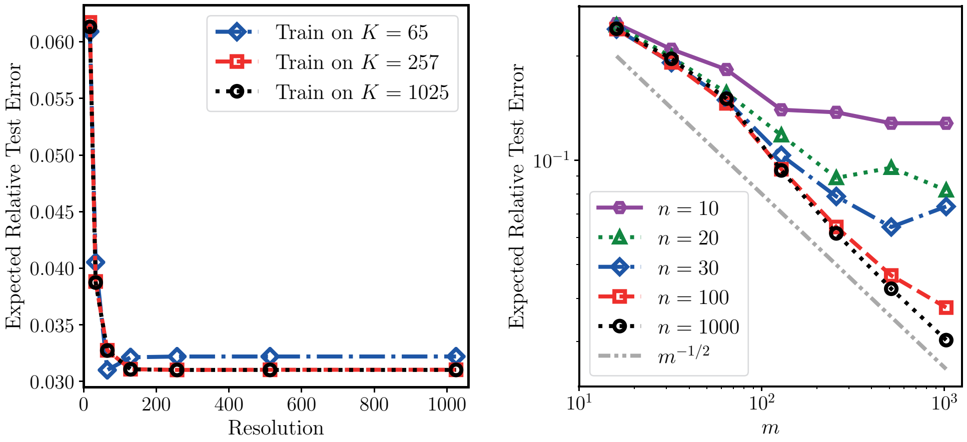
$K = 129$. We observe that the composed RFM map $\widehat{F}^j$ accurately captures $\Psi_{jT}$, though this accuracy deteriorates as $j$ increases due to error propagation in time, as is common with any traditional integrator. However, even after three compositions corresponding to 1.5 time units past the training time $T = 0.5$, the relative error only increases by around 0.04. It is remarkable that the RFM learns time evolution without explicitly time-stepping the PDE (3.1) itself. Such a procedure is coined *time upscaling* in the PDE context and in some sense breaks the CFL stability barrier [40]. Table 1 is evidence that the RFM has excellent out-of-distribution performance: although only trained on inputs $a \sim \nu$, the model outputs accurate predictions given new input samples $\Psi_{jT}(a) \sim (\Psi_{jT})_\sharp\nu$.

We next study the ability of the RFM to transfer its learned coefficients $\widehat{\alpha}$ obtained from training on mesh size $K$ to different mesh resolutions $K'$ in Figure 5(a). We fix $T := 1$ in what follows and observe that the lowest test error occurs when $K = K'$, that is, when the train and test resolutions are identical; this behavior was also observed in the contemporaneous work [97]. At very low resolutions, such as $K = 17$ here, the test error is dominated by discretization error which can become quite large; for example, resolving conceptually infinite-dimensional objects such as the Fourier space based feature map in (3.5) or the $L^2$ norms in (4.4) with only 17 grid points gives bad accuracy. However, outside this regime, the errors are essentially constant across resolution regardless of the training resolution $K$, indicating that the RFM learns its optimal coefficients independently of the resolution and hence generalizes well to any desired mesh size. In fact, the trained model could be deployed on different discretizations of the domain $D$ (e.g., various choices of finite elements, graph-based/particle methods), not just with different mesh sizes. Practically speaking, this means that high resolution training sets can be subsampled to mesh sizes $K$ that are smaller (yet still large enough to avoid large discretization error) for faster training, leading to a trained model with nearly the same accuracy at all higher resolutions.

The smallest expected relative test error achieved by the RFM is 0.0303 for the configuration in Figure 5(b). This excellent performance is encouraging because the error we report is of the same order of magnitude as that reported in [96, sect. 5.1] for the same Burgers solution operator that we study, but with slightly different problem parameter choices. We emphasize that the neural operator methods in that work are based on deep learning, which involves training NNs by solving a nonconvex optimization problem with stochastic gradient descent, while our random feature methods have orders of magnitude fewer trainable parameters that are easily optimized through convex optimization. In Figure 5(b), we see that for large enough $n$, the error empirically follows the $O(m^{-1/2})$ parameter complexity bound that is suggested by Theorem 2.12. This theorem does not directly apply here because it requires the regularization parameter $\lambda$ to be strictly positive and $F^\dagger$ to be in the RKHS of $(\varphi, \mu)$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

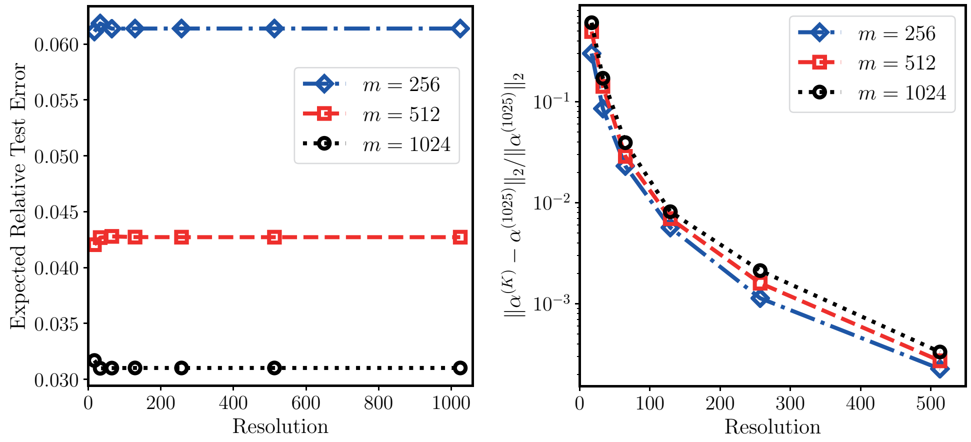
Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy



(a) Test error vs. resolution

(b) Test error vs. $m$ and $n$

FIG. 5. *Expected relative test error of a trained RFM for the Burgers evolution operator* $F^\dagger = \Psi_1$ *with* $n' = 4000$ *test pairs:* (a) *displays the invariance of test error w.r.t. training and testing on different resolutions for* $m = 1024$ *and* $n = 512$ *fixed; the RFM can train and test on different mesh sizes without loss of accuracy.* (b) *shows the decay of the test error for resolution* $K = 129$ *fixed as a function of* $m$ *and* $n$*; the error follows the* $O(m^{-1/2})$ *Monte Carlo rate remarkably well and the smallest error achieved is* 0.0303 *for* $n = 1000$ *and* $m = 1024$.



(a) Test error vs. resolution

(b) Minimizer error vs. resolution

FIG. 6. *Results of a trained RFM for the Burgers equation evolution operator* $F^\dagger = \Psi_1$*:* (a) *shows resolution-invariant test error for various* $m$. (b) *displays the relative error of the learned coefficient* $\alpha$ *w.r.t. the coefficient learned on the highest mesh size* ($K = 1025$)*. Here,* $n = 512$ *training and* $n' = 4000$ *testing pairs were used.*

from subsection 3.1, which we do not verify. Nonetheless, Figure 5(b) indicates that the error bounds for the trained RFM hold for a larger class of problems than the stated assumptions suggest.

Finally, Figure 6 demonstrates the invariance of the expected relative test error to the mesh resolution used for training and testing. This result is a consequence of framing the RFM on function space; other machine learning–based surrogate methods defined in finite dimensions exhibit an *increase* in test error as mesh resolution is

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

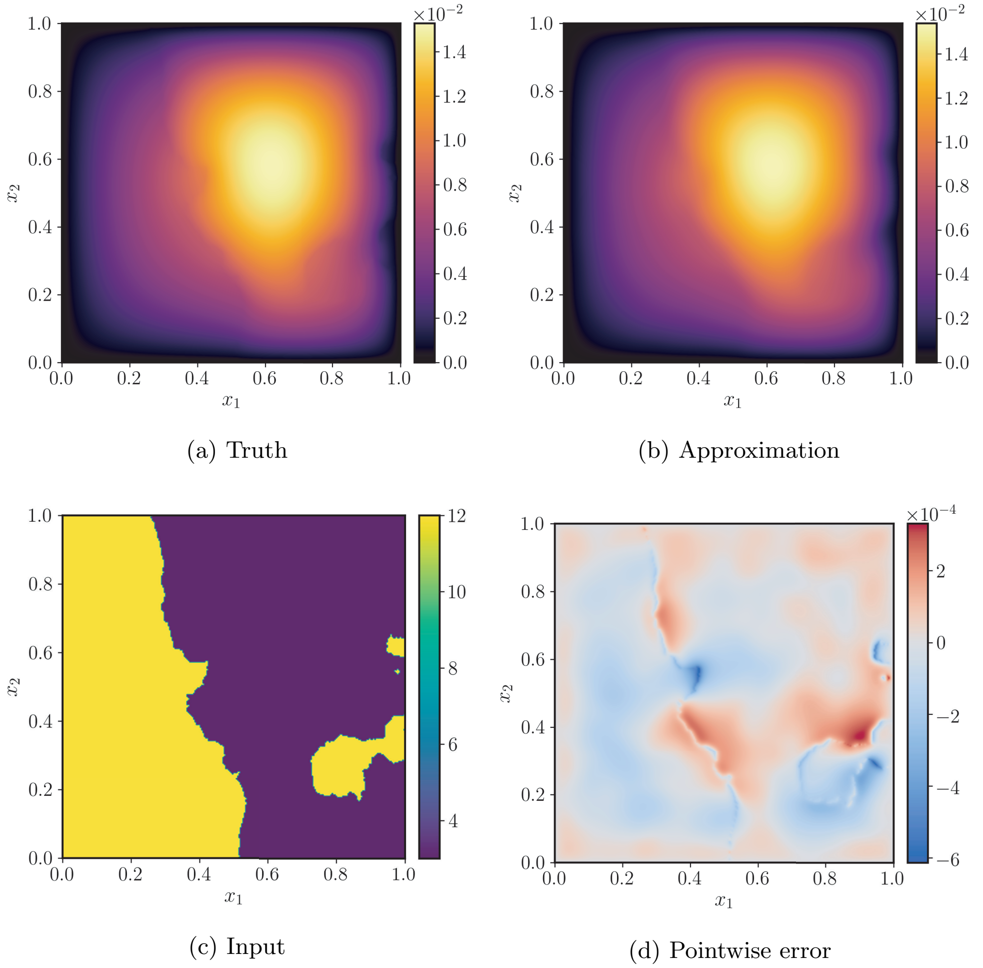
increased (see [19, sect. 4] for a numerical account of this phenomenon). Figure 6(a) shows the error as a function of mesh resolution for three values of $m$. For very low resolution, the error varies slightly but then flattens out to a constant value as $K \to \infty$. Figure 6(b) indicates that the learned coefficient $\alpha^{(K)}$ for each $K$ converges to some $\alpha^{(\infty)}$ as $K \to \infty$, again reflecting the design of the RFM as a mapping between infinite-dimensional spaces.

**4.2. Darcy Flow: Experiment.** In this section, we consider Darcy flow on the physical domain $D := (0,1)^2$, the unit square. We generate a high resolution dataset of input-output pairs for $F^\dagger$ (3.11) by solving (3.8) on an equispaced $257 \times 257$ mesh (size $K = 257^2$) using a second order finite difference scheme. All mesh sizes $K < 257^2$ are subsampled from this original dataset and hence we consider numerical realizations of $F^\dagger$ up to $\mathbb{R}^{66049} \to \mathbb{R}^{66049}$. We denote *resolution* by $r$ such that $K = r^2$. We fix $n = 128$ training and $n' = 1000$ testing pairs unless otherwise noted. The input data are drawn from the level set measure $\nu$ (3.9) with $\tau = 3$ and $\alpha = 2$ fixed. We choose $a^+ = 12$ and $a^- = 3$ in all experiments that follow and hence the contrast ratio $a^+/a^- = 4$ is fixed. The source is fixed to $f \equiv 1$, the constant function. We evaluate the predictor-corrector random features $\varphi$ (3.12) using an FFT-based fast Poisson solver corresponding to an underlying second order finite difference stencil at a cost of $O(K \log K)$ per solve. The smoothed coefficient $a_\varepsilon$ in the definition of $\varphi$ is obtained by solving (3.14) with time step 0.03 and diffusion constant $\eta = 10^{-4}$; with centered second order finite differences, this incurs 34 time steps and hence a cost $O(34K)$. We fix the hyperparameters $\alpha' = 2$, $\tau' = 7.5$, $s^+ = 1/12$, $s^- = -1/3$, and $\delta = 0.15$ for the map $\varphi$. Unlike in subsection 4.1, we find via grid search on $\lambda$ that regularization during training does improve the reconstruction of the Darcy flow solution operator and hence we train with $\lambda := 10^{-8}$ fixed. We remark that, for simplicity, the above hyperparameters were not systematically and jointly optimized; as a consequence the RFM performance has room to improve beyond the results in this section.

Darcy flow is characterized by the geometry of the high contrast coefficients $a \sim \nu$. As seen in Figure 7, the solution inherits the steep interfaces of the input. However, we see that a trained RFM with predictor-corrector random features (3.12) captures these interfaces well, albeit with slight smoothing; the error concentrates on the location of the interface. The effect of increasing $m$ and $n$ on the test error is shown in Figure 8(b). Here, the error appears to saturate more than was observed for the Burgers equation problem (Figure 5(b)) and does not follow the $O(m^{-1/2})$ rate. This is likely due to our fixing $\lambda$ to be constant instead of scaling it with $m$ as suggested by Theorem 2.12. It is also possible that the Darcy flow solution map does not belong to the RKHS $\mathcal{H}_{k_\mu}$, leading to an additional misspecification error. However, the smallest test error achieved for the best performing RFM configuration is 0.0381, which is on the same scale as the error reported in competing neural operator–based methods [19, 97] for the same setup.

The RFM is able to be successfully trained and tested on different resolutions for Darcy flow. Figure 8(a) shows that, again, for low resolutions, the smallest relative test error is achieved when the train and test resolutions are identical (here, for $r = 17$). However, when the resolution is increased away from this low resolution regime, the relative test error slightly increases then approaches a constant value, reflecting the function space design of the method. Training the RFM on a high resolution mesh poses no issues when transferring to lower or higher resolutions for model evaluation, and it achieves consistent error for test resolutions sufficiently large (i.e., $r \geq 33$, the

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy



(a) Truth

(b) Approximation

(c) Input

(d) Pointwise error

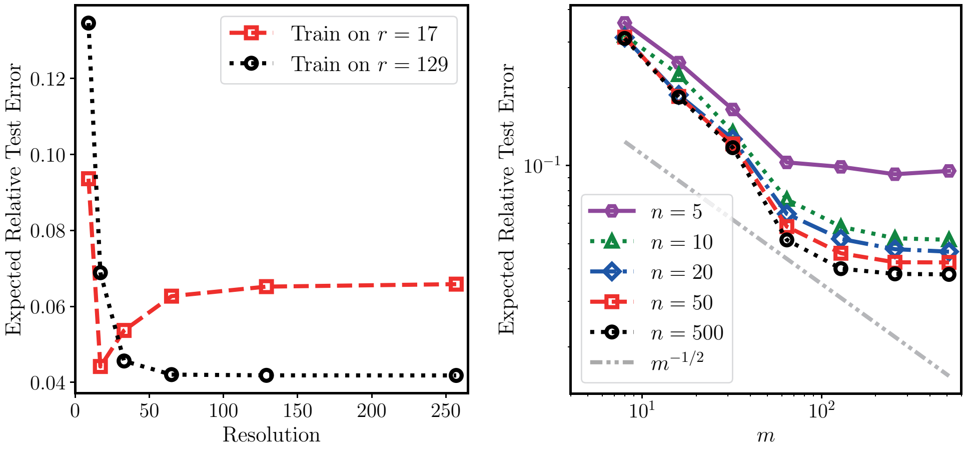
FIG. 7. *Representative input-output test sample for the Darcy flow solution map:* (c) *shows a sample input,* (a) *the resulting output (truth),* (b) *a trained RFM prediction, and* (d) *the pointwise error. The relative $L^2$ error for this single prediction is* 0.0122. *Here, $n$ = 256, $m$ = 350, and $K = 257^2$.*

regime where discretization error starts to become negligible). Additionally, the RFM basis functions $\{\varphi(\cdot\,;\theta_j)\}_{j=1}^m$ are defined without any dependence on the training data, unlike in other competing approaches based on similar shallow linear approximations, such as the reduced basis method or the PCA-Net method in [19]. Consequently, our RFM may be directly evaluated on any desired mesh resolution once trained ("superresolution"), whereas those aforementioned approaches require some form of interpolation to transfer between different mesh sizes (see [19, sect. 4.3]).

In Figure 9, we again confirm that our method is invariant to the refinement of the mesh and improves with more random features. While the difference at low resolutions is more pronounced than that observed for the Burgers equation, our results for Darcy flow still suggest that the expected relative test error converges to a constant value as resolution increases; an estimate of this rate of convergence is seen in Figure 9(b), where we plot the relative error of the learned parameter $\alpha^{(r)}$ at resolution $r$ w.r.t. the parameter learned at the highest resolution trained, which was $r = 129$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

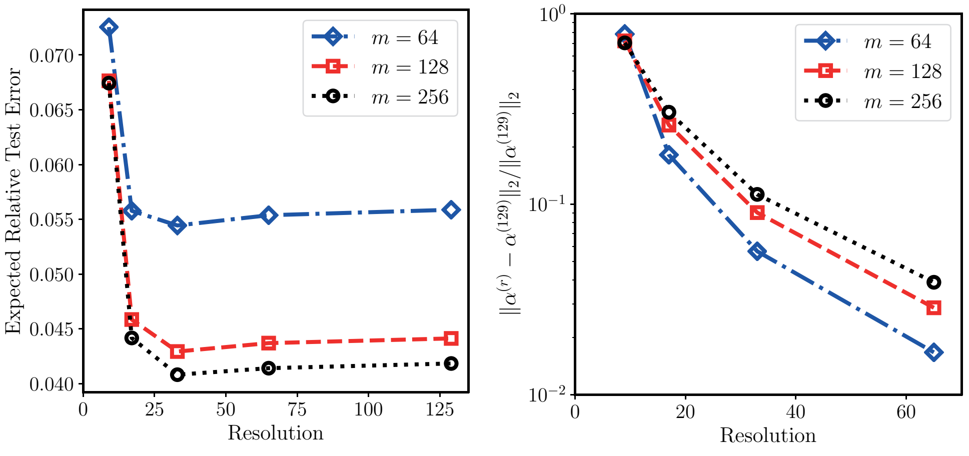
Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy



(a) Test error vs. resolution

(b) Test error vs. $m$ and $n$

FIG. 8. *Expected relative test error of a trained RFM for Darcy flow with* $n' = 1000$ *test pairs:* (a) *displays the invariance of test error w.r.t. training and testing on different resolutions for* $m = 512$ *and* $n = 256$ *fixed; the RFM can train and test on different mesh sizes without significant loss of accuracy.* (b) *shows the decay of the test error for resolution* $r = 33$ *fixed as a function of* $m$ *and* $n$*; the smallest error achieved is* 0.0381 *for* $n = 500$ *and* $m = 512$.



(a) Test error vs. resolution

(b) Minimizer error vs. resolution

FIG. 9. *Results of a trained RFM for Darcy flow:* (a) *demonstrates resolution-invariant test error for various* $m$*, while* (b) *displays the relative error of the learned coefficient* $\alpha^{(r)}$ *at resolution* $r$ *w.r.t. the coefficient learned on the highest resolution* ($r = 129$)*. Here,* $n = 128$ *training and* $n' = 1000$ *testing pairs were used.*

**5. Conclusion.** This paper introduces a random feature methodology for the data-driven estimation of operators mapping between infinite-dimensional Banach spaces. It may be interpreted as a low-rank approximation to operator-valued kernel ridge regression. Training the function-valued random features only requires solving a quadratic optimization problem for an $m$-dimensional coefficient vector. The conceptually infinite-dimensional algorithm is nonintrusive and results in a scalable method that is consistent with the continuum limit, robust to discretization, and

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

highly flexible in practical use cases. Numerical experiments confirm these benefits in scientific machine learning applications involving two nonlinear forward operators arising from PDEs. Backed by tractable training routines and theoretical guarantees, operator learning with the function-valued random features method displays considerable potential for accelerating many-query computational tasks and for discovering new models from high-dimensional experimental data in science and engineering.

Going beyond this paper, several directions for future research remain open. Some of the first theoretical results for function-valued random features are summarized in subsection 2.5. However, it is not yet known what conditions on the problem and the feature pair allow for faster rates of convergence. In addition, it is of interest to characterize the quality of the operator RKHS spaces induced by random feature pairs and whether practical problem classes actually belong to these spaces. Also of importance is the question of how to automatically adapt function-valued random features to data instead of manually constructing them. Some possibilities along this line of work include the Bayesian estimation of hyperparameters, as is frequently used in Gaussian process regression, or more general hierarchical learning of the random feature pair $(\varphi, \mu)$ itself. In tandem, there is a need for a mature function-valued random features software library that includes efficient linear solvers and GPU implementations, benchmark problems, and robust hyperparameter optimizers. These advances will further enable the random features method to learn from real-world data and solve challenging forward and inverse problems from the physical sciences, such as climate modeling and material modeling, with controlled computational complexity.

**Data and Code Availability.** Links to datasets and code used to produce the numerical results and figures in this paper are available at

https://github.com/nickhnelsen/random-features-banach .

**Acknowledgments.** The authors thank Bamdad Hosseini and Nikola Kovachki for helpful discussions and are grateful to the two anonymous referees for their careful reading of and insightful comments on the original publication of this paper. They are also grateful to Samuel Lanthaler for reading and commenting on an earlier draft of this paper.

## REFERENCES

[1] B. Adcock, S. Brugiapaglia, N. Dexter, and S. Moraga, *Deep neural networks are effective at learning high-dimensional Hilbert-valued functions from limited data*, Proc. Mach. Learn. Res., 134 (2022), pp. 1–36. (Cited on p. 539)

[2] B. Adcock, S. Brugiapaglia, N. Dexter, and S. Moraga, *Near-Optimal Learning of Banach-Valued, High-Dimensional Functions via Deep Neural Networks*, preprint, arXiv:2211.12633, 2022. (Cited on p. 540)

[3] B. Adcock, N. Dexter, and S. Moraga, *Optimal approximation of infinite-dimensional holomorphic functions*, Calcolo, 61 (2024), art. 12. (Cited on p. 540)

[4] A. Alexanderian, *Optimal experimental design for infinite-dimensional Bayesian inverse problems governed by PDEs: A review*, Inverse Problems, 37 (2021), art. 043001. (Cited on p. 538)

[5] N. Aronszajn, *Theory of reproducing kernels*, Trans. Amer. Math. Soc., 68 (1950), pp. 337–404. (Cited on p. 544)

[6] S. Arridge, P. Maass, O. Öktem, and C. Schönlieb, *Solving inverse problems using data-driven models*, Acta Numer., 28 (2019), pp. 1–174. (Cited on p. 538)

[7] F. Bach, *On the equivalence between kernel quadrature rules and random feature expansions*, J. Mach. Learn. Res., 18 (2017), pp. 714–751. (Cited on pp. 540, 544, 546)

[8] Y. Bar-Sinai, S. Hoyer, J. Hickey, and M. P. Brenner, *Learning data-driven discretizations for partial differential equations*, Proc. Natl. Acad. Sci. USA, 116 (2019), pp. 15344–

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

15349. (Cited on p. 539)

[9] J. Barnett, C. Farhat, and Y. Maday, *Neural-network-augmented projection-based model order reduction for mitigating the Kolmogorov barrier to reducibility*, J. Comput. Phys., 492 (2023), art. 112420. (Cited on p. 538)

[10] M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera, *An "empirical interpolation" method: Application to efficient reduced-basis discretization of partial differential equations*, C. R. Math. Acad. Sci. Paris, 339 (2004), pp. 667–672. (Cited on p. 538)

[11] A. R. Barron, *Universal approximation bounds for superpositions of a sigmoidal function*, IEEE Trans. Inform. Theory, 39 (1993), pp. 930–945. (Cited on p. 540)

[12] P. Batlle, M. Darcy, B. Hosseini, and H. Owhadi, *Kernel methods are competitive for operator learning*, J. Comput. Phys., 496 (2024), art. 112549. (Cited on p. 541)

[13] J. Bear and M. Y. Corapcioglu, *Fundamentals of Transport Phenomena in Porous Media*, NATO Sci. Ser. 82, Springer, New York, 2012. (Cited on p. 555)

[14] M. Belkin, D. Hsu, S. Ma, and S. Mandal, *Reconciling modern machine-learning practice and the classical bias–variance trade-off*, Proc. Natl. Acad. Sci. USA, 116 (2019), pp. 15849–15854. (Cited on p. 540)

[15] P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, *Model Reduction and Approximation: Theory and Algorithms*, Comput. Sci. Eng. 15, SIAM, Philadelphia, 2017, https://doi.org/10.1137/1.9781611974829. (Cited on pp. 538, 542)

[16] A. Berlinet and C. Thomas-Agnan, *Reproducing Kernel Hilbert Spaces in Probability and Statistics*, Springer, New York, 2011. (Cited on p. 544)

[17] C. Bernardi and R. Verfürth, *Adaptive finite element methods for elliptic equations with non-smooth coefficients*, Numer. Math., 85 (2000), pp. 579–608. (Cited on p. 555)

[18] G. Beylkin and M. J. Mohlenkamp, *Algorithms for numerical analysis in high dimensions*, SIAM J. Sci. Comput., 26 (2005), pp. 2133–2159, https://doi.org/10.1137/040604959. (Cited on p. 542)

[19] K. Bhattacharya, B. Hosseini, N. B. Kovachki, and A. M. Stuart, *Model reduction and neural networks for parametric PDEs*, SMAI J. Comput. Math., 7 (2021), pp. 121–157. (Cited on pp. 539, 547, 562, 563)

[20] D. Bigoni, Y. Chen, N. G. Trillos, Y. Marzouk, and D. Sanz-Alonso, *Data-driven forward discretizations for Bayesian inversion*, Inverse Problems, 36 (2020), art. 105008. (Cited on p. 539)

[21] N. Boullé, D. Halikias, and A. Townsend, *Elliptic PDE learning is provably data-efficient*, Proc. Natl. Acad. Sci. USA, 120 (2023), art. e2303904120. (Cited on p. 540)

[22] N. Boullé and A. Townsend, *A Mathematical Guide to Operator Learning*, preprint, arXiv:2312.14688, 2023. (Cited on p. 538)

[23] N. Boullé and A. Townsend, *Learning elliptic partial differential equations with randomized linear algebra*, Found. Comput. Math., 23 (2023), pp. 709–739. (Cited on p. 540)

[24] R. Brault, M. Heinonen, and F. Buc, *Random Fourier features for operator-valued kernels*, in Proceedings of the Asian Conference on Machine Learning, PMLR, 2016, pp. 110–125. (Cited on pp. 541, 548)

[25] S. Brivio, S. Fresca, N. R. Franco, and A. Manzoni, *Error estimates for POD-DL-ROMs: A deep learning framework for reduced order modeling of nonlinear parametrized PDEs enhanced by proper orthogonal decomposition*, Adv. Comput. Math., 50 (2024), art. 33. (Cited on p. 538)

[26] Y. Cao and Q. Gu, *Generalization bounds of stochastic gradient descent for wide and deep neural networks*, in Advances in Neural Information Processing Systems, 2019, pp. 10835–10845. (Cited on pp. 540, 548)

[27] A. Caponnetto and E. De Vito, *Optimal rates for the regularized least-squares algorithm*, Found. Comput. Math., 7 (2007), pp. 331–368. (Cited on pp. 540, 541, 553)

[28] C. Carmeli, E. De Vito, and A. Toigo, *Vector valued reproducing kernel Hilbert spaces of integrable functions and Mercer theorem*, Anal. Appl., 4 (2006), pp. 377–408. (Cited on p. 544)

[29] G. Chen and K. Fidkowski, *Output-based error estimation and mesh adaptation using convolutional neural networks: Application to a scalar advection-diffusion problem*, in Proceedings of the AIAA Scitech 2020 Forum, AIAA, 2020, art. 1143. (Cited on p. 539)

[30] J. Chen, X. Chi, W. E, and Z. Yang, *Bridging traditional and machine learning-based algorithms for solving PDEs: The random feature method*, J. Mach. Learn., 1 (2022), pp. 268–298. (Cited on p. 538)

[31] T. Chen and H. Chen, *Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems*, IEEE Trans. Neural Networks, 6 (1995), pp. 911–917. (Cited on p. 539)

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[32] M. Cheng, T. Y. Hou, M. Yan, and Z. Zhang, *A data-driven stochastic method for elliptic PDEs with random coefficients*, SIAM/ASA J. Uncertain. Quantif., 1 (2013), pp. 452–493, https://doi.org/10.1137/130913249. (Cited on p. 538)

[33] A. Chkifa, A. Cohen, R. DeVore, and C. Schwab, *Sparse adaptive Taylor approximation algorithms for parametric and stochastic elliptic PDEs*, ESAIM Math. Model. Numer. Anal., 47 (2013), pp. 253–280. (Cited on pp. 539, 540)

[34] A. Cohen and R. DeVore, *Approximation of high-dimensional parametric PDEs*, Acta Numer., 24 (2015), pp. 1–159. (Cited on pp. 538, 539, 542)

[35] A. Cohen and G. Migliorati, *Optimal weighted least-squares methods*, SMAI J. Comput. Math., 3 (2017), pp.181–203. (Cited on p. 542)

[36] S. L. Cotter, G. O. Roberts, A. M. Stuart, and D. White, *MCMC methods for functions: Modifying old algorithms to make them faster*, Statist. Sci., 28 (2013), pp. 424–446. (Cited on p. 541)

[37] F. Cucker and S. Smale, *On the mathematical foundations of learning*, Bull. Amer. Math. Soc., 39 (2002), pp. 1–49. (Cited on p. 544)

[38] M. Dashti and A. M. Stuart, *The Bayesian approach to inverse problems*, in Handbook of Uncertainty Quantification, Springer, New York, 2017, pp. 311–428, https://doi.org/10.1007/978-3-319-12385-1_7. (Cited on pp. 543, 554)

[39] M. V. de Hoop, N. B. Kovachki, N. H. Nelsen, and A. M. Stuart, *Convergence rates for learning linear operators from noisy data*, SIAM/ASA J. Uncertain. Quantif., 11 (2023), pp. 480–513, https://doi.org/10.1137/21M1442942. (Cited on p. 540)

[40] L. Demanet, *Curvelets, Wave Atoms, and Wave Equations*, Ph.D. thesis, California Institute of Technology, 2006. (Cited on p. 560)

[41] R. A. DeVore, *The theoretical foundation of reduced basis methods*, in Model Reduction and Approximation: Theory and Algorithms, SIAM, Philadelphia, 2017, pp. 137–168, https://doi.org/10.1137/1.9781611974829.ch3. (Cited on p. 538)

[42] A. Doostan and G. Iaccarino, *A least-squares approximation of partial differential equations with high-dimensional random inputs*, J. Comput. Phys., 228 (2009), pp. 4332–4345. (Cited on p. 542)

[43] M. M. Dunlop, M. A. Iglesias, and A. M. Stuart, *Hierarchical Bayesian level set inversion*, Statist. Comput., 27 (2017), pp. 1555–1584. (Cited on p. 554)

[44] W. E, *A proposal on machine learning via dynamical systems*, Commun. Math. Stat., 5 (2017), pp. 1–11. (Cited on p. 541)

[45] W. E, J. Han, and Q. Li, *A mean-field optimal control formulation of deep learning*, Res. Math. Sci., 6 (2019), art. 10. (Cited on p. 541)

[46] W. E, C. Ma, and L. Wu, *The Barron space and the flow-induced function spaces for neural network models*, Constr. Approx., 55 (2022), pp. 369–406. (Cited on p. 553)

[47] W. E, C. Ma, and L. Wu, *Machine learning from a continuous viewpoint*, I, Sci. China Math., 63 (2020), pp. 2233–2266. (Cited on p. 541)

[48] W. E, C. Ma, and L. Wu, *The generalization error of the minimum-norm solutions for over-parameterized neural networks*, Pure Appl. Funct. Anal., 5 (2020), pp. 1445–1460. (Cited on p. 540)

[49] W. E and B. Yu, *The deep Ritz method: A deep learning-based numerical algorithm for solving variational problems*, Commun. Math. Stat., 6 (2018), pp. 1–12. (Cited on p. 538)

[50] L. C. Evans, *Partial Differential Equations*, Grad. Ser. Math. 19, AMS, Providence, RI, 2010. (Cited on p. 555)

[51] Y. Fan and L. Ying, *Solving electrical impedance tomography with deep learning*, J. Comput. Phys., 404 (2020), pp. 109–119. (Cited on p. 540)

[52] J. Feliu-Faba, Y. Fan, and L. Ying, *Meta-learning pseudo-differential operators with deep neural networks*, J. Comput. Phys., 408 (2020), art. 109309. (Cited on p. 540)

[53] F. Ferraty, A. Mas, and P. Vieu, *Nonparametric regression on functional data: Inference and practical aspects*, Aust. N. Z. J. Stat., 49 (2007), pp. 267–286. (Cited on p. 540)

[54] B. Fornberg, *A Practical Guide to Pseudospectral Methods*, Vol. 1, Cambridge University Press, Cambridge, UK, 1998. (Cited on p. 558)

[55] N. R. Franco and S. Brugiapaglia, *A Practical Existence Theorem for Reduced Order Models Based on Convolutional Autoencoders*, preprint, arXiv:2402.00435, 2024. (Cited on pp. 538, 539)

[56] N. R. Franco, S. Fresca, A. Manzoni, and P. Zunino, *Approximation bounds for convolutional neural networks in operator learning*, Neural Networks, 161 (2023), pp. 129–141. (Cited on p. 538)

[57] N. R. Franco, D. Fraulin, A. Manzoni, and P. Zunino, *On the Latent Dimension of Deep Autoencoders for Reduced Order Modeling of PDEs Parametrized by Random Fields*,

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

preprint, arXiv:2310.12095, 2023. (Cited on p. 538)

[58] H. Gao, J.-X. Wang, and M. J. Zahr, *Non-intrusive model reduction of large-scale, nonlinear dynamical systems using deep learning*, Phys. D, 412 (2020), art. 132614. (Cited on p. 538)

[59] R. Geelen, S. Wright, and K. Willcox, *Operator inference for non-intrusive model reduction with quadratic manifolds*, Comput. Methods Appl. Mech. Eng., 403 (2023), art. 115717. (Cited on p. 538)

[60] M. Geist, P. Petersen, M. Raslan, R. Schneider, and G. Kutyniok, *Numerical solution of the parametric diffusion equation by deep neural networks*, J. Sci. Comput., 88 (2021), art. 22. (Cited on p. 539)

[61] D. Gilbarg and N. S. Trudinger, *Elliptic Partial Differential Equations of Second Order*, Springer, New York, 2015. (Cited on p. 555)

[62] C. R. Gin, D. E. Shea, S. L. Brunton, and J. N. Kutz, *DeepGreen: Deep learning of Green's functions for nonlinear boundary value problems*, Sci. Rep., 11 (2021), art. 21614. (Cited on p. 540)

[63] L. Gonon, L. Grigoryeva, and J. Ortega, *Approximation bounds for random neural networks and reservoir systems*, Ann. Appl. Probab., 33 (2023), pp. 28–69. (Cited on p. 540)

[64] R. Gonzalez-Garcia, R. Rico-Martinez, and I. Kevrekidis, *Identification of distributed parameter systems: A neural net based approach*, Computers Chem. Engrg., 22 (1998), pp. S965–S968. (Cited on p. 539)

[65] M. Griebel and C. Rieger, *Reproducing kernel Hilbert spaces for parametric partial differential equations*, SIAM/ASA J. Uncertain. Quantif., 5 (2017), pp. 111–137, https://doi.org/10.1137/15M1026870. (Cited on p. 541)

[66] E. Haber and L. Ruthotto, *Stable architectures for deep neural networks*, Inverse Problems, 34 (2017), art. 014004. (Cited on p. 541)

[67] A. Hashemi, H. Schaeffer, R. Shi, U. Topcu, G. Tran, R. Ward, *Generalization bounds for sparse random feature expansions*, Appl. Comput. Harmon. Anal., 62 (2023), pp. 310–330. (Cited on p. 540)

[68] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, New York, 2009. (Cited on p. 543)

[69] L. Herrmann, C. Schwab, and J. Zech, *Neural and Spectral Operator Surrogates: Unified Construction and Expression Rate Bounds*, preprint, arXiv:2207.04950, 2022. (Cited on pp. 539, 540)

[70] J. S. Hesthaven and S. Ubbiali, *Non-intrusive reduced order modeling of nonlinear problems using neural networks*, J. Comput. Phys., 363 (2018), pp. 55–78. (Cited on p. 538)

[71] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE Constraints*, Math. Model. Theory Appl. 23, Springer, New York, 2008. (Cited on p. 541)

[72] D. Z. Huang, N. H. Nelsen, and M. Trautner, *An Operator Learning Perspective on Parameter-to-Observable Maps*, preprint, arXiv:2402.06031, 2024. (Cited on pp. 538, 539, 540)

[73] Y. Ingster and N. Stepanova, *Estimation and detection of functions from anisotropic Sobolev classes*, Electron. J. Stat., 5 (2011), pp. 484–506. (Cited on p. 540)

[74] Y. Ingster and N. Stepanova, *Estimation and detection of functions from weighted tensor product spaces*, Math. Methods Stat., 18 (2009), pp. 310–340. (Cited on p. 540)

[75] A. Jacot, F. Gabriel, and C. Hongler, *Neural tangent kernel: Convergence and generalization in neural networks*, in Advances in Neural Information Processing Systems, 2018, pp. 8571–8580. (Cited on pp. 540, 548)

[76] J. Jin, Y. Lu, J. Blanchet, and L. Ying, *Minimax optimal kernel operator learning via multilevel training*, in the Eleventh International Conference on Learning Representations, 2022. (Cited on p. 540)

[77] H. Kadri, E. Duflos, P. Preux, S. Canu, A. Rakotomamonjy, and J. Audiffren, *Operator-valued kernels for learning from functional response data*, J. Mach. Learn. Res., 17 (2016), pp. 613–666. (Cited on pp. 539, 541, 545, 549)

[78] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, *Physics-informed machine learning*, Nat. Rev. Phys., 3 (2021), pp. 422–440. (Cited on p. 538)

[79] A.-K. Kassam and L. N. Trefethen, *Fourth-order time-stepping for stiff PDEs*, SIAM J. Sci. Comput., 26 (2005), pp. 1214–1233, https://doi.org/10.1137/S1064827502410633. (Cited on p. 558)

[80] R. Kempf, H. Wendland, and C. Rieger, *Kernel-based reconstructions for parametric PDEs*, in International Workshop on Meshfree Methods for Partial Differential Equations, Springer, New York, 2017, pp. 53–71. (Cited on p. 541)

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

[81] Y. Khoo, J. Lu, and L. Ying, *Solving parametric PDE problems with artificial neural networks*, European J. Appl. Math., 32 (2021), pp. 421–435. (Cited on p. 539)

[82] A. Kiselev, F. Nazarov, and R. Shterenberg, *Blow up and regularity for fractal Burgers' equation*, Dyn. Partial Differ. Equ., 5 (2008), pp. 211–240. (Cited on p. 554)

[83] Y. Korolev, *Two-layer neural networks with values in a Banach space*, SIAM J. Math. Anal., 54 (2022), pp. 6358–6389, https://doi.org/10.1137/21M1458144. (Cited on pp. 539, 553)

[84] V. Kostic, P. Novelli, A. Maurer, C. Ciliberto, L. Rosasco, and M. Pontil, *Learning dynamical systems via Koopman operator regression in reproducing kernel Hilbert spaces*, Adv. Neural Inform. Process. Syst., 35 (2022), pp. 4017–4031. (Cited on p. 540)

[85] N. B. Kovachki, S. Lanthaler, and S. Mishra, *On universal approximation and error bounds for Fourier neural operators*, J. Mach. Learn. Res., 22 (2021), pp. 1–76. (Cited on p. 539)

[86] N. B. Kovachki, S. Lanthaler, and A. M. Stuart, *Operator Learning: Algorithms and Analysis*, preprint, arXiv:2402.15715, 2024. (Cited on pp. 538, 539, 542, 552)

[87] N. B. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. M. Stuart, A. Anandkumar, *Neural operator: Learning maps between function spaces with applications to PDEs*, J. Mach. Learn. Res., 24 (2023), pp. 1–97. (Cited on pp. 539, 547)

[88] G. Kutyniok, P. Petersen, M. Raslan, and R. Schneider, *A theoretical analysis of deep neural networks and parametric PDEs*, Constr. Approx., 55 (2022), pp. 73–125. (Cited on p. 539)

[89] S. Lanthaler, *Operator learning with PCA-Net: Upper and lower complexity bounds*, J. Mach. Learn. Res., 24 (2023). pp. 1–67. (Cited on p. 539)

[90] S. Lanthaler, Z. Li, and A. M. Stuart, *The Nonlocal Neural Operator: Universal Approximation*, preprint, arXiv:2304.13221, 2023. (Cited on p. 539)

[91] S. Lanthaler, S. Mishra, and G. E. Karniadakis, *Error estimates for DeepONets: A deep learning framework in infinite dimensions*, Trans. Math. Appl., 6 (2022), art. tnac001. (Cited on p. 539)

[92] S. Lanthaler and N. H. Nelsen, *Error bounds for learning with vector-valued random features*, Adv. Neural Inform. Process. Syst., 36 (2023), pp. 71834–71861. (Cited on pp. 540, 543, 551, 552, 553)

[93] S. Lanthaler and A. M. Stuart, *The Parametric Complexity of Operator Learning*, preprint, arXiv:2306.15924, 2023. (Cited on p. 539)

[94] K. Lee and K. T. Carlberg, *Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders*, J. Comput. Phys., 404 (2020), art. 108973. (Cited on pp. 538, 540, 554)

[95] Y. Li, J. Lu, and A. Mao, *Variational training of neural network approximations of solution maps for physical models*, J. Comput. Phys., 409 (2020), art. 109338. (Cited on pp. 539, 540)

[96] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. M. Stuart, and A. Anandkumar, *Fourier neural operator for parametric partial differential equations*, in International Conference on Learning Representations, 2021. (Cited on pp. 539, 547, 560)

[97] Z. Li, N. B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. M. Stuart, and A. Anandkumar, *Neural Operator: Graph Kernel Network for Partial Differential Equations*, preprint, arXiv:2003.03485, 2020. (Cited on pp. 539, 560, 562)

[98] Z. Li, J. Ton, D. Oglic, and D. Sejdinovic, *Towards a unified analysis of random Fourier features*, J. Mach. Learn. Res., 22 (2021), pp. 1–51. (Cited on p. 540)

[99] H. Liu, H. Yang, M. Chen, T. Zhao, and W. Liao, *Deep nonparametric estimation of operators between infinite dimensional spaces*, J. Mach. Learn. Res., 25 (2024), pp. 1–67. (Cited on p. 540)

[100] Z. Long, Y. Lu, X. Ma, and B. Dong, *PDE-Net: Learning PDEs from data*, in Proceedings of the 35th International Conference on Machine Learning, PMLR, 2018, pp. 3208–3216. (Cited on p. 539)

[101] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, *DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators*, Nat. Mach. Intell., 3 (2021), pp. 218–229. (Cited on pp. 539, 547)

[102] D. G. Luenberger, *Optimization by Vector Space Methods*, John Wiley & Sons, New York, 1997. (Cited on p. 550)

[103] D. Luo, T. O'Leary-Roseberry, P. Chen, and O. Ghattas, *Efficient PDE-Constrained Optimization under High-Dimensional Uncertainty Using Derivative-Informed Neural Operators*, preprint, arXiv:2305.20053, 2023. (Cited on p. 538)

[104] B. Matérn, *Spatial Variation*, Lecture Notes in Statist. 36, Springer, Cham, 2013. (Cited on p. 554)

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

[105] S. Mei, T. Misiakiewicz, and A. Montanari, *Generalization error of random feature and kernel methods: Hypercontractivity and kernel matrix concentration*, Appl. Comput. Harmon. Anal., 59 (2022), pp. 3–84. (Cited on p. 540)

[106] H. N. Mhaskar and N. Hahm, *Neural networks for functional approximation and system identification*, Neural Comput., 9 (1997), pp. 143–159. (Cited on p. 539)

[107] C. A. Micchelli and M. Pontil, *On learning vector-valued functions*, Neural Comput., 17 (2005), pp. 177–204. (Cited on pp. 539, 544, 545, 549)

[108] H. Q. Minh, *Operator-Valued Bochner Theorem, Fourier Feature Maps for Operator-Valued Kernels, and Vector-Valued Learning*, preprint, arXiv:1608.05639, 2016. (Cited on p. 541)

[109] M. Mollenhauer, N. Mücke, and T. J. Sullivan, *Learning Linear Operators: Infinite-Dimensional Regression as a Well-Behaved Non-compact Inverse Problem*, preprint, arXiv:2211.08875, 2022. (Cited on p. 540)

[110] R. M. Neal, *Priors for infinite networks*, in Bayesian Learning for Neural Networks, Springer, New York, 1996, pp. 29–53. (Cited on pp. 540, 548)

[111] N. H. Nelsen and A. M. Stuart, *The random feature model for input-output maps between Banach spaces*, SIAM J. Sci. Comput., 43 (2021), pp. A3212–A3243, https://doi.org/10.1137/20M133957X. (Cited on pp. 535, 539, 544, 546, 547, 549)

[112] N. H. Nelsen, *Operator-valued kernels*, in ACM 204: Matrix Analysis, J. A. Tropp, ed., CMS Lecture Notes, California Institute of Technology, Pasadena, 2022, pp. 286–297. (Cited on p. 544)

[113] T. O'Leary-Roseberry, U. Villa, P. Chen, and O. Ghattas, *Derivative-informed projected neural networks for high-dimensional parametric maps governed by PDEs*, Comput. Methods Appl. Mech. Eng., 388 (2022), art. 114199. (Cited on p. 539)

[114] J. Oliva, B. Póczos, and J. Schneider, *Distribution to distribution regression*, in International Conference on Machine Learning, 2013, pp. 1049–1057. (Cited on p. 540)

[115] J. Oliva, W. Neiswanger, B. Póczos, E. Xing, H. Trac, S. Ho, and J. Schneider, *Fast function to function regression*, Proc. 18th International Conference on Artificial Intelligence and Statistics, 2015, pp. 717–725. (Cited on p. 541)

[116] H. Owhadi, *Do ideas have shape? Idea registration as the continuous limit of artificial neural networks*, Phys. D, 444 (2022), art. 133592. (Cited on p. 545)

[117] J. A. Opschoor, C. Schwab, and J. Zech, *Deep Learning in High Dimension: ReLU Network Expression Rates for Bayesian PDE Inversion*, SAM Research Report 2020-47, ETH, Zürich, 2020. (Cited on p. 539)

[118] R. G. Patel and O. Desjardins, *Nonlinear Integro-Differential Operator Regression with Neural Networks*, preprint, arXiv:1810.08552, 2018. (Cited on p. 554)

[119] R. G. Patel, N. A. Trask, M. A. Wood, and E. C. Cyr, *A physics-informed operator regression framework for extracting data-driven continuum models*, Comput. Methods Appl. Mech. Eng., 373 (2021), art. 113500. (Cited on pp. 539, 554)

[120] B. Peherstorfer, K. Willcox, and M. Gunzburger, *Survey of multifidelity methods in uncertainty propagation, inference, and optimization*, SIAM Rev., 60 (2018), pp. 550–591, https://doi.org/10.1137/16M1082469. (Cited on p. 538)

[121] E. Qian, B. Kramer, B. Peherstorfer, and K. Willcox, *Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems*, Phys. D, 406 (2020), art. 132401. (Cited on p. 538)

[122] A. Rahimi and B. Recht, *Random features for large-scale kernel machines*, in Advances in Neural Information Processing Systems, 2008, pp. 1177–1184. (Cited on pp. 540, 541, 542, 548)

[123] A. Rahimi and B. Recht, *Uniform approximation of functions with random bases*, in Proceedings of the 46th Annual Allerton Conference on Communication, Control, and Computing, IEEE, 2008, pp. 555–561. (Cited on pp. 540, 542, 548)

[124] A. Rahimi and B. Recht, *Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning*, in Advances in Neural Information Processing Systems 21, 2008, pp. 1313–1320. (Cited on pp. 540, 548)

[125] M. Raissi, P. Perdikaris, and G. E. Karniadakis, *Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations*, J. Comput. Phys., 378 (2019), pp. 686–707. (Cited on p. 538)

[126] J. O. Ramsay and C. Dalzell, *Some tools for functional data analysis*, J. R. Stat. Soc. Series B Stat. Methodol., 53 (1991), pp. 539–561. (Cited on p. 539)

[127] R. Rico-Martinez, K. Krischer, I. Kevrekidis, M. Kube, and J. Hudson, *Discrete- vs. continuous-time nonlinear signal processing of Cu electrodissolution data*, Chem. Engrg. Commun., 118 (1992), pp. 25–48. (Cited on p. 539)

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 08/08/24 to 131.215.250.149 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

[128] F. Rossi and B. Conan-Guez, *Functional multi-layer perceptron: A non-linear tool for functional data analysis*, Neural Networks, 18 (2005), pp. 45–60. (Cited on p. 539)

[129] A. Rudi and L. Rosasco, *Generalization properties of learning with random features*, in NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems, Curran Associates, 2017, pp. 3218–3228. (Cited on pp. 540, 553)

[130] L. Ruthotto and E. Haber, *Deep neural networks motivated by partial differential equations*, J. Math. Imaging Vision, 62 (2020), pp. 352–364. (Cited on p. 541)

[131] N. D. Santo, S. Deparis, and L. Pegolotti, *Data driven approximation of parametrized PDEs by reduced basis and neural networks*, J. Comput. Phys., 416 (2020), art. 109550. (Cited on pp. 538, 540)

[132] F. Schäfer and H. Owhadi, *Sparse recovery of elliptic solvers from matrix-vector products*, SIAM J. Sci. Comput., 46 (2024), pp. A998–A1025, https://doi.org/10.1137/22M154226X. (Cited on p. 540)

[133] C. Schwab and J. Zech, *Deep learning in high dimension: Neural network expression rates for generalized polynomial chaos expansions in UQ*, Anal. Appl., 17 (2019), pp. 19–55. (Cited on p. 539)

[134] J. Sirignano and K. Spiliopoulos, *DGM: A deep learning algorithm for solving partial differential equations*, J. Comput. Phys., 375 (2018), pp. 1339–1364. (Cited on p. 538)

[135] P. D. Spanos and R. Ghanem, *Stochastic finite element expansion for random media*, J. Engrg. Mech., 115 (1989), pp. 1035–1053. (Cited on p. 539)

[136] G. Stepaniants, *Learning partial differential equations in reproducing kernel Hilbert spaces*, J. Mach. Learn. Res., 24 (2023), pp. 1–72. (Cited on p. 540)

[137] B. Stevens and T. Colonius, *FiniteNet: A Fully Convolutional LSTM Network Architecture for Time-Dependent Partial Differential Equations*, preprint, arXiv:2002.03014, 2020. (Cited on p. 539)

[138] A. M. Stuart, *Inverse problems: A Bayesian perspective*, Acta Numer., 19 (2010), pp. 451–559. (Cited on pp. 538, 541)

[139] Y. Sun, A. Gilbert, A. Tewari, *But how does it work in theory? Linear SVM with random features*, in Advances in Neural Information Processing Systems, Vol. 31, Curran Associates, 2018. (Cited on p. 540)

[140] Y. Sun, A. Gilbert, and A. Tewari, *On the Approximation Properties of Random ReLU Features*, preprint, arXiv:1810.04374, 2019. (Cited on pp. 540, 548)

[141] T. Suzuki and S. Akiyama, *Benefit of deep learning with non-convex noisy gradient descent: Provable excess risk bound and superiority to kernel methods*, in International Conference on Learning Representations, 2021. (Cited on p. 550)

[142] N. Trask, R. G. Patel, B. J. Gross, and P. J. Atzberger, *GMLS-Nets: A Framework for Learning from Unstructured Data*, preprint, arXiv:1909.05371, 2019. (Cited on p. 539)

[143] R. K. Tripathy and I. Bilionis, *Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification*, J. Comput. Phys., 375 (2018), pp. 565–588. (Cited on p. 539)

[144] A. Townsend and L. N. Trefethen, *An extension of Chebfun to two dimensions*, SIAM J. Sci. Comput., 35 (2013), pp. C495–C518, https://doi.org/10.1137/130908002. (Cited on p. 541)

[145] A. Townsend and L. N. Trefethen, *Continuous analogues of matrix factorizations*, Proc. A, 471 (2015), art. 20140585 (Cited on p. 541)

[146] H. Wendland, *Scattered Data Approximation*, Cambridge, Monogr. Appl. Comput. Math. 17, Cambridge University Press, Cambridge, UK, 2004. (Cited on pp. 539, 540, 544)

[147] C. K. Williams, *Computing with infinite networks*, in Advances in Neural Information Processing Systems, 1997, pp. 295–301. (Cited on pp. 540, 548)

[148] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*, Vol. 2, MIT Press, Cambridge, MA, 2006. (Cited on pp. 539, 540)

[149] N. Winovich, K. Ramani, and G. Lin, *ConvPDE-UQ: Convolutional neural networks with quantified uncertainty for heterogeneous elliptic partial differential equations on varied domains*, J. Comput. Phys., 394 (2019), pp. 263–279. (Cited on p. 539)

[150] K. Wu and D. Xiu, *Data-driven deep learning of partial differential equations in modal space*, J. Comput. Phys., 408 (2020), art. 109307. (Cited on pp. 539, 559)

[151] G. Yehudai and O. Shamir, *On the power and limitations of random features for understanding neural networks*, in Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, 2019. (Cited on p. 548)

[152] Y. Zhu and N. Zabaras, *Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification*, J. Comput. Phys., 366 (2018), pp. 415–447. (Cited on p. 539)

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.