SIAM J. SCI. COMPUT.
Vol. 43, No. 5, pp. A3212–A3243

THE RANDOM FEATURE MODEL FOR INPUT-OUTPUT MAPS BETWEEN BANACH SPACES*

NICHOLAS H. NELSEN† AND ANDREW M. STUART†

Abstract. Well known to the machine learning community, the random feature model is a parametric approximation to kernel interpolation or regression methods. It is typically used to approximate functions mapping a finite-dimensional input space to the real line. In this paper, we instead propose a methodology for use of the random feature model as a data-driven surrogate for operators that map an input Banach space to an output Banach space. Although the methodology is quite general, we consider operators defined by partial differential equations (PDEs); here, the inputs and outputs are themselves functions, with the input parameters being functions required to specify the problem, such as initial data or coefficients, and the outputs being solutions of the problem. Upon discretization, the model inherits several desirable attributes from this infinite-dimensional viewpoint, including mesh-invariant approximation error with respect to the true PDE solution map and the capability to be trained at one mesh resolution and then deployed at different mesh resolutions. We view the random feature model as a nonintrusive data-driven emulator, provide a mathematical framework for its interpretation, and demonstrate its ability to efficiently and accurately approximate the nonlinear parameter-to-solution maps of two prototypical PDEs arising in physical science and engineering applications: the viscous Burgers' equation and a variable coefficient elliptic equation.

Key words. random feature, surrogate model, emulator, solution map, high-dimensional approximation, model reduction, parametric PDE, supervised learning, data-driven computing

AMS subject classifications. 65D15, 65D40, 62M45, 35R60

DOI. 10.1137/20M133957X

1. Introduction.

The *random feature model* (RFM), an architecture for the data-driven approximation of maps between finite-dimensional spaces, was formalized in [70, 71, 72], building on earlier precursors in [6, 64, 89]. The goal of this paper is to extend the RFM to a methodology for the data-driven approximation of maps between infinite-dimensional spaces. Canonical examples of such maps include the semigroup generated by a time-dependent partial differential equation (PDE) mapping the initial condition (an input parameter) to the solution at a later time and the operator mapping a coefficient function (an input parameter) appearing in a PDE to its solution. Obtaining efficient and potentially low-dimensional representations of PDE solution maps is not only conceptually interesting but also practically useful. Many applications in science and engineering require repeated evaluations of a complex and expensive forward model for different configurations of a system parameter. The model often represents a discretized PDE and the parameter, serving as input to the model, often represents a high-dimensional discretized quantity such as an initial condition or uncertain coefficient field. These *outer loop* applications commonly arise in inverse problems or uncertainty quantification tasks that involve

*Submitted to the journal's Methods and Algorithms for Scientific Computing section May 21, 2020; accepted for publication (in revised form) May 20, 2021; published electronically September 20, 2021.

https://doi.org/10.1137/20M133957X

Funding: The work of the first author was supported by the National Science Foundation (NSF) Graduate Research Fellowship Program under award DGE-1745301. The work of the second author was supported by NSF grant DMS-1818977 and by the Office of Naval Research (ONR) through grant N00014-17-1-2079. This work was supported by NSF grant AGS-1835860 and ONR grant N00014-19-1-2408.

†Division of Engineering and Applied Science, California Institute of Technology, Pasadena, CA 91125 USA (nnelsen@caltech.edu, astuart@caltech.edu).

control, optimization, or inference [69]. Full order forward models do not perform well in such many-query contexts, either due to excessive computational cost (requiring the most powerful high performance computing architectures) or slow evaluation time (unacceptable in real-time contexts such as on-the-fly optimal control). In contrast to that of the *big data* regime that dominates computer vision and other technological fields, only a relatively small amount of high resolution data can be generated from computer simulations or physical experiments in scientific applications. Fast approximate solvers built from this limited available data that can efficiently and accurately emulate the full order model would be highly advantageous.

In this work, we demonstrate that the RFM holds considerable potential for such a purpose. Resembling [58, 92] and the contemporaneous work in [13, 51, 56, 65], we present a methodology for true function space learning of black-box input-output maps between a Banach space and separable Hilbert space. We formulate the approximation problem as supervised learning in infinite dimensions and show that the natural hypothesis space is a reproducing kernel Hilbert space (RKHS) associated with an operator-valued kernel. For a suitable loss functional, training the RFM is equivalent to solving a finite-dimensional convex optimization problem. As a consequence of our careful construction of the method as mapping between Banach spaces, the resulting emulator naturally scales favorably with respect to (w.r.t.) the high input and output dimensions arising in practical, discretized applications; furthermore, it is shown to achieve small relative test error for two model problems arising from approximation of a semigroup and of the solution map corresponding to an elliptic PDE exhibiting parametric dependence on a coefficient function.

1.1. Literature review. In recent years, two different lines of research have emerged that address PDE approximation problems with 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) to solve a prescribed initial boundary value problem with as high accuracy as possible. Given a point cloud in a spatio-temporal domain $\tilde{D}$ 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 (see [73, 79, 87] and the references therein). To clarify, the object approximated with this novel method is a *low-dimensional* input-output map $\tilde{D} \to \mathbb{R}$, i.e., the real-valued function that solves the PDE. This approach is mesh-free by definition but 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 solution. We do not explore this first approach any further in this article.

The second direction is arguably more ambitious: use an NN as an emulator for the infinite-dimensional mapping between an input parameter and the PDE solution itself or a functional of the solution, i.e., a quantity of interest; the latter is widely prevalent in uncertainty quantification problems. We emphasize that the object approximated in this setting, unlike in the aforementioned first approach, is an input-output map $\mathcal{X} \to \mathcal{Y}$, i.e., the PDE solution operator, where $\mathcal{X}$, $\mathcal{Y}$ are infinite-dimensional Banach spaces; this map is generally nonlinear. For an approximation-theoretic treatment of parametric PDEs in general, we refer the reader to the article of Cohen and DeVore [23]. In applications, the solution operator is represented by a discretized forward model $\mathbb{R}^K \to \mathbb{R}^K$, where K is the mesh size, and hence represents a *high-dimensional* object. It is this second line of research that inspires our work.

Of course, there are many approaches to forward model reduction that do not explicitly involve machine learning ideas. The reduced basis method (see [5, 9, 29] and the references therein) is a classical idea based on constructing an empirical basis from 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. Recently, 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) type, have further improved the applicability of these methods [21, 36, 43, 53, 77]. However, the input-output maps considered in these works involve high dimension in only one of the input or the output space, not both. Other popular surrogate modeling techniques include Gaussian processes [90], polynomial chaos expansions [80], and radial basis functions [88], yet these are only practically suitable for problems with input space of low to moderate dimension. Classical numerical methods for PDEs may also represent the forward model $\mathbb{R}^K \to \mathbb{R}^K$, 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.

Instead, deep NNs have been identified as strong candidate surrogate models for parametric PDE problems due to their empirical ability to emulate high-dimensional nonlinear functions with minimal evaluation cost once trained. 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 [20, 39, 74]. There are now more theoretical justifications for NNs breaking the *curse of dimensionality* [51, 52, 61], leading to increased interest in PDE applications [1, 37, 66, 78]. A suite of work on data-driven discretizations of PDEs has surfaced that allows for identification of the governing model [4, 14, 57, 68, 81, 83]; however, we note that only the operators appearing in the equation itself are approximated with these approaches, not the solution operator of the PDE. More in line with our focus in this article, architectures based on deep convolutional NNs have proven quite successful for learning elliptic PDE solution maps (for example, see [84, 91, 93], which take an image-to-image regression approach). Other NNs have been used in similar elliptic problems for quantity of interest prediction [49], error estimation [19], or unsupervised learning [54]. Yet in all the approaches above, 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 [20, 75] or, for functional data analysis, [46, 63]). The aforementioned reduced basis method is an example, as is the method of [22, 23], which approximates the solution map with sparse Taylor polynomials and is proved to achieve optimal convergence rates in idealized settings. However, it is only recently that machine learning methods have been explicitly designed to operate in an infinite-dimensional setting, and there is little work in this direction [13, 56]. Here we propose the RFM as another such method.

The RFM [70, 71, 72], detailed in subsection 2.3, 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. These *random features* could be defined, for example, by randomizing the internal parameters of an NN. Compared to NN emulators with enormous learnable parameter counts (e.g., $O(10^5)$ to $O(10^6)$; see [33, 34, 54]) and methods that are intrusive or lead to nontrivial

implementations [22, 53, 77], the RFM is one of the simplest models to formulate and train (often $O(10^3)$ parameters, or fewer, suffice). The theory of the RFM for real-valued outputs is well developed, partly due to its close connection to kernel methods [3, 16, 45, 70, 88] and Gaussian processes [64, 89], and includes generalization rates and dimension-free estimates [61, 71, 82]. A quadrature viewpoint on the RFM provides further insight and leads to Monte Carlo sampling ideas [3]; we remark on this further in subsection 2.3. As in modern deep learning practice, the RFM has also been shown to perform best when the model is overparametrized [8]. In a similar high-dimensional setting of relevance in this paper, the authors of [40, 48] theoretically investigated nonparametric kernel regression for parametric PDEs with real-valued solution map outputs. The specific random Fourier feature approach of Rahimi and Recht [70] was generalized in [15] to the finite-dimensional matrix-valued kernel setting with vector-valued random Fourier features. However, most of these works require explicit knowledge of the kernel itself. Here our viewpoint is to work directly with random features as the basis for a standalone method, choosing them for their properties and noting that they implicitly define a kernel, but not working directly with this kernel; furthermore, our work considers both infinite-dimensional input *and* output spaces, not just one or the other. A key idea underlying our approach is to formulate the proposed random feature algorithm on infinite-dimensional space and only then discretize. This philosophy in algorithm development has been instructive in a number of areas in scientific computing, such as optimization [44] and the development of Markov chain Monte Carlo methodology [25]. It has recently been promoted as a way of designing and analyzing algorithms within machine learning [41, 60, 76, 85, 86], and our work may be understood within this general framework.

1.2. Contributions. Our primary contributions in this paper are now listed.

1. We develop the RFM, directly formulated on the function space level, for learning input-output maps between Banach spaces purely from data. As a method for parametric PDEs, the methodology is non-intrusive but also has the additional advantage that it may be used in settings where only data is available and no model is known.
2. 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.
3. We apply our 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.
4. We demonstrate, by means of numerical experiments, 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.

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, and explain the training procedure. 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 discussion and 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 $\mathcal{X}$, $\mathcal{Y}$ are infinite-dimensional spaces of real-valued functions defined on some bounded open subset of $\mathbb{R}^d$, and $F^\dagger$ is defined

by $a \mapsto F^\dagger(a) \coloneqq u$, where u is the solution of a (possibly time-dependent) PDE and a 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 [70, 71] and generalized here to the infinite-dimensional setting; they have also been shown to work well in other algorithms for high-dimensional numerical approximation [12, 24, 30]. Within the broader scope of reduced order modeling techniques [9], the approach we adopt in this paper falls within the class of data-fit emulators. In its essence, our method interpolates the solution manifold

$$\mathcal{M} = \{u \in \mathcal{Y} \colon u = F^\dagger(a),\, a \in \mathcal{X}\}\,. \tag{2.1}$$

The solution map $F^\dagger$, as the inverse of a differential operator, is often smoothing and admits a notion of compactness, i.e., the output space compactly embeds into the input space. Then, the idea is that $\mathcal{M}$ should have some compact, low-dimensional structure (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 $\mathcal{M}$ [23]. While such subtleties of approximation theory are crucial to developing rigorous theory and provably convergent algorithms, 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 aforementioned are essentially unavailable.

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 RFM in subsection 2.3. In subsection 2.4, we describe the optimization principle which leads to algorithms for the RFM and an example problem in which $\mathcal{X}$ and $\mathcal{Y}$ are one-dimensional vector spaces.

2.1. Problem formulation. Let $\mathcal{X}$, $\mathcal{Y}$ 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 $a_i \sim \nu$ i.i.d., ν is a probability measure supported on $\mathcal{X}$, and $y_i = F^\dagger(a_i) \sim F^\dagger_\sharp \nu$ with, potentially, noise added to the evaluations of $F^\dagger(\cdot)$. 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 but are not 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 determine parameters in supervised learning is to first define a loss functional $\ell \colon \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ and then minimize the expected risk,

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

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

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

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 [42]. 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 $F^\dagger$. From here on out, 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. For a Borel measurable map $G\colon \mathcal{U}\to\mathcal{V}$ between two Banach spaces $\mathcal{U}$, $\mathcal{V}$ and a probability measure π supported on $\mathcal{U}$, we denote the expectation of G under π by

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

in the sense of Bochner integration (see, e.g., [27, sect. A.2]). We will drop the domain of integration in situations where no confusion is caused by doing so.

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 [2, 10, 26, 88]. 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 [3, 18, 63].

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 $a\in\mathcal{X}$, $k(\cdot,a)\in\mathcal{H}$ 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 [2].

We now introduce the needed 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. *Let $\mathcal{X}$ be a real Banach space and $\mathcal{Y}$ a real separable Hilbert space. An* operator-valued kernel *is a map*

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

where $\mathcal{L}(\mathcal{Y},\mathcal{Y})$ 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, $a'\in\mathcal{X}$ and for every $N\in\mathbb{N}$,

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

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 $\mathcal{H}_k = \mathcal{H}_k(\mathcal{X};\mathcal{Y})$. Now, choosing a probability measure ν supported on $\mathcal{X}$, we define a kernel integral operator T_k associated to k by

$$\begin{aligned} &T_k\colon L^2_\nu(\mathcal{X};\mathcal{Y})\to L^2_\nu(\mathcal{X};\mathcal{Y})\,,\\ &\qquad F\mapsto T_kF \coloneqq \int k(\cdot,a')F(a')\nu(da')\,, \end{aligned} \tag{2.8}$$

which is nonnegative, self-adjoint, and compact (provided $k(a,a)\in\mathcal{L}(\mathcal{Y},\mathcal{Y})$ is compact for all $a\in\mathcal{X}$ [18]). Let us further assume that all conditions needed for $T_k^{1/2}$ to be an isometry from L^2_ν 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., [3, 10]), we may write the RKHS inner product as

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

Note that while (2.9) appears to depend on the measure ν on $\mathcal{X}$, the RKHS $\mathcal{H}_k$ is itself determined by the kernel without any reference to a measure (see [26, Chap. 3, Thm. 4]). With the inner product now explicit, we may directly deduce a reproducing property. A fully rigorous justification of the methodology is outside the scope of this article; however, we perform formal computations which provide intuition underpinning the methodology. To this end we fix $a\in\mathcal{X}$ and $y\in\mathcal{Y}$. Then

$$\begin{aligned} \langle k(\cdot,a)y, T_k^{-1}F\rangle_{L^2_\nu} &= \int \big\langle k(a',a)y, (T_k^{-1}F)(a')\big\rangle_{\mathcal{Y}}\,\nu(da')\\ &= \int \big\langle y, k(a,a')(T_k^{-1}F)(a')\big\rangle_{\mathcal{Y}}\,\nu(da')\\ &= \left\langle y, \int k(a,a')(T_k^{-1}F)(a')\,\nu(da')\right\rangle_{\mathcal{Y}}\\ &= \langle y, F(a)\rangle_{\mathcal{Y}} \end{aligned}$$

by using Definition 2.2 of operator-valued kernel and the fact that $k(\cdot,a)y\in\mathcal{H}_k$ [18]. So, we deduce the following.

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

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

This identity, paired with a special choice of 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 $\mathcal{X}$, $\mathcal{Y}$ 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 [46, 63], but these all involve some simple generalization of scalar-valued kernels. Instead, the RFM allows one to implicitly work with operator-valued kernels through the use of a *random feature map* $\varphi\colon \mathcal{X} \times \Theta \to \mathcal{Y}$ and a probability measure μ supported on Banach space Θ. The map φ 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 ν is the (sometimes a modeling choice at our discretion, sometimes unknown) data distribution on $\mathcal{X}$. Together, (φ, μ) 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. Given a Hilbert space $(H, \langle\cdot,\cdot\rangle, \|\cdot\|)$, the *outer product* $a \otimes b \in \mathcal{L}(H, H)$ is defined by $(a \otimes b)c = \langle b, c\rangle a$ for any $a, b, c \in H$.

Given the pair (φ, μ), consider maps $k_\mu\colon \mathcal{X} \times \mathcal{X} \to \mathcal{L}(\mathcal{Y}, \mathcal{Y})$ of the form

$$k_\mu(a, a') \coloneqq \int \varphi(a; \theta) \otimes \varphi(a'; \theta)\mu(d\theta)\,. \tag{2.11}$$

Such representations need not be unique; different pairs (φ, μ) may induce the same kernel $k = k_\mu$ in (2.11). Since k_μ 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 approximation theory will be based on this space or 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 $a \in \mathcal{X}$, $y \in \mathcal{Y}$,

$$\begin{aligned}
\langle y, F(a)\rangle_{\mathcal{Y}} = \langle k_\mu(\cdot, a)y, F\rangle_{\mathcal{H}_{k_\mu}} &= \left\langle \int \langle \varphi(a;\theta), y\rangle_{\mathcal{Y}}\, \varphi(\cdot;\theta)\, \mu(d\theta), F \right\rangle_{\mathcal{H}_{k_\mu}} \\
&= \int \langle \varphi(a;\theta), y\rangle_{\mathcal{Y}} \langle \varphi(\cdot;\theta), F\rangle_{\mathcal{H}_{k_\mu}}\, \mu(d\theta) \\
&= \int c_F(\theta) \langle y, \varphi(a;\theta)\rangle_{\mathcal{Y}}\, \mu(d\theta) \\
&= \left\langle y, \int c_F(\theta)\varphi(a;\theta)\mu(d\theta) \right\rangle_{\mathcal{Y}},
\end{aligned}$$

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

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

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

$$F = \int c_F(\theta)\varphi(\cdot;\theta)\mu(d\theta)\,. \tag{2.13}$$

The formal expression (2.12) for $c_F(\theta)$ needs careful interpretation (provided in Appendix B). For instance, if $\varphi(\cdot;\theta)$ is a realization of a Gaussian process as in Example 2.9, then $\varphi(\cdot;\theta) \notin \mathcal{H}_{k_\mu}$ with probability one; indeed, in this case c_F is defined

only as an L^2_μ limit. Nonetheless, the RKHS may be completely characterized by this integral representation. Define the map

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

$\mathcal{A}$ may be shown to be a bounded linear operator that is a particular square root of T_{k_μ} (Appendix B). We have the following result whose proof, provided in Appendix A, is a straightforward generalization of the real-valued case given in [3, sect. 2.2].

RESULT 2.5. *Under the assumption that $\varphi \in L^2_{\nu\times\mu}(\mathcal{X}\times\Theta;\mathcal{Y})$, the RKHS defined by the kernel k_μ in* (2.11) *is precisely*

$$\mathcal{H}_{k_\mu} = \operatorname{im}(\mathcal{A}) = \left\{ \int c(\theta)\varphi(\cdot;\theta)\mu(d\theta)\colon c \in L^2_\mu(\Theta;\mathbb{R}) \right\}. \tag{2.15}$$

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 Appendix B.

A central role in what follows is the approximation of measure μ by the empirical measure

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

Given this, define $k^{(m)} \coloneqq k_{\mu^{(m)}}$ to be the empirical approximation to k_μ:

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

Then we 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 variables. The following characterization of $\mathcal{H}_{k^{(m)}}$ is proved in Appendix A.

RESULT 2.6. *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 the $\{\varphi_j \coloneqq \varphi(\cdot;\theta_j)\}_{j=1}^m$.*

Applying a simple Monte Carlo sampling approach to elements in RKHS (2.15) by replacing probability measure μ by empirical measure $\mu^{(m)}$ gives, for $c \in L^2_\mu$,

$$\frac{1}{m}\sum_{j=1}^m c(\theta_j)\varphi(\cdot;\theta_j) \approx \int c(\theta)\varphi(\cdot;\theta)\mu(d\theta)\,. \tag{2.18}$$

This approximation achieves the Monte Carlo rate $O(m^{-1/2})$ 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_μ, and $\mathcal{H}_{k_\mu}$ are all unknown; only the random feature pair (φ,μ) is assumed to be given. Hence one cannot apply (2.12) (or (B.2)) 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 approximation 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 approximation to $F^\dagger$, also from the space $\mathcal{H}_{k^{(m)}}$. We now define the RFM.

DEFINITION 2.7. *Given probability spaces* $(\mathcal{X}, \mathcal{B}(\mathcal{X}), \nu)$ *and* $(\Theta, \mathcal{B}(\Theta), \mu)$ *with* $\mathcal{X}$, Θ *being real finite- or infinite-dimensional Banach spaces, 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*

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

We use the Borel σ-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 $\alpha \in \mathbb{R}^m$ so as to approximate mappings $F^\dagger \in \mathcal{H}_{k_\mu}$ (in the ideal setting) by mappings $F_m(\cdot;\alpha) \in \mathcal{H}_{k^{(m)}}$. The RFM is itself a random variable and may be viewed as a *spectral method* since the randomized basis $\varphi(\cdot;\theta)$ in the linear expansion (2.19) is defined on all of $\mathcal{X}$ ν-a.e. Determining the coefficient vector α from data obviates the difficulties associated with the Monte Carlo approach since the method only requires knowledge of the pair (φ,μ) and knowledge of sample input-output pairs from target operator $F^\dagger$.

As written, (2.19) is incredibly simple. It is clear that the choice of random feature map and measure pair (φ,μ) will determine the quality of approximation. Most papers deploying these methods, including [15, 70, 71], 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 [72, 82], is the opposite in that we allow the choice of random feature map φ to implicitly *define* the kernel via the formula (2.11) instead of picking the kernel first. This methodology also has implications for numerics: the kernel never explicitly appears in any computations, which leads to memory savings. It does, however, leave open the question of characterizing the universality [82] 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.

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 [64, 82, 89]. To see this explicitly, consider the setting where $\mathcal{X}$, $\mathcal{Y}$ are both equal to the Euclidean space $\mathbb{R}$ and choose φ to be a family of hidden neurons $\varphi_{\text{NN}}(a;\theta) \coloneqq \sigma(\theta^{(1)}\cdot a + \theta^{(2)})$. A single hidden layer NN would seek to find $\{(\alpha_j,\theta_j)\}_{j=1}^m$ in $\mathbb{R}\times\mathbb{R}^2$ so that

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

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 $\varphi_{\text{NN}}(\cdot;\theta)$ to be any deep NN. However, while the RFM has the same *form* as (2.20), there is a difference in the *training*: the θ_j are drawn i.i.d. from a probability measure and then fixed, and only the α_j are chosen to fit the training data. This connection is quite profound: given any deep NN with randomly initialized parameters θ, studies of the lazy training regime and neural tangent kernel [16, 45] suggest that adopting an RFM approach and optimizing over only α is quite natural, as 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 $\{\alpha_j\}_{j=1}^m$ adapt considerably.

Once the feature parameters $\{\theta_j\}_{j=1}^m$ are chosen at random and fixed, training the RFM F_m only requires optimizing over $\alpha \in \mathbb{R}^m$ which, due to linearity of F_m in α, is a straightforward task to which we now turn our attention.

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_μ 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 into complicated black-box computer codes.

We now proceed to show that a regularized version of the optimization problem (2.3)–(2.4) arises naturally from approximation of a nonparametric regression 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 $\{a_i, y_i = F^\dagger(a_i)\}_{i=1}^n \subset \mathcal{X}\times\mathcal{Y}$ as data, with the a_i drawn from (possibly unknown) probability measure ν on $\mathcal{X}$, the objective is to find an approximation $\hat{F}$ to the map $F^\dagger$. Let $\mathcal{H}_{k_\mu}$ be the hypothesis space and k_μ 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 $\hat{F}$ of

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

where $\lambda \geq 0$ is a penalty parameter. By the representer theorem for operator-valued kernels [63, Theorems 2 and 4], the minimizer has the form

$$\hat{F} = \sum_{j=1}^n k_\mu(\cdot, a_j)\beta_j \tag{2.22}$$

for some functions $\{\beta_j\}_{j=1}^n \subset \mathcal{Y}$. 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 from both operation count and memory required. A few workarounds were proposed in [46] 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_μ; in our setting we assume access only to the random feature pair (φ, μ) which defines k_μ and not k_μ 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 Appendix 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. *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 $\hat{\alpha} \in \mathbb{R}^m$ be the unique minimum norm solution of the following problem:*

$$\min_{\alpha\in\mathbb{R}^m} \left\{ \sum_{j=1}^n \frac{1}{2}\left\| y_j - \frac{1}{m}\sum_{\ell=1}^m \alpha_\ell \varphi(a_j;\theta_\ell)\right\|_{\mathcal{Y}}^2 + \frac{\lambda}{2m}\|\alpha\|_2^2 \right\}. \tag{2.23}$$

Then, the RFM defined by this choice $\alpha = \hat{\alpha}$ satisfies

$$F_m(\cdot;\hat{\alpha}) = \operatorname*{argmin}_{F\in\mathcal{H}_{k^{(m)}}} \left\{ \sum_{j=1}^{n} \frac{1}{2}\big\|y_j - F(a_j)\big\|_{\mathcal{Y}}^2 + \frac{\lambda}{2}\big\|F\big\|_{\mathcal{H}_{k^{(m)}}}^2 \right\}. \tag{2.24}$$

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

$$\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{n} \alpha_i \big\langle \varphi(a_j;\theta_i), \varphi(a_j;\theta_\ell)\big\rangle_{\mathcal{Y}} + \lambda\alpha_\ell = \sum_{j=1}^{n} \big\langle y_j, \varphi(a_j;\theta_\ell)\big\rangle_{\mathcal{Y}} \tag{2.25}$$

for each $\ell \in \{1,\dots,m\}$. 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 may be written in terms of a pseudoinverse operator (see [59, sect. 6.11]).

Example 2.9 (Brownian bridge). We now provide a one-dimensional instantiation of the RFM to illustrate the methodology. Take the input space as $\mathcal{X} \coloneqq (0,1)$, output space $\mathcal{Y} \coloneqq \mathbb{R}$, input space measure $\nu \coloneqq U(0,1)$, and random parameter space $\Theta \coloneqq \mathbb{R}^\infty$. 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*

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

where $\theta \coloneqq \{\theta^{(j)}\}_{j\in\mathbb{N}}$ and $\mu \coloneqq N(0,1)\times N(0,1)\times\cdots$. 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:

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

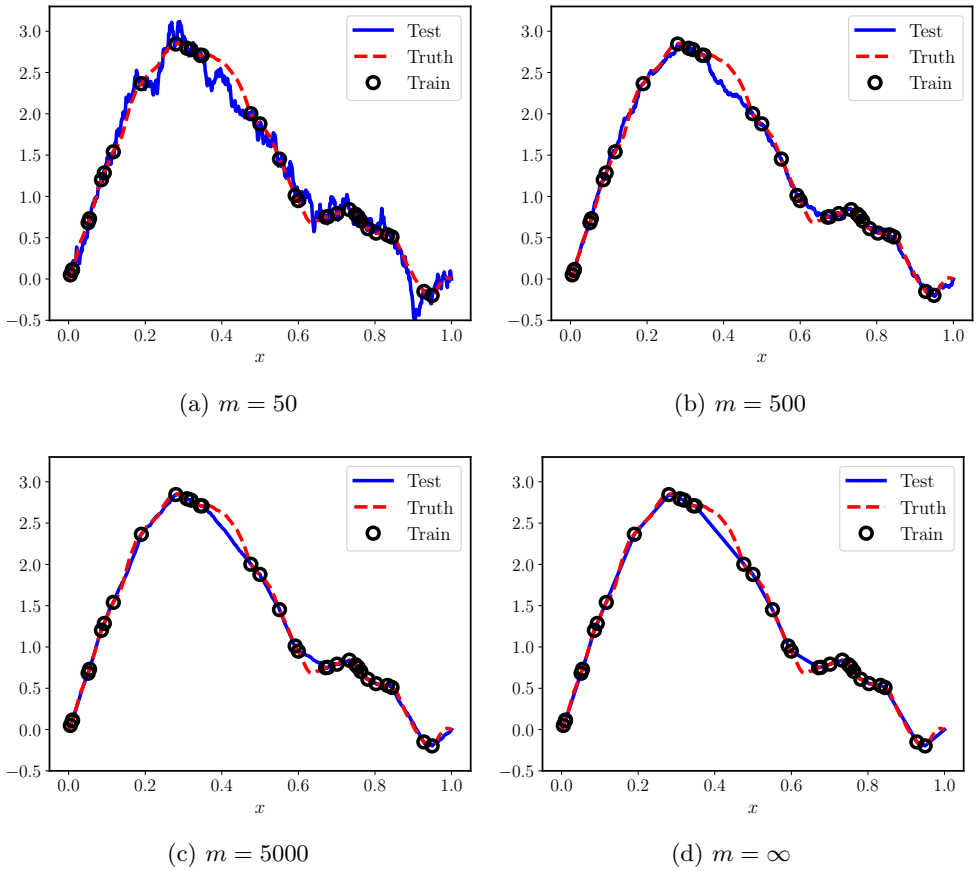
Note that k_μ is the Green's function for the negative Laplacian on $(0,1)$ with Dirichlet boundary conditions. Using this fact, we may explicitly characterize the associated RKHS $\mathcal{H}_{k_\mu}$ as follows. First, we have

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

where the negative Laplacian has domain $H^2((0,1);\mathbb{R}) \cap H_0^1((0,1);\mathbb{R})$. Viewing T_{k_μ} as an operator from $L^2((0,1);\mathbb{R})$ into itself, from (2.9) we conclude, upon integration by parts, that

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

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 $m\to\infty$ (see [61] for a related theoretical result).



(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 (Figure 1(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. But 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.

3. Application to PDE solution maps. In this section, we design the random feature maps $\varphi\colon \mathcal{X} \times \Theta \to \mathcal{Y}$ and measures μ 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 (φ, μ)) 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 (φ, μ). 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 φ that we choose to employ are nonlinear in both arguments. We also detail the probability measure ν on the input space $\mathcal{X}$ for each of the two 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 ν.

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

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

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

$$a \sim \nu \coloneqq N(0, C) \tag{3.2}$$

with Matérn-like covariance operator [31, 62]

$$C \coloneqq \tau^{2\alpha - d}(-\Delta + \tau^2 \operatorname{Id})^{-\alpha}\,, \tag{3.3}$$

where $d = 1$ and the negative Laplacian $-\Delta$ is defined over $\mathbb{T}^1 = [0,1]_{\text{per}}$ 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 $\alpha - 1/2$ [27, Thm. 12, p. 338], so in particular $a \in \mathcal{X} \coloneqq L^2(\mathbb{T}^1; \mathbb{R})$. Then for all $\varepsilon > 0$, the unique global solution $u(t, \cdot)$ to (3.1) is real analytic for all $t > 0$ (see [50, Thm. 1.1]). Hence, setting the output space to be $\mathcal{Y} \coloneqq H^s(\mathbb{T}^1; \mathbb{R})$ for any $s > 0$, we may define the solution map

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

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

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

where $\sigma(\cdot)$, the ELU function defined below, is defined as a mapping on $\mathbb{R}$ and applied pointwise to functions. Viewing $\Theta \subseteq L^2(\mathbb{T}^1;\mathbb{R})$, the randomness enters through $\theta \sim \mu := N(0, C')$ with 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

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

where δ, $\beta > 0$. 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 φ. Although simply hand-tuned for performance and not optimized, the filter χ 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).

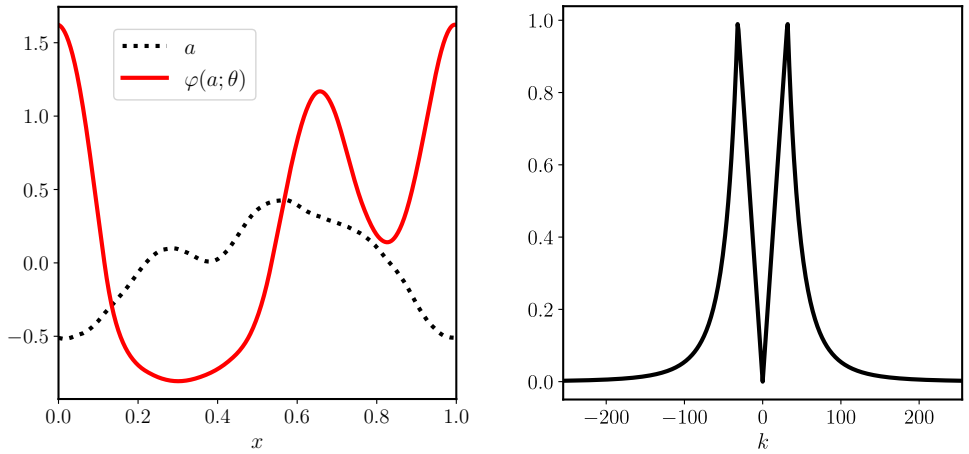
We choose the activation function σ in (3.5) to be the exponential linear unit

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

ELU has successfully been used as activation in other machine learning frameworks for related nonlinear PDE problems [53, 67, 68]. We also find ELU to perform better in the RFM framework over several other choices including ReLU$(\cdot)$, tanh$(\cdot)$, sigmoid$(\cdot)$, sin$(\cdot)$, SELU$(\cdot)$, and softplus$(\cdot)$. Note that the pointwise evaluation of ELU 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.

3.2. Darcy flow: Formulation. Divergence form elliptic equations [38] arise in a variety of applications, in particular, the groundwater flow in a porous medium governed by Darcy's law [7]. This linear elliptic boundary value problem reads

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



(a) (b)

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

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 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 ν on a is a high contrast level set prior constructed as the pushforward of a Gaussian measure:

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

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

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

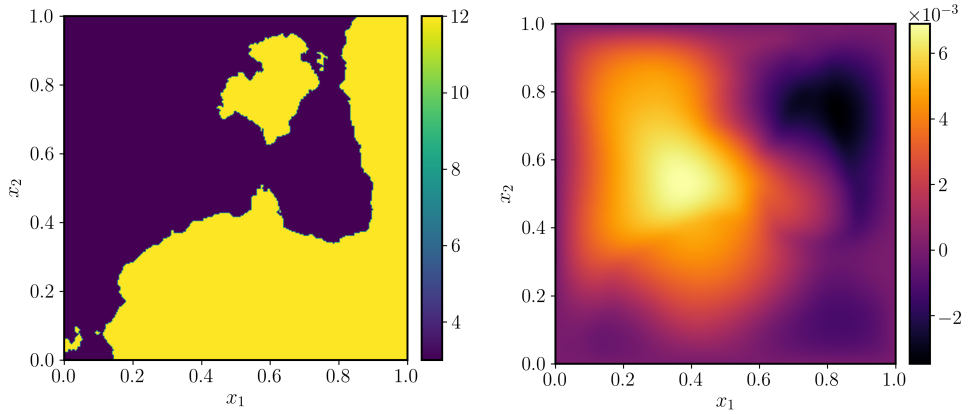
applied pointwise to functions, and the covariance operator C is given in (3.3) with $d = 2$ and homogeneous Neumann boundary conditions on $-\Delta$. That is, the resulting coefficient a almost surely takes only two values (a^+ or a^-) and, as the zero level set of a Gaussian random field, exhibits random geometry in the physical domain D. It follows that $a \in L^\infty(D; \mathbb{R}_{\ge 0})$ 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 [11]. 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 $a \in \mathcal{X} \coloneqq L^\infty(D; \mathbb{R}_{\ge 0})$, there exists a unique weak solution $u \in \mathcal{Y} \coloneqq H_0^1(D; \mathbb{R})$ for (3.8) (see, e.g., Evans [32]). Thus, we define the ground truth solution map

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

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 $\varphi(a; \theta) \coloneqq p_1$ such that



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

FIG. 3. *Random feature map construction for Darcy flow: Figure* 3(a) *displays a representative input draw a with $\tau = 3$, $\alpha = 2$ and $a^+ = 12$, $a^- = 3$; Figure* 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 $\delta = 0.15$.*

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

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

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

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

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

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

where n is the outward unit normal vector to ∂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 θ_1, θ_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 φ 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 φ 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 φ requires full knowledge of the Darcy equation and a naïve evaluation of φ may be as expensive as solving the original PDE, which is itself a linear PDE; however, we believe that the ideas underlying the random features used here are intuitive and suggestive of what is possible in other applications 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 on the original problem, either because the operator to be inverted is spatially inhomogeneous or because of the complicated geometry of the physical domain.

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 $\mathcal{X}$, $\mathcal{Y}$. 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 FFT. 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:

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

where $\{\xi_j\} \sim N(0,1)$ i.i.d. 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 $C\phi_{k'} = \lambda_{k'}\phi_{k'}$ for nonnegative, symmetric, trace-class operator 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 $d = 2$, $k' = (k_1', k_2') \in \mathbb{Z}^2_{\geq 0}\backslash\{0\}$, $x = (x_1, x_2) \in (0,1)^2$, and 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 integrate to zero over D by manually setting to zero the Fourier coefficient corresponding to multi-index $k' = 0$. We use such 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 (φ, μ) 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, the draws $\{\theta_j\}_{j=1}^m$ are resampled i.i.d. from μ. To measure the distance between the trained RFM $F_m(\cdot;\hat{\alpha})$ and the ground truth $F^\dagger$, we employ the *approximate expected relative test error*

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

where the $\{a'_j\}_{j=1}^{n'}$ are drawn i.i.d. from ν 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 $\nu = N(0, C)$ using (4.1), where 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 [35] and a fourth order Runge–Kutta integrating factor time-stepping scheme for time discretization [47]. 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 ε leads to smaller length scale solutions and more difficult reconstruction; more data (higher n) and features (higher m) or a more expressive choice of (φ, μ) 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 were hand-tuned but not optimized. We find that regularization during training had 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; [92] 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 Ψ_T with $T \coloneqq 0.5$ to obtain $\hat{F} \coloneqq F_m(\cdot;\hat{\alpha})$. Then, it should follow from (4.5) that $\hat{F}^j \approx \Psi_{jT}$, that is, each application of $\hat{F}$ should evolve the solution T time units. We test this semigroup approximation by learning the map $\hat{F}$ and then comparing $\hat{F}^j$ on $n' = 4000$ fixed inputs to outputs from each of the operators Ψ_{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

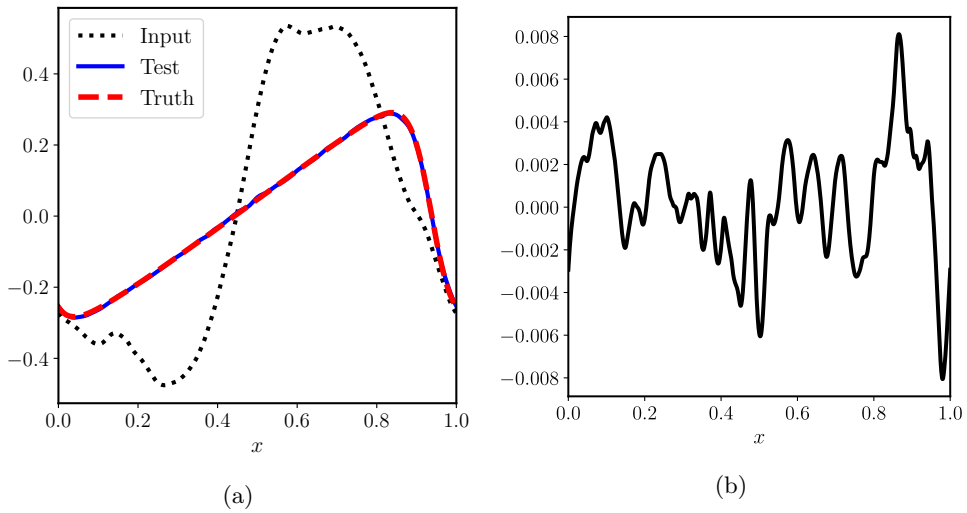


FIG. 4. *Representative input-output test sample for the Burgers' equation solution map* $F^\dagger := \Psi_1$*: Figure* 4(a) *shows a sample input, output (truth), and trained RFM prediction (test), while Figure* 4(b) *displays the pointwise error. The relative* L^2 *error for this single prediction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

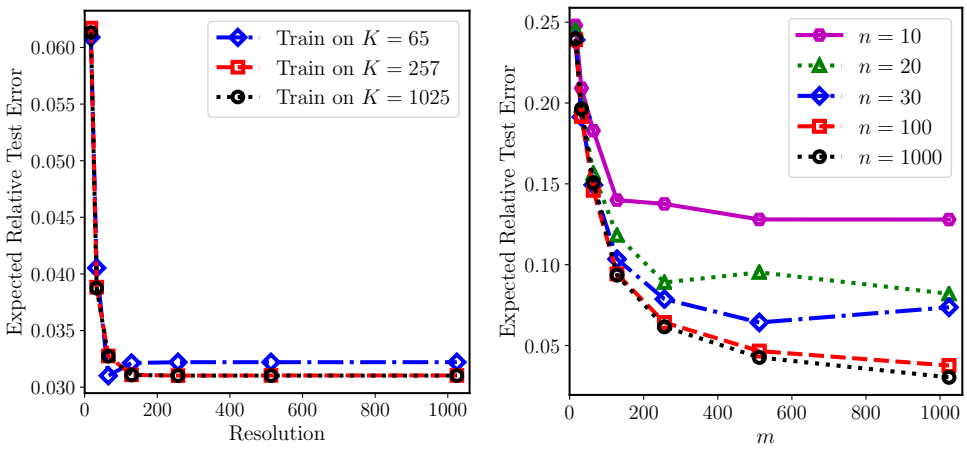
TABLE 1

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* Ψ_{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 $\hat{F}^j$ accurately captures Ψ_{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 [28]. 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 $\hat{\alpha}$ obtained from training on mesh size K to different mesh resolutions K' in Figure 5(a). We fix $T := 1$ from here on 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 [56]. 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. But outside this regime, the errors are essentially constant across resolution



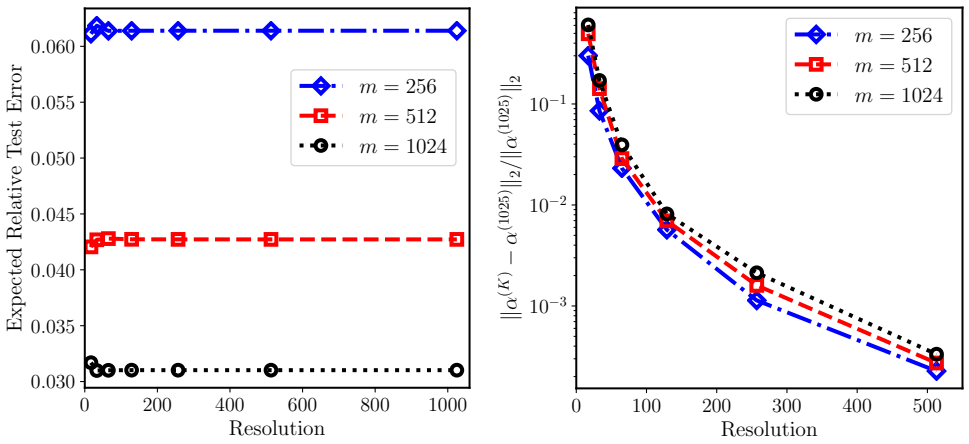
(a) (b)

FIG. 5. *Expected relative test error of a trained RFM for the Burgers' evolution operator* $F^\dagger = \Psi_1$ *with* $n' = 4000$ *test pairs: Figure* 5(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. Figure* 5(b) *shows the decay of the test error for resolution* $K = 129$ *fixed as a function of* m *and* n*; the smallest error achieved is* 0.0303 *for* $n = 1000$ *and* $m = 1024$.

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 smaller mesh sizes K (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 [55, 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 also note that for a small number of training data n, the error does not always decrease as the number of random features m increases. This indicates a delicate dependence of m as a function of n, in particular, n must increase with m as is expected from parametric estimation; we observe the desired monotonic decrease in error with m when n is increased to 100 or 1000. In the overparametrized regime, the authors in [61] present a loose bound for this dependence for real-valued outputs. We leave a detailed account of the dependence of m on n required to achieve a certain error tolerance to future work and refer the interested reader to [17] for detailed statistical analysis in a related setting.

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



(a) (b)

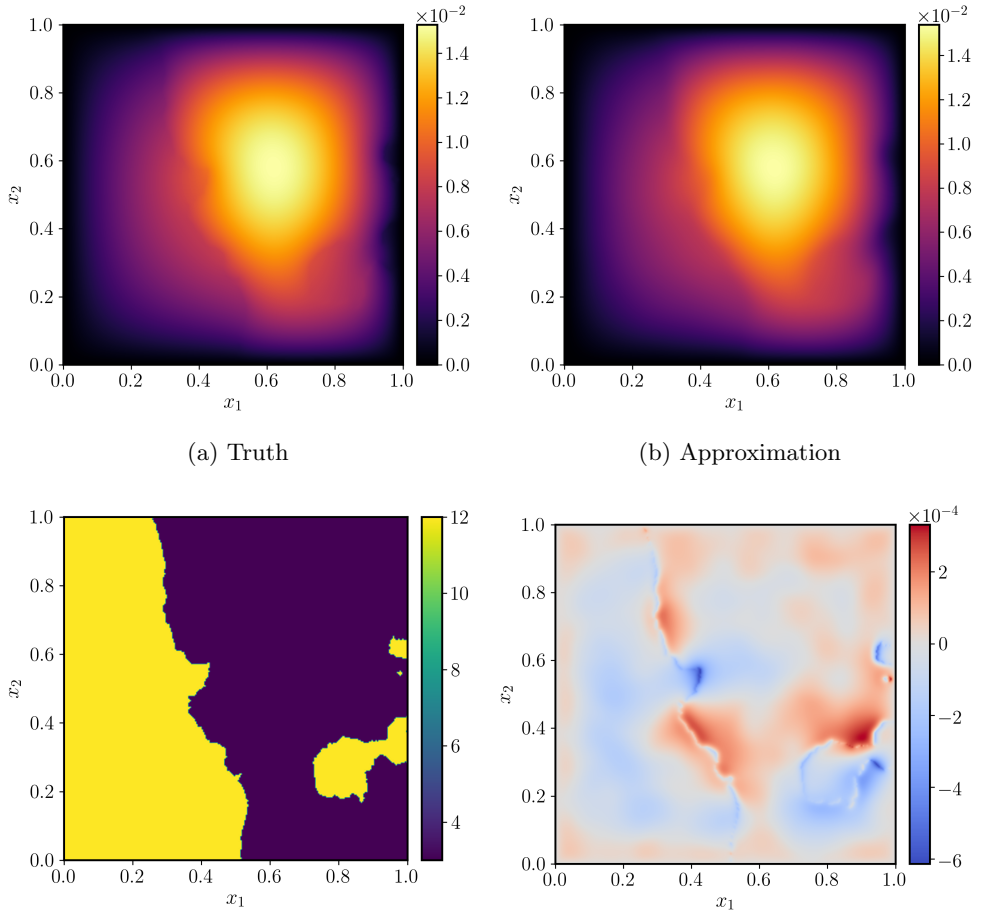
FIG. 6. *Results of a trained RFM for the Burgers' equation evolution operator* $F^\dagger = \Psi_1$*: Figure* 6(a) *shows resolution-invariant test error for various* m*; the error follows the* $O(m^{-1/2})$ *Monte Carlo rate remarkably well. Figure* 6(b) *displays the relative error of the learned coefficient* α *w.r.t. the coefficient learned on the highest mesh size* ($K = 1025$)*. Here,* $n = 512$ *training and* $n' = 4000$ *testing pairs were used.*

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 increased (see [13, 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 $K \to \infty$. More interestingly, these constant values of error, $e_{n',m} = 0.063$, 0.043, and 0.031 corresponding to $m = 256$, 512, and 1024, respectively, closely match the Monte Carlo rate $O(m^{-1/2})$. While more theory is required to understand this behavior, it suggests that the optimization process finds coefficients close to those arising from a Monte Carlo approximation of $F^\dagger$ as discussed in subsection 2.3. 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 257×257 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 ν (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 φ (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_ε in the definition of φ 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 φ. Unlike in subsection 4.1, we find via grid search on λ 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 the capacity 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)). However, the smallest test error achieved



(a) Truth (b) Approximation

(c) Input (d) Pointwise Error

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

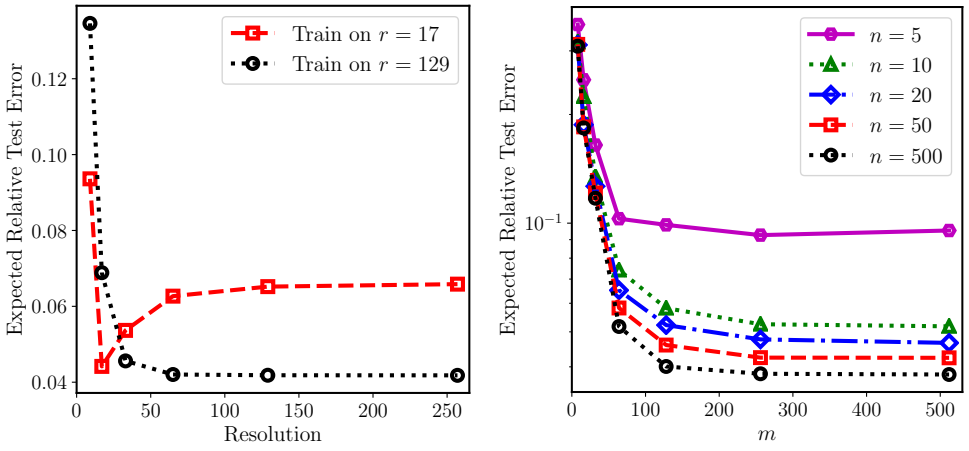
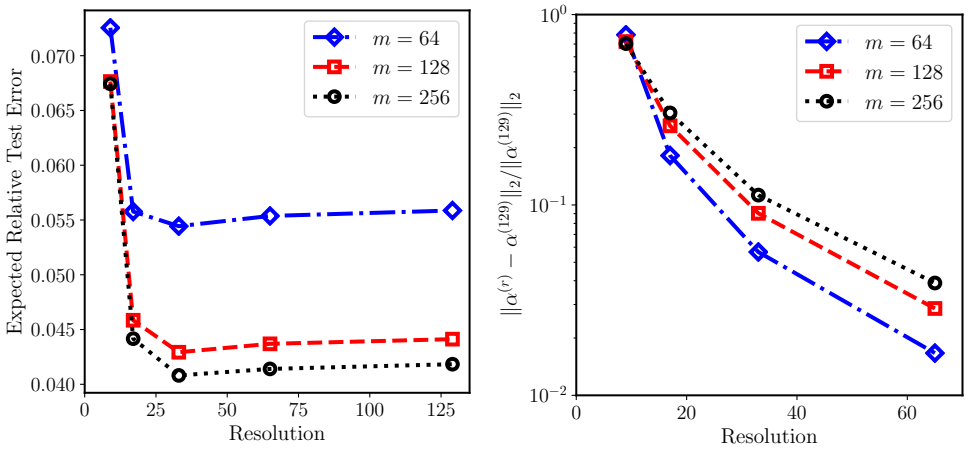


FIG. 8. *Expected relative test error of a trained RFM for Darcy flow with* $n' = 1000$ *test pairs: Figure* 8(a) *displays the invariance of test error w.r.t. training and testing on different resolutions for* $m = 512$ *and* $n = 256$ *fixed; the RFM can train and test on different mesh sizes without significant loss of accuracy. Figure* 8(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* $m = 512$.

for the best performing RFM configuration is 0.0381, which is on the same scale as the error reported in competing NN-based methods [13, 56] 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 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-NN method in [13]. 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 [13, 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 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 $r = 129$. Although we do not observe the limiting error following the Monte Carlo rate in m, which suggests that the RKHS $\mathcal{H}_{k_\mu}$ induced by the choice of φ may not be expressive enough (e.g., not universal [82]), the numerical results make clear that our method nonetheless performs well as an operator approximator.



(a) (b)

FIG. 9. *Results of a trained RFM for Darcy flow: Figure* 9(a) *demonstrates resolution-invariant test error for various* m*, while Figure* 9(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. Conclusions. In this article, we introduced a random feature methodology for the data-driven approximation of maps between infinite-dimensional Banach spaces. The RFM, as an emulator of such maps, performs dimension reduction in the sense that the original infinite-dimensional learning problem reduces to an approximate problem of finding m real numbers (section 2). Our conceptually infinite-dimensional algorithm is nonintrusive and results in a scalable method that is consistent with the continuum limit, robust to discretization, and highly flexible in practical use. These benefits were verified in numerical experiments for two nonlinear forward operators based on PDEs, one involving a semigroup and another a coefficient-to-solution operator (section 4). While the random feature–based operator emulator learned from data is not guaranteed to be cheaper to evaluate than a full order solver in general, our design of problem-specific random feature maps in section 3 leads to efficient $O(mK \log K)$ evaluation of an m-term RFM for simple physical domain geometries and hence competitive computational cost in many-query settings. A straightforward GPU implementation would provide further acceleration.

There are various directions for future work. We are interested in application of random feature methods to more challenging problems in the sciences, such as climate modeling and material modeling, and to the solution of design and inverse problems arising in those settings with the RFM serving as a cheap emulator. Of great importance in furthering the methodology is the question of how to adapt the random features to data instead of manually constructing them. Some possibilities along this line of work include the Bayesian optimization of RFM hyperparameters, as is frequently used in Gaussian process regression, or more general hierarchical learning of the pair (φ, μ) itself, both of which would lead to data-adapted induced kernels. Such developments would make the RFM more streamlined and competitive with deep learning alternatives and would serve to further clarify the effectiveness of function space learning algorithms. Finally, the development of a theory which underpins our method, allows for proof of convergence, and characterizes the quality of the RKHS spaces induced by random feature maps would be both mathematically interesting and highly desirable as it would help guide methodological development.

Appendix A. Proofs of results.

Proof of Result 2.5. Fix $a \in \mathcal{X}$ and $y \in \mathcal{Y}$. Then, we note that

$$k_\mu(\cdot, a)y = \int \langle \varphi(a;\theta), y\rangle_{\mathcal{Y}} \varphi(\cdot;\theta)\mu(d\theta) = \mathcal{A}\langle \varphi(a;\cdot), y\rangle_{\mathcal{Y}} \in \mathrm{im}(\mathcal{A})\,, \tag{A.1}$$

since $\langle \varphi(a;\cdot), y\rangle_{\mathcal{Y}} \in L^2_\mu(\Theta;\mathbb{R})$ by the Cauchy–Schwarz inequality.

Now we show that $\mathrm{im}(\mathcal{A})$ admits a reproducing property of the form (2.10). First, note that $\mathcal{A}$ can be viewed as a bijection between its coimage and image spaces, and we denote this bijection by

$$\tilde{\mathcal{A}}\colon \ker(\mathcal{A})^\perp \to \mathrm{im}(\mathcal{A})\,. \tag{A.2}$$

For any $F, G \in \mathrm{im}(\mathcal{A})$, define the candidate RKHS inner product $\langle \cdot, \cdot \rangle$ by

$$\langle F, G\rangle \coloneqq \big\langle \tilde{\mathcal{A}}^{-1}F, \tilde{\mathcal{A}}^{-1}G\big\rangle_{L^2_\mu(\Theta;\mathbb{R})}\,. \tag{A.3}$$

This is indeed a valid inner product since $\tilde{\mathcal{A}}$ is invertible. Note that for any $q \in \ker(\mathcal{A})$,

$$\begin{aligned}\big\langle q, \langle \varphi(a;\cdot), y\rangle_{\mathcal{Y}}\big\rangle_{L^2_\mu(\Theta;\mathbb{R})} &= \int q(\theta)\langle \varphi(a;\theta), y\rangle_{\mathcal{Y}}\,\mu(d\theta)\\ &= \left\langle \int q(\theta)\varphi(a;\theta)\mu(d\theta), y\right\rangle_{\mathcal{Y}}\\ &= 0\end{aligned}$$

so that $\langle \varphi(a;\cdot), y\rangle_{\mathcal{Y}} \in \ker(\mathcal{A})^\perp$. Then for any $F \in \mathrm{im}(\mathcal{A})$, we compute

$$\begin{aligned}\langle k_\mu(\cdot, a)y, F\rangle &= \big\langle \langle \varphi(a;\cdot), y\rangle_{\mathcal{Y}}, \tilde{\mathcal{A}}^{-1}F\big\rangle_{L^2_\mu(\Theta;\mathbb{R})}\\ &= \int \langle \varphi(a;\theta), y\rangle_{\mathcal{Y}}(\tilde{\mathcal{A}}^{-1}F)(\theta)\mu(d\theta)\\ &= \left\langle \int (\tilde{\mathcal{A}}^{-1}F)(\theta)\varphi(a;\theta)\mu(d\theta), y\right\rangle_{\mathcal{Y}}\\ &= \big\langle y, (\mathcal{A}\tilde{\mathcal{A}}^{-1}F)(a)\big\rangle_{\mathcal{Y}}\\ &= \langle y, F(a)\rangle_{\mathcal{Y}}\,,\end{aligned}$$

which gives exactly (2.10) if our candidate inner product is defined to be the RKHS inner product. Since $F \in \mathrm{im}(\mathcal{A})$ is arbitrary, this and (A.1) together imply that $\mathrm{im}(\mathcal{A}) = \mathcal{H}_{k_\mu}$ is the RKHS induced by k_μ as shown in [26, 46]. ∎

Proof of Result 2.6. Since $L^2_{\mu^{(m)}}(\Theta;\mathbb{R})$ is isomorphic to $\mathbb{R}^m$, we can consider the map $\mathcal{A}\colon \mathbb{R}^m \to L^2_\nu(\mathcal{X};\mathcal{Y})$ defined in (2.14), and use Result 2.5 to conclude that

$$\mathcal{H}_{k^{(m)}} = \mathrm{im}(\mathcal{A}) = \left\{ \frac{1}{m}\sum_{j=1}^m c_j\varphi(\cdot;\theta_j)\colon c \in \mathbb{R}^m \right\} = \mathrm{span}\{\varphi_j\}_{j=1}^m\,, \tag{A.4}$$

since the $\{\varphi_j\}_{j=1}^m$ are assumed linearly independent. ∎

Proof of Result 2.8. Recall from Result 2.6 that the RKHS $\mathcal{H}_{k^{(m)}}$ comprises the linear span of the $\{\varphi_j \coloneqq \varphi(\cdot;\theta_j)\}_{j=1}^m$. Hence $\varphi_j \in \mathcal{H}_{k^{(m)}}$, and note that by the reproducing kernel property (2.10), for any $F \in \mathcal{H}_{k^{(m)}}$, $a \in \mathcal{X}$, and $y \in \mathcal{Y}$,

$$\begin{aligned}\langle y, F(a)\rangle_{\mathcal{Y}} &= \big\langle k^{(m)}(\cdot,a)y, F\big\rangle_{\mathcal{H}_{k^{(m)}}} \\ &= \frac{1}{m}\sum_{j=1}^m \langle \varphi_j(a), y\rangle_{\mathcal{Y}} \langle \varphi_j, F\rangle_{\mathcal{H}_{k^{(m)}}} \\ &= \left\langle y, \frac{1}{m}\sum_{j=1}^m \langle \varphi_j, F\rangle_{\mathcal{H}_{k^{(m)}}} \varphi_j(a)\right\rangle_{\mathcal{Y}}.\end{aligned}$$

Since this is true for all $y \in \mathcal{Y}$, we deduce that

$$F = \frac{1}{m}\sum_{j=1}^m \alpha_j \varphi_j\,, \quad \alpha_j = \langle \varphi_j, F\rangle_{\mathcal{H}_{k^{(m)}}}\,. \tag{A.5}$$

As the $\{\varphi_j\}_{j=1}^m$ are assumed linearly independent, we deduce that the representation (A.5) is unique.

Finally, we calculate the RKHS norm of any such F in terms of α:

$$\begin{aligned}\|F\|^2_{\mathcal{H}_{k^{(m)}}} = \langle F, F\rangle_{\mathcal{H}_{k^{(m)}}} &= \left\langle \frac{1}{m}\sum_{j=1}^m \alpha_j\varphi_j, F\right\rangle_{\mathcal{H}_{k^{(m)}}} \\ &= \frac{1}{m}\sum_{j=1}^m \alpha_j \langle \varphi_j, F\rangle_{\mathcal{H}_{k^{(m)}}} \\ &= \frac{1}{m}\sum_{j=1}^m \alpha_j^2\,.\end{aligned}$$

Substituting this into (2.24), we obtain the desired equivalence with (2.23). ∎

Appendix B. Further remarks on integral representation of RKHS. We recall the linear operator $\mathcal{A}$ (2.14) from subsection 2.3. In this appendix, we clarify the meaning of (2.12) and show that $\mathcal{A}$ is a square root of T_{k_μ}. A similar discussion is provided by Bach in [3, sect. 2] for the special case $\mathcal{Y} = \mathbb{R}$.

By the assumption $\varphi \in L^2_{\nu\times\mu}(\mathcal{X}\times\Theta;\mathcal{Y})$ and the Cauchy–Schwarz inequality,

$$\mathcal{A} \in \mathcal{L}\big(L^2_\mu(\Theta;\mathbb{R}), L^2_\nu(\mathcal{X};\mathcal{Y})\big)\,. \tag{B.1}$$

Now let $F \in \operatorname{im}(\mathcal{A}) = \mathcal{H}_{k_\mu}$. We have $F = \mathcal{A}c$ for some $c \in L^2_\mu$. But since $\ker(\mathcal{A})$ is closed, $L^2_\mu = \ker(\mathcal{A}) \oplus \ker(\mathcal{A})^\perp$ and hence there exist unique $q_F \in \ker(\mathcal{A})$ and $c_F \in \ker(\mathcal{A})^\perp$ such that $c = q_F + c_F$. Using the notation in (A.2), we have $c_F = \tilde{\mathcal{A}}^{-1}F$ by definition of $\tilde{\mathcal{A}}$. The reproducing property in the proof of Result 2.5 produced the representation $F = \mathcal{A}c_F$; in fact, the similar calculation leading to (2.12) in subsection 2.3 also identified the unique c_F, there defined formally by $c_F(\theta) = \langle \varphi(\cdot;\theta), F\rangle_{\mathcal{H}_{k_\mu}}$. Indeed,

$$\begin{aligned}\langle c_F, q\rangle_{L^2_\mu(\Theta;\mathbb{R})} &= \int \langle \varphi(\cdot;\theta), F\rangle_{\mathcal{H}_{k_\mu}} q(\theta)\mu(d\theta) \\ &= \left\langle \int q(\theta)\varphi(\cdot;\theta)\mu(d\theta), F\right\rangle_{\mathcal{H}_{k_\mu}} \\ &= 0\end{aligned}$$

for any $q \in \ker(\mathcal{A})$. Hence $c_F \in \ker(\mathcal{A})^\perp$, and we interpret (2.12) as formal notation for the unique element $\tilde{\mathcal{A}}^{-1}F \in \ker(\mathcal{A})^\perp$. Using formula (A.3) and orthogonality, we also obtain the following useful characterization of the RKHS norm:

$$\|F\|^2_{\mathcal{H}_{k_\mu}} = \big\|\tilde{\mathcal{A}}^{-1}F\big\|^2_{L^2_\mu} = \|c_F\|^2_{L^2_\mu} = \min_{c\in\mathcal{C}_F} \|c\|^2_{L^2_\mu}\,, \tag{B.2}$$

where $\mathcal{C}_F \coloneqq \{c \in L^2_\mu(\Theta;\mathbb{R})\colon \mathcal{A}c = F\}$.

Finally, we show that $\mathcal{A}\mathcal{A}^* = T_{k_\mu}$. This means that the RKHS is equal to the image of two different square roots of integral operator $T_{k_\mu}\colon \mathcal{H}_{k_\mu} = \mathrm{im}(T^{1/2}_{k_\mu}) = \mathrm{im}(\mathcal{A})$. First, for any $F \in L^2_\nu(\mathcal{X};\mathcal{Y})$ and $c \in L^2_\mu(\Theta;\mathbb{R})$,

$$\begin{aligned}
\langle F, \mathcal{A}c\rangle_{L^2_\nu} &= \left\langle F, \int c(\theta)\varphi(\cdot;\theta)\mu(d\theta)\right\rangle_{L^2_\nu} \\
&= \int c(\theta)\langle F, \varphi(\cdot;\theta)\rangle_{L^2_\nu}\,\mu(d\theta) \\
&= \left\langle \int \langle F(a'), \varphi(a';\cdot)\rangle_{\mathcal{Y}}\,\nu(da'), c\right\rangle_{L^2_\mu}
\end{aligned}$$

by the Fubini–Tonelli theorem. So, we deduce that the adjoint of $\mathcal{A}$ is

$$\begin{aligned}
&\mathcal{A}^*\colon\ L^2_\nu(\mathcal{X};\mathcal{Y}) \to L^2_\mu(\Theta;\mathbb{R})\,, \\
&\qquad\qquad F \mapsto \mathcal{A}^*F \coloneqq \int \langle F(a'), \varphi(a';\cdot)\rangle_{\mathcal{Y}}\,\nu(da')\,,
\end{aligned} \tag{B.3}$$

which is bounded since $\mathcal{A}$ is bounded. For any $F \in L^2_\nu(\mathcal{X};\mathcal{Y})$, we compute

$$\begin{aligned}
\mathcal{A}\mathcal{A}^*F &= \int_\Theta (\mathcal{A}^*F)(\theta)\varphi(\cdot;\theta)\mu(d\theta) \\
&= \int_\Theta \int_{\mathcal{X}} \langle F(a'), \varphi(a';\theta)\rangle_{\mathcal{Y}}\,\varphi(\cdot;\theta)\nu(da')\mu(d\theta) \\
&= \int_{\mathcal{X}} \left(\int_\Theta \varphi(\cdot;\theta)\otimes\varphi(a';\theta)\mu(d\theta)\right) F(a')\nu(da') \\
&= T_{k_\mu}F\,,
\end{aligned}$$

again by Fubini–Tonelli, as desired.

Acknowledgments. The authors thank Bamdad Hosseini and Nikola B. Kovachki for helpful discussions and are grateful to the two anonymous referees for their careful reading and insightful comments.

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*, preprint, arXiv:2012.06081, 2020.

[2] N. Aronszajn, *Theory of reproducing kernels*, Trans. Amer. Math. Soc., 68 (1950), pp. 337–404.

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

[4] 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. 15344–15349.
[5] 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., 339 (2004), pp. 667–672.
[6] A. R. Barron, *Universal approximation bounds for superpositions of a sigmoidal function*, IEEE Trans. Inform. Theory, 39 (1993), pp. 930–945.
[7] J. Bear and M. Y. Corapcioglu, *Fundamentals of Transport Phenomena in Porous Media*, NATO. Sci. Ser. 82, Springer, New York, 2012.
[8] 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.
[9] P. Benner, A. Cohen, M. Ohlberger, and K. Willcox, *Model Reduction and Approximation: Theory and Algorithms*, Comput. Sci. Eng. 15, SIAM, Philadelphia, 2017.
[10] A. Berlinet and C. Thomas-Agnan, *Reproducing Kernel Hilbert Spaces in Probability and Statistics*, Springer, New York, 2011.
[11] C. Bernardi and R. Verfürth, *Adaptive finite element methods for elliptic equations with non-smooth coefficients*, Numer. Math., 85 (2000), pp. 579–608.
[12] G. Beylkin and M. J. Mohlenkamp, *Algorithms for numerical analysis in high dimensions*, SIAM J. Sci. Comput., 26 (2005), pp. 2133–2159.
[13] K. Bhattacharya, B. Hosseini, N. B. Kovachki, and A. M. Stuart, *Model Reduction and Neural Networks for Parametric PDEs*, preprint, arXiv:2005.03180, 2020.
[14] D. Bigoni, Y. Chen, N. G. Trillos, Y. Marzouk, and D. Sanz-Alonso, *Data-Driven Forward Discretizations for Bayesian Inversion*, preprint, arXiv:2003.07991, 2020.
[15] R. Brault, M. Heinonen, and F. Buc, *Random fourier features for operator-valued kernels*, in Proceedings of the Asian Conference on Machine Learning, 2016, pp. 110–125.
[16] 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.
[17] A. Caponnetto and E. De Vito, *Optimal rates for the regularized least-squares algorithm*, Found. Comput. Math., 7 (2007), pp. 331–368.
[18] 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.
[19] 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, 2020, 1143.
[20] 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.
[21] 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. Quantifi., 1 (2013), pp. 452–493.
[22] A. Chkifa, A. Cohen, R. DeVore, and C. Schwab, *Sparse adaptive taylor approximation algorithms for parametric and stochastic elliptic PDEs*, ESAIM Mathe. Model. Numer. Anal., 47 (2013), pp. 253–280.
[23] A. Cohen and R. DeVore, *Approximation of high-dimensional parametric PDEs*, Acta Numer., 24 (2015), pp. 1–159.
[24] A. Cohen and G. Migliorati, *Optimal Weighted Least-Squares Methods*, preprint, arXiv:1608.00512, 2016.
[25] 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.
[26] F. Cucker and S. Smale, *On the mathematical foundations of learning*, Bull. Amer. Math. Soc., 39 (2002), pp. 1–49.
[27] M. Dashti and A. M. Stuart, *The Bayesian approach to inverse problems*, in Handbook of Uncertainity Quantification, Springer, New York, 2017, pp. 311–428, https://doi.org/10.1007/978-3-319-12385-1_7.
[28] L. Demanet, *Curvelets, Wave Atoms, and Wave Equations*, Ph.D. thesis, California Institute of Technology, 2006.
[29] R. A. DeVore, *The theoretical foundation of reduced basis methods*, in Model Reduction and Approximation: Theory and Algorithms, SIAM, Philadelphia, 2014, pp. 137–168.
[30] 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.
[31] M. M. Dunlop, M. A. Iglesias, and A. M. Stuart, *Hierarchical bayesian level set inversion*, Statist. Comput., 27 (2017), pp. 1555–1584.

[32] L. C. Evans, *Partial Differential Equations*, Grad. Ser. Math. 19, AMS, Providence, RI, 2010.
[33] Y. Fan and L. Ying, *Solving electrical impedance tomography with deep learning*, J. Comput. Phys., 404 (2020), pp. 109–119.
[34] J. Feliu-Faba, Y. Fan, and L. Ying, *Meta-learning pseudo-differential operators with deep neural networks*, J. Comput. Phys., 408 (2020), 109309.
[35] B. Fornberg, *A Practical Guide to Pseudospectral Methods*, Vol. 1, Cambridge University Press, Cambridge, UK, 1998.
[36] H. Gao, J.-X. Wang, and M. J. Zahr, *Non-Intrusive Model Reduction of Large-Scale, Nonlinear Dynamical Systems Using Deep Learning*, preprint, arXiv:1911.03808, 2019.
[37] M. Geist, P. Petersen, M. Raslan, R. Schneider, and G. Kutyniok, *Numerical Solution of the Parametric Diffusion Equation by Deep Neural Networks*, preprint, arXiv:2004.12131, 2020.
[38] D. Gilbarg and N. S. Trudinger, *Elliptic Partial Differential Equations of Second Order*, Springer, New York, 2015.
[39] R. Gonzalez-Garcia, R. Rico-Martinez, and I. Kevrekidis, *Identification of distributed parameter systems: A neural net based approach*, Computers Chemical Engineering, 22 (1998), pp. S965–S968.
[40] M. Griebel and C. Rieger, *Reproducing kernel Hilbert spaces for parametric partial differential equations*, SIAM/ASA J. Uncertain. Quantifi., 5 (2017), pp. 111–137.
[41] E. Haber and L. Ruthotto, *Stable architectures for deep neural networks*, Inverse Problems, 34 (2017), 014004.
[42] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, New York, 2009.
[43] J. S. Hesthaven and S. Ubbiali, *Non-intrusive reduced order modeling of nonlinear problems using neural networks*, J. Computat. Phys., 363 (2018), pp. 55–78.
[44] M. Hinze, R. Pinnau, M. Ulbrich, and S. Ulbrich, *Optimization with PDE Constraints*, Math. Model. Theory Appl. 23, Springer, New York, 2008.
[45] 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.
[46] 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.
[47] A.-K. Kassam and L. N. Trefethen, *Fourth-order time-stepping for stiff PDEs*, SIAM J. Sci. Comput., 26 (2005), pp. 1214–1233.
[48] 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.
[49] Y. Khoo, J. Lu, and L. Ying, *Solving Parametric PDF Problems with Artificial Neural Networks*, preprint, arXiv:1707.03351, 2017.
[50] A. Kiselev, F. Nazarov, and R. Shterenberg, *Blow Up and Regularity for Fractal Burgers Equation*, preprint, arXiv:0804.3549, 2008.
[51] Y. Korolev, *Two-Layer Neural Networks with Values in a Banach Space*, preprint, arXiv:2105.02095, 2021.
[52] G. Kutyniok, P. Petersen, M. Raslan, and R. Schneider, *A Theoretical Analysis of Deep Neural Networks and Parametric PDEs*, preprint, arXiv:1904.00377, 2019.
[53] K. Lee and K. T. Carlberg, *Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders*, J. Comput. Phys., 404 (2020), p. 108973.
[54] Y. Li, J. Lu, and A. Mao, *Variational training of neural network approximations of solution maps for physical models*, J. Comput. Phys., 409 (2020), 109338.
[55] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, *Fourier Neural Operator for Parametric Partial Differential Equations*, preprint, arXiv:2010.08895, 2020.
[56] Z. Li, N. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, and A. Anandkumar, *Neural Operator: Graph Kernel Network for Partial Differential Equations*, preprint, arXiv:2003.03485, 2020.
[57] Z. Long, Y. Lu, X. Ma, and B. Dong, *PDE-Net: Learning PDEs from Data*, preprint, arXiv:1710.09668, 2017.
[58] L. Lu, P. Jin, and G. E. Karniadakis, *DeepONet: Learning Nonlinear Operators for Identifying Differential Equations Based on the Universal Approximation Theorem of Operators*, preprint, arXiv:1910.03193, 2019.
[59] D. G. Luenberger, *Optimization by Vector Space Methods*, John Wiley & Sons, New York, 1997.

[60] C. Ma, L. Wu, and E. Weinan, *Machine Learning from a Continuous Viewpoint*, preprint, arXiv:1912.12777, 2019.
[61] C. Ma, L. Wu, and E. Weinan, *On the Generalization Properties of Minimum-Norm Solutions for Over-parameterized Neural Network Models*, preprint, arXiv:1912.06987, 2019.
[62] B. Matérn, *Spatial Variation*, Lecture Notes in Statist. 36, Springer, Cham, 2013.
[63] C. A. Micchelli and M. Pontil, *On learning vector-valued functions*, Neural Comput., 17 (2005), pp. 177–204.
[64] R. M. Neal, *Priors for infinite networks*, in Bayesian Learning for Neural Networks, Springer, New York, 1996, pp. 29–53.
[65] T. O'Leary-Roseberry, U. Villa, P. Chen, and O. Ghattas, *Derivative-Informed Projected Neural Networks for High-Dimensional Parametric Maps Governed by PDEs*, preprint, arXiv:2011.15110, 2020.
[66] 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.
[67] R. G. Patel and O. Desjardins, *Nonlinear Integro-Differential Operator Regression with Neural Networks*, preprint, arXiv:1810.08552, 2018.
[68] 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*, preprint, arXiv:2009.11992, 2020.
[69] B. Peherstorfer, K. Willcox, and M. Gunzburger, *Survey of multifidelity methods in uncertainty propagation, inference, and optimization*, SIAM Rev., 60 (2018), pp. 550–591.
[70] A. Rahimi and B. Recht, *Random features for large-scale kernel machines*, in Advances in Neural Information Processing Systems, 2008, pp. 1177–1184.
[71] 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.
[72] 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.
[73] 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.
[74] R. Rico-Martinez, K. Krischer, I. Kevrekidis, M. Kube, and J. Hudson, *Discrete- vs. continuous-time nonlinear signal processing of Cu electrodissolution data*, Chemical Engineering Communications, 118 (1992), pp. 25–48.
[75] 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.
[76] L. Ruthotto and E. Haber, *Deep neural networks motivated by partial differential equations*, J. Math. Imaging Vision, (2019), pp. 1–13.
[77] N. D. Santo, S. Deparis, and L. Pegolotti, *Data Driven Approximation of Parametrized PDEs by Reduced Basis and Neural Networks*, preprint, arXiv:1904.01514, 2019.
[78] 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.
[79] J. Sirignano and K. Spiliopoulos, *DGM: A deep learning algorithm for solving partial differential equations*, J. Comput. Phys., 375 (2018), pp. 1339–1364.
[80] P. D. Spanos and R. Ghanem, *Stochastic finite element expansion for random media*, J. Engineering Mechanics, 115 (1989), pp. 1035–1053.
[81] B. Stevens and T. Colonius, *FiniteNet: A Fully Convolutional LSTM Network Architecture for Time-Dependent Partial Differential Equations*, preprint, arXiv:2002.03014, 2020.
[82] Y. Sun, A. Gilbert, and A. Tewari, *On the Approximation Properties of Random ReLU Features*, preprint, arXiv:1810.04374, 2019.
[83] 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.
[84] 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.
[85] E. Weinan, *A proposal on machine learning via dynamical systems*, Commun. Math. Stat., 5 (2017), pp. 1–11.
[86] E. Weinan, J. Han, and Q. Li, *A mean-field optimal control formulation of deep learning*, Res. Math. Sci., 6 (2019), 10.

[87] E. Weinan 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.
[88] H. Wendland, *Scattered Data Approximation*, Cambridge, Monogr. Appl. Comput. Math. 17, Cambridge University Press, Cambridge, UK, 2004.
[89] C. K. Williams, *Computing with infinite networks*, in Advances in Neural Information Processing Systems, 1997, pp. 295–301.
[90] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*, Vol. 2, MIT Press, Cambridge, MA, 2006.
[91] 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.
[92] K. Wu and D. Xiu, *Data-driven deep learning of partial differential equations in modal space*, J. Comput. Phys., 408 (2020), 109307.
[93] 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.