# Operator Learning: A Statistical Perspective

**Unique Subedi, Ambuj Tewari**

Department of Statistics, University of Michigan, Ann Arbor, Michigan
Email: subedi@umich.edu, tewaria@umich.edu

Annu. Rev. Stat. Appl. 2025. 13:1–28

https://doi.org/10.1146/((please add article doi))



**Keywords**

Operator Learning, Partial Differential Equations, Scientific Machine Learning

**Abstract**

Operator learning has emerged as a powerful tool in scientific computing for approximating mappings between infinite-dimensional function spaces. A primary application of operator learning is the development of surrogate models for the solution operators of partial differential equations (PDEs). These methods can also be used to develop black-box simulators to model system behavior from experimental data, even without a known mathematical model. In this article, we begin by formalizing operator learning as a function-to-function regression problem and review some recent developments in the field. We also discuss PDE-specific operator learning, outlining strategies for incorporating physical and mathematical constraints into architecture design and training processes. Finally, we end by highlighting key future directions such as active data collection and the development of rigorous uncertainty quantification frameworks.

## 1. INTRODUCTION

Artificial Intelligence (AI) is making rapid advances in a variety of areas ranging from language and vision modeling to applications in the physical sciences. AI is transforming scientific research and accelerating scientific discovery by providing new tools for modeling complex systems (Tang et al. 2020) and optimizing workflows (Wang et al. 2023). A notable example is the protein structure prediction with AlphaFold (Jumper et al. 2021), which was awarded the 2024 Nobel Prize in Chemistry (The Nobel Committee 2024). This growing impact has led to the emergence of "AI for Science" (Zhang et al. 2023), a paradigm that seeks to integrate AI into scientific problem-solving. The subject of this article, operator learning, is one of the key approaches within this new paradigm.

In mathematics, a mapping between infinite dimensional function spaces is often called an *operator*. Operator learning is an area at the intersection of applied mathematics, computer science, and statistics which studies how we can *learn* such operators from data. Its primary application is the development of fast and accurate surrogate models (Bhattacharya et al. 2021) for the solution operators of partial differential equations (PDEs). Additionally, as a data-driven approach, operator learning techniques can be used to develop black-box simulators that simulate system behavior based on observed experimental data (You et al. 2022a,b), even when the underlying mathematical model is unknown.

Before formally defining the problem of operator learning, let us discuss a motivating example that illustrates its relevance. Many physical systems are governed by PDEs, which describe how the system evolves given particular initial conditions. A classic example is the heat equation (Evans 2022, Section 2.3)

$$\frac{\partial u}{\partial t} = \tau \nabla^2 u, \tag{1}.$$

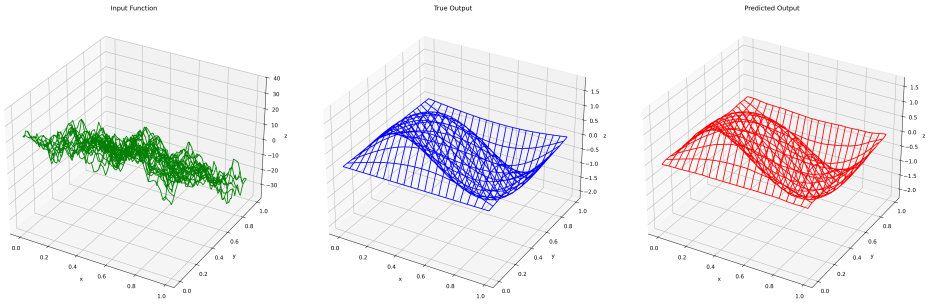
that arises in heat conduction and diffusion problems. Here, $\tau > 0$ could be the thermal conductivity of a material, $u : \mathcal{X} \times [0, \infty) \to \mathbb{R}$ for some set $\mathcal{X} \subseteq \mathbb{R}^d$ could define a temperature profile at any given space-time coordinate, and $\nabla^2$ is the Laplacian operator defined as $\nabla^2 u := \sum_{j=1}^{d} \partial^2 u / \partial x_j^2$. The solution of the heat equation can be written using a linear operator defined as

$$\exp(\tau t \nabla^2) := \sum_{k=0}^{\infty} \frac{(\tau t \, \nabla^2)^k}{k!}.$$

That is, given an initial condition $u_0$, the solution function can be written as $u_t = \exp(\tau t \nabla^2) u_0$ for any time point $t > 0$ (Hunter 2023, Chapter 5.4).

This solution operator is useful primarily for conceptual understanding and cannot be used to obtain the solution function in all but a few cases of simple domain geometry and simple initial conditions. The solution is generally obtained using PDE solvers which use numerical methods to map the initial conditions $u_0$ to $u_t$ at some desired time point $t > 0$. Such solver starts from scratch for every new initial condition $u_0$ of interest. Since the solver is computationally slow and expensive, this ab initio approach to evaluating solutions can be limiting in applications such as engineering design where the solution needs to be evaluated for many different initial conditions (Umetani & Bickel 2018). To solve this problem, operator learning aims to learn the solution operator directly from the data. By amortizing the computational cost through upfront training, these learned operators allow for significantly efficient solution evaluation compared to traditional solvers while sacrificing a small degree of accuracy.

More precisely, for some prespecified time point $t = T$, let $\mathrm{G} := \exp(\tau T \nabla^2)$ denote the solution operator of interest. Then, given training data $(v_1, w_1), \ldots, (v_n, w_n)$ where $v_i$ is the initial condition and $w_i = \mathrm{G}(v_i)$ is the solution at time point $T$, operator learning involves estimating an approximation $\widehat{\mathrm{F}}_n$ of G by searching over a a predefined operator class $\mathcal{F}$ (Kovachki et al. 2024b, Section 2). Once trained, the estimated operator can be used to predict an approximate solution $\widehat{w} = \widehat{\mathrm{F}}_n(v)$ for a new initial condition $v$. The objective is to design an estimation procedure such that $\widehat{w}$ closely approximates the true solution $w = \mathrm{G}(v)$ under a suitable metric. For illustration, Figure 1 compares the Fourier neural operator's prediction and the actual output from a PDE solver.




(a) Input ($v$) (b) Output ($w$) (c) Prediction ($\widehat{w}$)

Figure 1: Input function, ground truth solution of the heat equation, and the predicted solution by a Fourier Neural Operator. Here, we use $\mathcal{X} = [0, 1)^2, \tau = 0.05$, and $T = 1$. The corresponding code is available in this notebook.

In the example of the heat equation, we wanted to learn an operator that maps an initial condition to the solution function. But operator learning methods can also be used to learn how the parameters of steady-state equations map to their corresponding solution functions. For example, for a prototypical elliptic PDE

$$-\mathrm{div}(v(\cdot)\,\nabla w) = f, \qquad v(x) = 0 \quad \forall x \in \mathrm{boundary}(\mathcal{X}),$$

one may seek to learn the nonlinear ground truth operator G that maps the coefficient function $v$ to the solution function $w$. The function $v : \mathcal{X} \to [0, \infty]$ typically represents permeability of the medium in fluid flow applications (Cheng 1984). A surrogate operator trained on for this mapping can efficiently predict how the solution function $w$ changes in response to variations in the system's properties defined by $v$.

Additionally, these operator surrogates can also be used to complement traditional PDE solvers rather than completely replace them. For example, the prediction of operator surrogate can be used as an initialization for spectral solvers of nonlinear PDEs, which often rely on iterative methods such as Newton's method (Aghili et al. 2025). A well-chosen initialization can substantially accelerate convergence, reducing the overall computational cost.

Although operator learning is a relatively new field, it has already had a practical impact. For example, the European Centre for Medium-Range Weather Forecasts has incorporated an operator learning model, FourcastNet, into its experimental forecasting suite (ECMWF 2025). In the medical field, operator learning has been applied to optimize

catheter prototypes–devices used in surgical procedures and disease treatment (Zhou et al. 2024). Similarly, operator learning has been applied to model plasma evolution in nuclear fusion research. Figure 2 illustrates the ability of an operator surrogate to predict the long-term plasma dynamics within a tokamak reactor using an operator surrogate (Gopakumar et al. 2024). In addition, operator learning has also shown promise in biological modeling, where the underlying dynamics are often unknown. For example, You et al. (2022a) used it to model the mechanical response of biological tissues from digital image correlation measurements, without making assumptions about tissue structure or the underlying mapping. Finally, operator learning has been used to accelerate sampling from generative models. In diffusion models, sampling generally requires hundreds of iterations of the discretized reverse diffusion process. However, Zheng et al. (2023) demonstrated that learning the solution operator of the reverse process enables one-shot sampling without iteration.

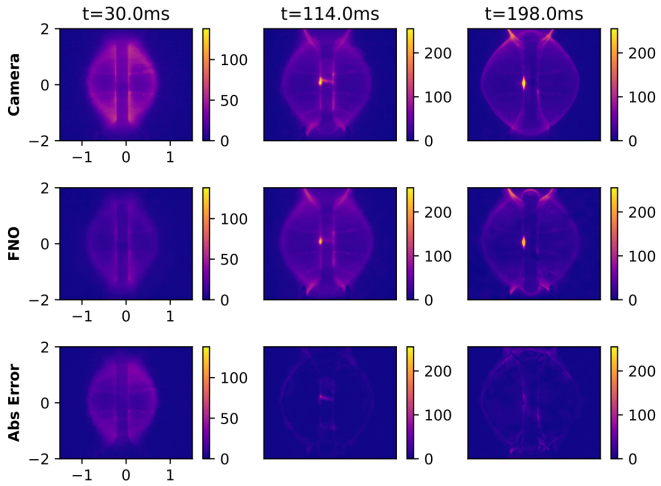



Figure 2: Long-term prediction of plasma evolution around the central solenoid in the tokamak. The top row shows experimental camera images, the middle row presents predictions from a Fourier Neural Operator surrogate, and the bottom row depicts the absolute error of the prediction. Figure recreated from (Gopakumar et al. 2024).

Given these early successes, operator learning not only received coverage in public media (Ananthaswamy 2021, Anandkumar 2023) but also attracted the attention of several research communities. For example, Kovachki et al. (2024b) explore the foundations operator learning from an approximation theory perspective, Boullé & Townsend (2024) discuss its connections to numerical linear algebra, and Azizzadenesheli et al. (2024) provide an application-focused overview. Our article contributes to this growing body of work by adopting a statistical perspective on operator learning.

We begin by formalizing operator learning as a regression problem between function spaces, a problem that has been studied by the statistics community as "functional regression" or "function-on-function regression" within the functional data analysis (FDA) literature. Recent advances in data-driven surrogate modeling for PDEs have revitalized the interest in this problem. We adopt the term "operator learning" instead of "functional regression" as it better reflects our viewpoint of using statistical learning framework with

an appropriate loss function and a class of operators. Next, we review recent progress in operator learning, and finally, we conclude by discussing key open questions and future research directions.

**Banach Space:** A vector space $\mathcal{V}$ over $\mathbb{R}$ (or $\mathbb{C}$) equipped with a norm $\|\cdot\|$ such that $(\mathcal{V}, \|\cdot\|)$ is complete, i.e., every Cauchy sequence in $\mathcal{V}$ converges to an element of $\mathcal{V}$.

## 2. A STATISTICAL FRAMEWORK FOR OPERATOR LEARNING

We approach operator learning from the point of view of statistical learning theory (Vapnik 2000) with the important difference that both input and output spaces are spaces consisting of *functions*. To this end, fix $\mathcal{X} \subseteq \mathbb{R}^d$, and let $\mathcal{V}, \mathcal{W}$ to be separable Banach spaces of $\mathbb{R}^p$-valued functions on $\mathcal{X}$. As an example, one may have $\mathcal{V} = \mathcal{W} = L^2_\nu(\mathcal{X}, \mathbb{R}^p)$, the set of square-integrable $\mathbb{R}^p$-valued functions defined on the domain $\mathcal{X}$. The base measure $\nu$ is typically the Lebesgue measure; however, for certain cases, such as when $\mathcal{X} = \mathbb{R}^d$, a weighted measure like Gaussian (with density proportional to $e^{-\alpha^2\|x\|^2}\,dx$) may be considered to ensure the base measure is finite. In general, $\mathcal{V}$ can be defined as functions mapping from $\mathbb{R}^{d_1}$ to $\mathbb{R}^{p_1}$, and $\mathcal{W}$ as functions mapping from $\mathbb{R}^{d_2}$ to $\mathbb{R}^{p_2}$. However, the learning-theoretic and practical aspects of the problem remain largely unchanged under such generality. Thus, we take $d_1 = d_2$ and $p_1 = p_2$ to minimize notational complexity and simplify exposition.

We now formally define the statistical problem of operator learning. Let $\mathrm{G} : \mathcal{V} \to \mathcal{W}$ be the ground truth operator of interest. For example, this could be the solution operator of the PDE of interest. In the statistical learning framework, the learner is provided with $n$ i.i.d. samples $\{(v_i, \mathrm{G}(v_i))\}_{i=1}^n$, where $v_i$'s are drawn iid from a distribution $\mu$. Throughout the work, we will assume that $\mu$ belongs to a family of distributions $\mathcal{P}$ on $\mathcal{V}$, and the learner has knowledge of the family.

Using this sample and a predefined learning rule, the learner constructs an estimator $\widehat{\mathrm{F}}_n$. For simplicity, we use $\widehat{\mathrm{F}}_n$ to denote both the estimator and the estimation rule. Although $\widehat{\mathrm{F}}_n$ can theoretically be any operator from $\mathcal{V}$ to $\mathcal{W}$, in practice, the learner typically searches within a prespecified operator class $\mathcal{F} \subseteq \mathcal{W}^{\mathcal{V}}$. For example, $\mathcal{F}$ might be the class of bounded linear operators or neural network-based operator classes (see Section 3). However, the true operator of interest G need not be in the class $\mathcal{F}$. Accordingly, the estimator $\widehat{\mathrm{F}}_n$ is evaluated relative to the best operator within the class $\mathcal{F}$. Thus, for a prespecified loss function $\ell : \mathcal{W} \times \mathcal{W} \to \mathbb{R}_{\geq 0}$, the worst-case (over $\mu \in \mathcal{P}$) expected excess risk of $\widehat{\mathrm{F}}_n$ is defined as

$$\mathcal{E}_n(\widehat{\mathrm{F}}_n, \mathcal{F}, \mathcal{P}, \mathrm{G}) := \sup_{\mu \in \mathcal{P}} \left( \mathop{\mathbb{E}}_{v_1,\ldots,v_n \sim_{\text{iid}} \mu} \left[ \mathop{\mathbb{E}}_{v\sim\mu} \left[ \ell\big(\widehat{\mathrm{F}}_n(v), \mathrm{G}(v)\big) \right] \right] - \inf_{\mathrm{F}\in\mathcal{F}} \mathop{\mathbb{E}}_{v\sim\mu} \left[ \ell\big(\mathrm{F}(v), \mathrm{G}(v)\big) \right] \right). \tag{2}$$

Then, the learner's objective is to develop an estimation rule such that $\mathcal{E}_n(\widehat{\mathrm{F}}_n, \mathcal{F}, \mathcal{P}, \mathrm{G}) \to 0$ as $n \to \infty$.

To make the problem non-trivial from an estimation perspective, we assume that the learner does not know the exact form of G. Otherwise, the learner could simply set $\widehat{\mathrm{F}}_n = \mathrm{G}$ or its closest approximation in $\mathcal{F}$, making the estimation problem trivial. However, the learner may still have some structural knowledge about G, such as its linearity or smoothness properties, based on prior information of the underlying PDE. Alternatively, the learner may know that the class $\mathcal{F}$ contains an operator that approximates G to within a small error, meaning that

$$\inf_{\mathrm{F}\in\mathcal{F}} \mathop{\mathbb{E}}_{v\sim\mu} \left[ \ell\big(\mathrm{F}(v), \mathrm{G}(v)\big) \right] \leq \varepsilon$$

for some small $\varepsilon > 0$. This assumption is reasonable for function classes with a universal

approximation property. In statistical learning, this is often referred to as the well-specified or $\varepsilon$-realizable setting. The well-known additive noise model is a special case of this setting. To see this, let $\mathrm{G}(v) = \mathrm{F}^\star(v) + \delta$ for some $\mathrm{F}^\star \in \mathcal{F}$ and a random noise term $\delta \in \mathcal{W}$ such that $\mathbb{E}[\|\delta\|_{\mathcal{W}}^2] \leq \varepsilon$. Here, instead of G being the ground truth, it is a $\delta$ perturbation of the output of ground truth $\mathrm{F}^\star$. If the loss function $\ell$ is the squared norm in the Banach space $\mathcal{W}$, then $\inf_{\mathrm{F}\in\mathcal{F}} \mathbb{E}\left[\|\mathrm{F}(v) - \mathrm{G}(v)\|_{\mathcal{W}}^2\right] \leq \mathbb{E}\left[\|\mathrm{F}^\star(v) - \mathrm{G}(v)\|_{\mathcal{W}}^2\right] = \mathbb{E}[\|\delta\|_{\mathcal{W}}^2] \leq \varepsilon$. Here, the expectation accounts for the randomness in both $v$ and $\delta$.

## 2.1. Loss Functions

A common choice for the loss function is $\ell(\widehat{w}, w) = \|\widehat{w} - w\|_{\mathcal{W}}^q$, where $\|\cdot\|_{\mathcal{W}}$ denotes the canonical norm associated with the Banach space $\mathcal{W}$. In practice, $q$ is typically set to 1 or 2. Another frequently used loss function is the relative loss, given by

$$\ell(\widehat{w}, w) = \frac{\|\widehat{w} - w\|_{\mathcal{W}}^q}{\|w\|_{\mathcal{W}}^q},$$

which is often preferred in empirical settings, as it has been observed to yield better results (Kovachki et al. 2023, Section 6.5). Additionally, if prior knowledge about the smoothness of the functions is available, $\mathcal{W}$ may be chosen as an appropriate Sobolev space to incorporate this information into the learning process, generally referred to as Sobolev training (Son et al. 2021).

**Gaussian Process:** A stochastic process $\{v(x) : x \in \mathcal{X}\}$ such that for any finite set $\{x_1, \ldots, x_m\}$, the random vector $(v(x_1), \ldots, v(x_m))$ is a multivariate Gaussian in $\mathbb{R}^m$.

## 2.2. Distribution Families

In principle, one could define $\mathcal{P}$ as the set of all Borel probability distributions on $\mathcal{V}$. This family with a worst-case analysis over all operators $\mathrm{G} : \mathcal{V} \to \mathcal{W}$ leads to the standard minimax framework. However, minimax analysis fails to capture the prior knowledge that a practitioner generally has. Thus, it is common to impose additional constraints on $\mathcal{P}$.

In the applied operator learning literature, $\mathcal{P}$ is often chosen as a class of Gaussian process with specific covariance structures. Works such as Bhattacharya et al. (2021), Li et al. (2021), Kovachki et al. (2023) assume that input functions are drawn from a Gaussian process $\mathrm{GP}(0, \alpha(-\nabla^2 + \beta\mathbf{I})^{-\gamma})$ for some $\alpha, \beta, \gamma > 0$. This assumption is justified in these works as the datasets are synthetically generated by directly sampling functions from this distribution. Similarly, Lu et al. (2021a) generate input functions using a Gaussian process with an RBF kernel, while Boullé & Townsend (2024) suggest more general covariance structures, such as the Matérn kernel, which provides explicit control over the smoothness of sampled functions. Further discussion on sampling from such distributions is provided in Section 4.1.

For theoretical analysis, Lanthaler et al. (2022) consider $\mathcal{P}$ as the class of all distributions whose covariance operators have finite trace norm. This corresponds to mean-square continuous processes with covariances that are integral operators of Mercer kernels, as considered by Subedi & Tewari (2025b). Meanwhile, Liu et al. (2024) adopts a non-parametric perspective, defining $\mathcal{P}$ as the set of all compactly supported measures on $\mathcal{V}$.

# 3. OPERATOR CLASSES

In this section, we present an overview of select classes frequently used in the operator learning literature. Our primary focus is to describe how mappings between infinite-dimensional

spaces are defined and to examine the learning-theoretic aspects of these classes. Thus, this discussion is not meant to serve as an exhaustive review of all classes in operator learning. For a more comprehensive overview, we refer the readers to articles by Boullé & Townsend (2024) and Kovachki et al. (2024b).

**Hilbert Space:** A complete inner product space $(\mathcal{H}, \langle\cdot,\cdot\rangle)$ with norm $\|x\| = \sqrt{\langle x, x\rangle}$. Every Hilbert space is a Banach space with the inner product structure.

## 3.1. Linear Operators

For scalar-valued linear regression, the class of linear mappings consists of functions of the form $x \mapsto \langle \beta, x\rangle$. A straightforward analog of this model can be defined for operator learning. Let $\mathcal{V}$ and $\mathcal{W}$ be Hilbert spaces with orthonormal bases $\{\varphi_j\}_{j\in\mathbb{N}}$ and $\{\psi_j\}_{j\in\mathbb{N}}$, respectively. For a sequence $\beta = \{\beta_j\}_{j\in\mathbb{N}}$, consider the mapping

$$v \mapsto \sum_{j=1}^{\infty} \beta_j \langle v, \varphi_j\rangle \, \psi_j. \tag{3}.$$

One can define a class of such mappings as $\mathcal{F} = \{v \mapsto \sum_{j=1}^{\infty} \beta_j \langle v, \varphi_j\rangle \, \psi_j \;:\; \|\beta\|_{\ell^2(\mathbb{N})} \le c\}$.

Since the orthonormal bases are fixed apriori, learning this linear map reduces to estimating the sequence $\{\beta_j\}_{j\in\mathbb{N}}$. This is effectively a regression task in the frequency domain, a well-studied problem in statistical modeling (Harvey 1978). Moreover, this estimation framework is closely related to principal component functional regression, which has been studied in works of Hörmann & Kidziński (2015) and Reimherr (2015). Recently, de Hoop et al. (2023) and Subedi & Tewari (2025a) analyzed this model in the context of learning solution operators of PDEs.

**Matrix-Valued Kernel:** A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{p\times p}$ with entries $k_{ij}$, which are each a scalar-valued kernel $k_{ij} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. A scalar-valued kernel is a function that is (i) symmetric $k_{ij}(x, x') = k_{ij}(x', x)$ and (ii) positive semidefinite, $\sum_{a,b=1}^{m} c_a c_b \, k_{ij}(x_a, x_b) \ge 0 \; \forall x_1, \ldots, x_m \in \mathcal{X}$ and $c_1, \ldots, c_m \in \mathbb{R}$.

Despite its simplicity, this model can be used to approximate the solution operator of the heat equation (1). Let $\{\varphi_j\}_{j=1}^{\infty}$ be the eigenfunctions of the Laplacian operator $-\nabla^2$ on the domain $\mathcal{X}$, with corresponding eigenvalues $\{\eta_j\}_{j\ge1}$. That is, $-\nabla^2\varphi_j = \eta_j \, \varphi_j$ for all $j \in \mathbb{N}$. Then, the solution operator of the heat equation has a spectral representation (Dodziuk 1981) such that any initial condition $u_0$ is mapped to the solution function at time point $t$ as

$$u_0 \mapsto \sum_{j=1}^{\infty} e^{-\tau\eta_j t} \, \langle u_0, \varphi_j\rangle \, \varphi_j,$$

which can be parametrized as Equation (3).

A slightly more general linear model can be defined using an integral operator associated with a kernel. Let $k_\theta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{p\times p}$ be a kernel parameterized by $\theta$. Define the operator $\mathrm{K}_\theta$ as follows

$$(\mathrm{K}_\theta \, v)(y) = \int_{\mathcal{X}} k_\theta(y, x) \, v(x) \, d\nu(x), \tag{4}.$$

where $\nu$ is a measure on $\mathcal{X}$. This model has been extensively studied in the functional data analysis literature, especially when $p = 1$ and $v \in L^2([0, 1], \mathbb{R})$ (Wang et al. 2016, Section 3.2).

Kernel integral operators are particularly useful for learning linear boundary value problems of the form $\mathcal{L}w = v$, where $w(x) = 0$ for all $x \in \text{boundary}(\mathcal{X})$. Under regularity conditions such as uniform ellipticity, the solution operator for these problems can be represented as an integral operator of the associated Green's kernel (Boullé et al. 2023). For example, when $\mathcal{X} = \mathbb{R}^d$, the solution operator of the heat equation can also be written as the integral operator of Gaussian kernel (Evans 2022, Section 2.3), that is

$$u_t(y) = \int_{\mathbb{R}^d} \frac{1}{(4\pi\tau t)^{d/2}} \exp\left(-\frac{\|y - x\|^2}{4\tau t}\right) \, u_0(x) \, dx.$$

However, for general domains $\mathcal{X}$, the heat kernel does not always have a closed-form solution. In fact, deriving an explicit Green's function is not possible for most linear PDEs, particularly in high dimensions.

To address this, Boullé et al. (2022) proposed learning the Green's function directly from data by parameterizing it with a neural network. This approach offers two main benefits. First, the problem reduces from estimating an operator on infinite-dimensional spaces to estimating a function defined on a finite-dimensional domain. Second, Green's functions capture fundamental properties such as conservation laws and symmetries, providing better interpretability compared to the general solution operator. Additionally, when the underlying boundary value problem is non-linear, Gin et al. (2021) proposed to use an encoder-decoder framework, where the Green function is learned in an encoding space where the PDE is linearized. More recently, Stepaniants (2023) developed a theoretical framework for learning the Green's function within a reproducing kernel Hilbert space (RKHS). Using the theory of kernel methods, Stepaniants (2023) derived rigorous statistical guarantees for the resulting estimator.

## 3.2. Neural Operators

Unlike the heat equation, many PDEs of practical interest have nonlinear solution operators. Neural operators are a class of neural network-based architectures developed to approximate such nonlinear operators. Recall that a multilayer neural network is defined by sequential composition of multiple single layer networks. In standard finite dimensional settings, a single layer network is a map $x \mapsto \sigma(Wx + b)$ for some pointwise non-linearity $\sigma(\cdot)$. Neural operator is a natural generalization of such mapping to function spaces. In particular, for a given input function $v$, a single layer of a neural operator is defined as

$$v \mapsto \sigma\Big((\mathrm{K}_\theta\, v)(\cdot) + b(\cdot)\Big), \qquad \text{where} \qquad (\mathrm{K}_\theta\, v)(y) = \int_{\mathcal{X}} k_\theta(y, x)\, v(x)\, d\nu(x).$$

Here, $\mathrm{K}_\theta$ is a kernel integral operator of a kernel $k_\theta$ parameterized by $\theta$, $b : \mathcal{X} \to \mathbb{R}^p$ is a bias function, and $\sigma : \mathbb{R}^p \to \mathbb{R}^p$ is a pointwise nonlinear activation function, such as ReLU. We refer interested readers to Kovachki et al. (2023) for a detailed discussion.

Although the neural operator model in its current form was introduced by Li et al. (2020a,b) and further developed in Kovachki et al. (2023), the underlying mapping $v \mapsto \sigma\Big((\mathrm{K}_\theta\, v)(\cdot) + b(\cdot)\Big)$ has been studied in the FDA literature as functional single and multi-index models (see (Wang et al. 2016, Equation 13) and (Chen et al. 2011)). However, the operator learning literature has developed novel techniques that enable fast training and efficient evaluation of such nonlinear mappings.

Since data is typically provided on a grid over a domain of size $N$, a naive implementation of this model based on numerical integration would require a time complexity of $O(N^2)$. Since the grid size $N$ is typically exponentially large in $d$, this naive approach becomes a significant bottleneck while training large neural operator models with multiple layers. Thus, recent advances in neural operator methods have focused on overcoming this computational limitation by introducing techniques that reduce the complexity and enable efficient training and evaluation of these models at a large scale. One such technique involves parametrizing the kernel $k_{\theta_t}$ in the Fourier domain, which gives rise to a well-known architecture called the Fourier Neural Operator.

**3.2.1. Fourier Neural Operators (FNO).** The original formulation of Fourier Neural Operator (FNO) (Li et al. 2021) assumes a translation-invariant kernel and applies the convolution theorem. Here, we present an equivalent formulation using a Mercer-type decomposition of $k_\theta$. Suppose $k_\theta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{p\times p}$ admits the expansion

$$[k_\theta(y,x)]_{ij} = \sum_{m\in\mathbb{Z}^d} \lambda_{ij}(m) e^{2\pi i m\cdot(y-x)},$$

where $\lambda_{ij} : \mathbb{Z}^d \to \mathbb{C}$. Define the matrix-valued function $\Lambda : \mathbb{Z}^d \to \mathbb{C}^{p\times p}$ such that $[\Lambda(m)]_{ij} = \lambda_{ij}(m)$. Using this decomposition and changing the order of sum and integral, the kernel operator $\mathrm{K}_\theta$ can be written as

$$(\mathrm{K}_\theta\, v)(y) = \sum_{m\in\mathbb{Z}^d} e^{2\pi i m\cdot y} \left(\Lambda(m) \int_{\mathcal{X}} e^{-2\pi i m\cdot x}\, v(x)\, d\nu(x)\right).$$

This formulation describes FNOs as first computing the Fourier transform of $v$, applying a transformation $\Lambda(m)$ to each mode, and reconstructing the output using an inverse Fourier transform, often written succinctly as $\mathrm{IFT}\left(\Lambda \cdot \mathrm{FT}(v)\right)$.

There are two key challenges in a practical implementation of this transformation. First, the infinite summation over $\mathbb{Z}^d$, and second, the approximation of $\mathrm{FT}(v)$ due to discretization. Li et al. (2021) address the first challenge by truncating the sum to $|m|_{\ell^\infty} \le M$, reducing parameters of the model to at most $M^d$ matrices. The second challenge is addressed by approximating $\mathrm{FT}(v)$ using the discrete Fourier transform (DFT) of $v$, computed over the finite grid of domain points. Suppose the data is available on a uniform grid of size $N$. Then, DFT can be efficiently computed using fast Fourier transform (FFT) algorithms, which have a computational complexity of $O(N \log N)$.

Note that this parameterization is just a special case of Equation (3), where the basis is explicitly chosen as the Fourier basis. More generally, kernel expansion can use alternative bases, such as Chebyshev polynomials or wavelets (Tripura & Chakraborty 2023, Gupta et al. 2021), which can be better for non-periodic or non-smooth functions. Variants of FFT also exist for these transformations, though they may require different grid structures such as Chebyshev nodes (Trefethen 2019, Chapter 2) or dyadic grids for wavelets (Mallat 1999, Chapter 7).

**3.2.2. DeepOnet.** While neural operators are typically described as architectures composed of mappings of the form $v \mapsto \sigma(\mathrm{K}\, v + b)$, there are also other ways in which neural network modeling has inspired operator learning architectures. Perhaps the most popular one is the DeepOnet architecture proposed by Lu et al. (2021a), based on the pioneering work of Chen & Chen (1995). Given the input function $v$, the DeepOnet architecture is a mapping

$$v \mapsto \sum_{j=1}^{q} b_j\big(\mathcal{E}(v)\big)\, t_j(\cdot).$$

Here, $\mathcal{E} : \mathcal{V} \to \mathbb{R}^r$ is an encoder that encodes the input function $v$ on $\mathbb{R}^r$, $b_j : \mathbb{R}^r \to \mathbb{R}$ is typically a neural network that processes an encoding of $v$ in $\mathbb{R}^r$, and $t_j : \mathcal{X} \to \mathbb{R}^p$ is another neural network that takes the evaluation point $y$ of the output function in $\mathcal{W}$ as an input. Although the learned functions $t_j$ may not be orthogonal in practice, this architecture can be interpreted as follows: the functions $(t_j)_{j=1}^q$ learn the dominant basis of $\mathcal{W}$ that captures

**Mercer's Expansion:** If $k$ is a continuous, symmetric, positive semidefinite kernel on a compact domain $\mathcal{X}$, then there exist eigenpairs $(\lambda_\ell, \phi_\ell)$ of the integral operator $(Tf)(y) = \int_{\mathcal{X}} k(y,x) f(x)\, dx$ such that $\sum_{\ell=1}^{\infty} \lambda_\ell \phi_\ell(y)\phi_\ell(x)$ converges uniformly to $k$.

**Fourier Transform Operator:** A linear operator $\mathrm{FT} : L^2(\mathbb{R}^d) \to L^2(\mathbb{R}^d)$ defined by

$$\mathrm{FT}(v) = \int_{\mathbb{R}^d} v(x)\, e^{-2\pi i\langle x,\cdot\rangle}\, dx.$$

Its inverse $\mathrm{IFT}(\cdot)$ is a linear operator such that $\mathrm{IFT}(\mathrm{FT}(v)) = v$.

the most significant variations in the target functions, while the functions $(b_j)_{j=1}^q$ map an input function to the corresponding basis coefficients.

A common choice for the encoder is $\mathcal{E}(v) = (v(x_1), \ldots, v(x_N))$, where $x_1, \ldots, x_N$ are the grid points at which $v$ is sampled. However, this encoding is not discretization-invariant, meaning the model cannot generalize to evaluate a new input function $v'$ sampled on a different grid. To achieve discretization invariance, $v$ can instead be encoded as $\mathcal{E}(v) = (\langle v, \varphi_1 \rangle, \ldots, \langle v, \varphi_r \rangle)$, where $(\varphi_j)_{j=1}^r$ are orthonormal functions in the space $\mathcal{V}$. In a special case when $(\varphi_j)_{j=1}^r$ correspond to the first $r$ principal components of the covariance operator of $\mu$ and $(t_j)_{j=1}^q$ represent the first $q$ principal components of the covariance operator of the pushforward measure $\mathrm{G}_{\#}(\mu)$, the resulting architecture is referred to as PCA-Net (Bhattacharya et al. 2021).

## 3.3. RKHS and Random Features

Let $\mathcal{W}$ be a Hilbert space. Recall that the usual reproducing kernel Hilbert space (RKHS) of scalar-valued functions can be defined using a real-valued kernel. However, defining a RKHS of operators requires an *operator-valued* kernel. To that end, let $\mathcal{B}(\mathcal{W})$ denote the space of bounded linear operators on $\mathcal{W}$. A function $K : \mathcal{V} \times \mathcal{V} \to \mathcal{B}(\mathcal{W})$ is an operator-valued kernel if: (1) $K$ is Hermitian, i.e., $K(u, v) = K(v, u)^*$ for all $u, v \in \mathcal{V}$ and (2) $K$ is non-negative, meaning that for any $\{(v_i, w_i)\}_{i=1}^n$, we have $\sum_{i=1}^n \sum_{j=1}^n \langle w_i, K(v_i, v_j) w_j \rangle_{\mathcal{W}} \geq 0$.

The class $\mathcal{F}$ is an operator RKHS associated with $K$ if

(1) $K(v, \cdot)w \in \mathcal{F}$ for all $v \in \mathcal{V}$, $w \in \mathcal{W}$.
(2) For any $v \in \mathcal{V}$, $w \in \mathcal{W}$, and $\mathrm{F} \in \mathcal{F}$, we have $\langle \mathrm{F}, K(v, \cdot)w \rangle_{\mathcal{F}} = \langle \mathrm{F}(v), w \rangle_{\mathcal{W}}$.

The second property is the reproducing property, which implies a closed-form solution for penalized least-squares estimation under the RKHS norm. Given training data $\{(v_i, w_i)\}_{i=1}^n$, the solution to such penalized-estimation problem takes the form

$$\widehat{\mathrm{F}}_n(\cdot) = \sum_{i=1}^{n} K(v_i, \cdot)\alpha_i,$$

for some coefficients $\alpha_1, \ldots, \alpha_n \in \mathcal{W}$. Further details on operator RKHS and the corresponding estimation problem can be found in the work of Micchelli & Pontil (2005). The framework of using operator RKHS for function valued regression was developed in a series of works by Kadri et al. (2010, 2011, 2016). In an interesting work, Batlle et al. (2024) showed that these RKHS-based methods for operator learning are competitive with neural network-based methods on standard benchmark datasets. More recently, Mora et al. (2025) developed a Gaussian process framework for operator learning where they estimate a real-valued measurement of the solution operator, which allows them to use finite-dimensional kernel methods.

Despite the theoretical appeal of the estimator mentioned above, its practical implementation is challenging. The primary challenge arises from the fact that the kernel $K$ is operator-valued. Kadri et al. (2016) studied a simplified case where $K$ takes the form $K(u, v) = k(u, v)L$ for a scalar-valued kernel $k$ and a fixed linear operator $L$. Even under this restrictive assumption, computing the solution requires expensive tensor-based techniques when $L$ is non-diagonal. For a general kernel $K$, the computational cost becomes prohibitive for most practical applications.

A more practical alternative to working directly with operator-valued kernels is the

random feature model for operators proposed by Nelsen & Stuart (2021). The random feature model (RFM) for scalar-valued functions was originally introduced by Rahimi & Recht (2007) as a scalable approximation of kernel methods by constructing a randomized low-dimensional approximation of the true RKHS feature map. This idea was later extended to matrix- and operator-valued kernels by Brault et al. (2016) and Minh (2016), respectively. However, given the difficulty of defining and implementing non-trivial operator-valued kernels, Nelsen & Stuart (2021) proposed RFM as a standalone model rather than an approximation technique for kernel methods. Thus, our focus here is only on the resulting trainable random feature model. For a detailed discussion on operator RFMs and their connection to kernel methods, we refer the readers to (Nelsen & Stuart 2021, 2024).

Defining an RFM requires a feature map $\Phi : \mathcal{V} \times \Theta \to \mathcal{W}$, where $\Theta$ is a set of parameters and a probability measure $\rho$ on $\Theta$. Then, a random feature model indexed by $\vartheta$ is

$$\mathrm{F}_{\mathrm{RFM}}(v;\vartheta) := \frac{1}{M}\sum_{j=1}^{M}\beta_j \Phi(v;\theta_j)$$

for all $v \in \mathcal{V}$. Here, $\beta_1, \ldots, \beta_M$ are some scalars and $\vartheta := \{\theta_j\}_{j=1}^M$, where $\theta_j$'s are iid draw from the probability measure $\rho$. The class $\mathcal{F}$ indexed by $\vartheta$ can be defined as all such RFMs such that $\sum_{j=1}^M |\beta_j|^2 \leq c$ for some $c > 0$. This model is trained in so-called lazy regime, where the random parameters are fixed and one only trains the scalar parameters $\beta_1, \ldots, \beta_M$. Since this is a linear optimization problem, one can derive the unique minimizer and establish statistical guarantees of the resulting estimator (Lanthaler & Nelsen 2023).

Although RFM was originally conceptualized in Rahimi & Recht (2007) from its connection to kernel methods, one can go beyond feature maps of the associated kernels. For example, one can consider a random feature model where $\Phi(\cdot, \theta_j)$'s are random initialization of large powerful models such Fourier Neural Operators or DeepOnets. Such a model can potentially give us the expressivity of neural networks while providing rigorous statistical guarantees.

## 4. DATA GENERATION, ESTIMATION, AND EVALUATION

Beyond the model itself, the success of operator learning depends on the choice of data distribution, estimation strategy, and evaluation framework. We next discuss these key practical aspects of operator learning.

### 4.1. Data Generation and Sampling

One of the defining features of operator learning, particularly for applications involving partial differential equations (PDEs), is the flexibility in data acquisition. Unlike traditional machine learning tasks, where data is often collected from fixed experiments or observations, operator learning benefits from the ability to generate labeled data by directly querying a PDE solver. The process begins with sampling input functions $v$ from a carefully chosen distribution $\mu$.

In the applied literature, $v$ is typically sampled from a mean-zero Gaussian process with a covariance operator of the form $\alpha(-\nabla^2 + \beta\mathbf{I})^{-\gamma}$. This distribution, widely used in the applied stochastic PDEs literature (Lord et al. 2014), was first proposed for operator learning by Bhattacharya et al. (2021) and also implemented in later works (Li et al. 2021, Kovachki et al. 2023). As Figure 3 shows, the parameter $\gamma$ controls the average smoothness of the

**Karhunen–Loève Decomposition:** An orthogonal expansion of a centered stochastic process $v$ with covariance operator $\Sigma$, defined as $v(x) = \sum_{\ell=1}^{\infty} \sqrt{\lambda_\ell}\, \xi_\ell\, \phi_\ell(x)$, where $(\lambda_\ell, \phi_\ell)$ are eigenpairs of $\Sigma$ and $\{\xi_\ell\}$ are zero-mean, uncorrelated random variables. It is an infinite-dimensional analog of principal component expansion.

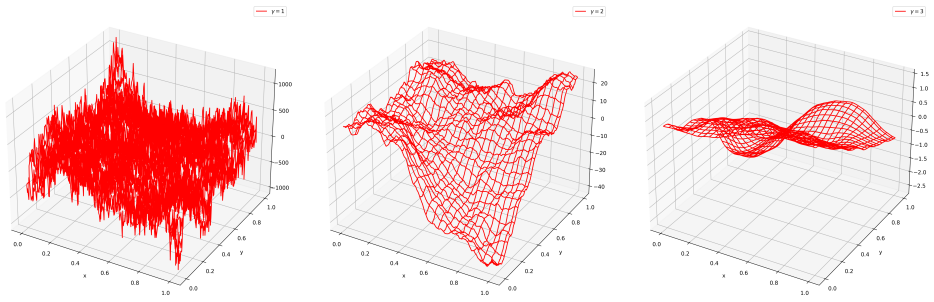
generated samples. This allows practitioners to adjust $\gamma$ to incorporate prior knowledge about the smoothness of input functions relevant to the specific application.

To sample input functions from such a distribution, one makes use of Karhunen-Loève decomposition. Let $\{\varphi_j\}_{j=1}^{\infty}$ denote the eigenfunctions of $-\nabla^2$ on $\mathcal{X}$, with corresponding eigenvalues being $(\eta_j)_{j\geq 1}$. By the Spectral Mapping Theorem, $\{\varphi_j\}_{j=1}^{\infty}$ remain the eigenfunctions of $(-\nabla^2 + \beta\mathbf{I})^{-\gamma}$, with eigenvalues being $\lambda_j := \alpha(\eta_j + \beta)^{-\gamma}$. If $\sum_{j=1}^{\infty} \lambda_j < \infty$, then the Karhunen-Loève Theorem (Hsing & Eubank 2015, Theorem 7.3.5) states that any sample $v \sim \mathrm{GP}(0, \alpha(-\nabla^2 + \beta\mathbf{I})^{-\gamma})$ can be decomposed as

$$v(x) = \sum_{j=1}^{\infty} \sqrt{\lambda_j}\, \xi_j\, \varphi_j(x) \qquad \forall x \in \mathcal{X},$$

where $\{\xi_j\}_{j=1}^{\infty}$ are uncorrelated standard Gaussian random variables on $\mathbb{R}$. Thus, sampling $v$ is reduced to sampling a sequence of independent Gaussian random variables $(\xi_j)_{j=1}^{\infty}$, which is often truncated to $(\xi_j)_{j=1}^{M}$ in practice. The resulting truncation yields a sample $v(x) = \sum_{j=1}^{M} \sqrt{\lambda_j}\, \xi_j\, \varphi_j(x)$, which is then evaluated on a predefined discrete grid of $\mathcal{X}$.

Sampling from this distribution requires knowledge of eigenpairs $(\lambda_j, \varphi_j)_{j\geq 1}$, which depend on the domain geometry. In certain cases, these can be computed analytically. For example, on the $1d$ torus ($\mathbb{T} \simeq [0, 1]$), the eigenfunctions are the Fourier basis, with eigenvalues $\lambda_j = \alpha(\beta + 4\pi^2|j|^2)^{-\gamma}$. This also extends to higher-dimensional torus $\mathbb{T}^d$ (Subedi & Tewari 2025a, Section 3.2.1). Similarly, on a $d$-dimensional sphere $S^d$, the eigenfunctions are spherical harmonics, though closed-form expressions may not exist for arbitrary domains.



(a) $\gamma = 1$ (b) $\gamma = 2$ (c) $\gamma = 3$

Figure 3: Samples from GP $\left(0, (-\nabla^2 + \mathbf{I})^{-\gamma}\right)$ for different values of $\gamma$ on the domain $[0, 1)^2$, illustrating how $\gamma$ determines the smoothness of the generated functions.

Moreover, other kernels such as the Matérn or radial basis function (RBF) can also be used. For example, Lu et al. (2021a) generate samples using the RBF kernel, $K(y, x) = \exp(-\|x - y\|^2/(2\ell^2))$. The eigenfunctions of the RBF kernel are Hermite polynomials with corresponding eigenvalues that decay exponentially fast (Williams & Rasmussen 2006, Section 4.3.1). When the eigenpairs of the kernels are not available in closed form, various numerical methods can be used to approximate them (Williams & Rasmussen 2006, Section 4.3.2). Additionally, it is possible to go beyond Gaussian processes and sample $(\xi_j)_{j\geq 1}$ from other distributions with heavier tails such as from $t$-distribution or Uniform($[-c, c]$) for some $c > 0$.

Once the input function $v$ has been generated, numerical solvers are used to obtain the corresponding solution $w = \mathrm{G}(v)$. There are many numerical methods such as finite difference, spectral, and finite element. To keep this discussion focused on the statistical aspect of operator learning, we will not expand further on these numerical methods. We refer the interested readers to Boullé & Townsend (2024, Section 4.1.2) and references therein.

## 4.2. Estimation

Given $n$ i.i.d. samples $\{(v_i, \mathrm{G}(v_i))\}_{i=1}^n$, the estimator is typically obtained by empirical risk minimization,

$$\widehat{\mathrm{F}}_n \in \arg\min_{\mathrm{F}\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \ell(\mathrm{F}(v_i), \mathrm{G}(v_i)).$$

As discussed in Section 2.1, common choice of the loss function includes squared $L^2$ loss, $\ell(\mathrm{F}(v_i), \mathrm{G}(v_i)) = \|\mathrm{F}(v_i) - \mathrm{G}(v_i)\|_{L^2}^2$, or a related relative loss. Since the data is only available on a discrete grid $\{x_1, \ldots, x_N\} \subset \mathcal{X}$, the $L^2$ norm is approximated empirically as $\|\mathrm{F}(v_i) - \mathrm{G}(v_i)\|_{L^2}^2 \approx \frac{1}{N}\sum_{j=1}^{N} \big|\mathrm{F}(v_i)(x_j) - \mathrm{G}(v_i)(x_j)\big|^2$. The optimization problem is then solved using first-order methods such as stochastic gradient descent (SGD) and its variants. When the underlying PDE is known, additional constraints imposed by the PDE such as conservation laws are often incorporated as a regularization term in the optimization problem. This technique, referred to as physics-informed training, is discussed further in Section 6.

## 4.3. Evaluation & Out-of-Distribution Generalization

Suppose $\widehat{\mathrm{F}}_n$ is the operator estimated from the training data. Its performance is then evaluated on a separate set of held-out test samples $\{(v_i, \mathrm{G}(v_i))\}_{i=1}^{n_{\text{test}}}$. The test samples may be drawn from the same distribution as the training set or from a different distribution. For example, if the training distribution $\mu$ is a Gaussian process with covariance $\alpha(-\nabla^2+\beta\mathbf{I})^{-\gamma_1}$, the test distribution $\mu_{\text{test}}$ could have a covariance of the form $\alpha(-\nabla^2 + \beta\mathbf{I})^{-\gamma_2}$ for some $\gamma_2 < \gamma_1$. Since the parameter $\gamma_j$ determines the smoothness of the sampled functions, evaluating on a test set with a smaller $\gamma_2$ allows for assessing the model's ability to generalize to functions with lower smoothness.

Moreover, as discussed in Section 4.1, functions are typically sampled using the Karhunen-Loève decomposition. That is, a function can be expressed as $v(x) = \sum_{j=1}^{M} \sqrt{\lambda_j}\,\xi_j\,\varphi_j(x)$, where function samples are generated by drawing values for $\xi_j$. To assess out-of-distribution generalization, one can modify the distribution from which $\xi_j$ is sampled. For example, while training data may be generated by drawing $\xi_j$ from a Gaussian distribution, test samples could be generated using a long-tailed distribution, such as a t-distribution.

# 5. ERROR ANALYSIS AND CONVERGENCE RATES

For a fixed $\mu$, the excess risk $\mathcal{E}_n(\widehat{\mathrm{F}}_n, \mathcal{F}, \mathcal{P}, \mathrm{G})$ defined in Equation (2) consists of two terms: the risk of the estimator $\widehat{\mathrm{F}}_n$ and the risk of the optimal operator in $\mathcal{F}$. The latter term,

$$\inf_{\mathrm{F}\in\mathcal{F}} \mathop{\mathbb{E}}_{(v,w)\sim\mu} \big[\ell\big(\mathrm{F}(v), \mathrm{G}(v)\big)\big],$$

is referred to as the *approximation error* of $\mathcal{F}$. This error arises due to model misspecification– the ground truth G lying outside the class $\mathcal{F}$. Since this error is irreducible and does not vanish even as $n \to \infty$, it is subtracted from the error of the estimator so that the resulting excess risk can go to zero in the limit.

In traditional learning problems, the excess risk $\mathcal{E}_n(\widehat{\mathrm{F}}_n, \mathcal{F}, \mathcal{P}, \mathrm{G})$ is typically referred to as the *statistical error* of the estimator $\widehat{\mathrm{F}}_n$. Statistical error arises because the learner must identify the optimal operator in $\mathcal{F}$ for the distribution $\mu$, despite only having access to a finite number of samples from that distribution. However, in operator learning, unlike traditional learning settings, the excess risk contains at least two additional sources of error beyond the statistical error: the *discretization error* and the *truncation error*. The discretization error arises because the learner has access to samples only on a discrete grid of domain points, but is evaluated on the entire domain. On the other hand, the truncation error arises due to various finite-dimensional approximations of the infinite-dimensional object of interest.

## 5.1. Approximation Error

Most theoretical works in operator learning have focused on quantifying the approximation error of various model classes. One of the earliest works was by Chen & Chen (1995), who introduced a neural network-based architecture called operator networks and established its universality for representing continuous operators on compact subsets. Building on this work, Lu et al. (2021a) proposed DeepONet and proved its universality. While these initial results were qualitative, Lanthaler et al. (2022) provided quantitative error estimates for these architectures. The universality of Fourier Neural Operators (FNOs) and their associated error bounds were established by Kovachki et al. (2021). Similar approximation error estimates were later established for PCA-Net by Lanthaler (2023). For a more comprehensive overview of works on approximation error of various operator classes, we refer the readers to (Kovachki et al. 2024b, Sections 4 & 5).

## 5.2. Truncation Error

Truncation errors can arise from a variety of sources. For example, as discussed in Section 4.1, while samples $v \sim \mathrm{GP}(0, \alpha(-\nabla^2 + \beta\mathbf{I})^{-\gamma})$ has the representation $v = \sum_{j=1}^{\infty} \sqrt{\lambda_j}\xi_j\varphi_j$, only a truncated sum $\sum_{j=1}^{M} \sqrt{\lambda_j}\xi_j\varphi_j$ is used in practice. This introduces a truncation error in the data itself.

Truncation errors also arise during the estimation process. For example, in model (3), only a finite subset of the parameters $(\beta_j)_{j=1}^{\infty}$ can be estimated in practice, introducing a truncation error. More generally, in operator learning literature, the model class $\mathcal{F}$ is often assumed to have a low-dimensional latent structure. Specifically, for any $\mathrm{F} \in \mathcal{F}$, there exists a mapping $h : \mathbb{R}^r \to \mathbb{R}^q$ belonging to another class $\mathcal{H}$ such that $\mathrm{F} = \mathrm{D} \circ h \circ \mathrm{E}$, where E and D are predefined encoder and decoder operators. A common choice for the encoder is a finite-dimensional projection, defined as $\mathrm{E}(v) = (\langle v, \varphi_1\rangle, \ldots, \langle v, \varphi_r\rangle)$, where $(\varphi_j)_{j=1}^{\infty}$ is an orthonormal basis of the input space $\mathcal{V}$. Similarly, given a vector $\zeta \in \mathbb{R}^q$, the decoder can be defined as $\zeta \mapsto \sum_{j=1}^{q} \zeta_j\psi_j$, where $\zeta = (\zeta_1, \ldots, \zeta_q)$ and $(\psi_j)_{j=1}^{\infty}$ is an orthonormal basis of the output space $\mathcal{W}$. Such encoders and decoders naturally introduce a truncation error. Lanthaler et al. (2022) characterized the encoding and decoding errors for the DeepONet architecture, while a similar analysis for PCA-Net was provided in Lanthaler (2023). Furthermore, Liu et al. (2024) studied truncation error of various encoder-decoder

frameworks for the estimation of Lipschitz operators.

Truncation errors are also inherent in FNO estimation (Section 3.2.1), where the infinite sum is approximated by a finite truncation. Subedi & Tewari (2025b) quantify this error for the linear core of FNO. Similarly, PCA-based estimators in functional linear regression (Hörmann & Kidziński 2015, Reimherr 2015) also incur truncation errors as only a finite number of principal components are used in practice.

## 5.3. Discretization Error

Discretization error fundamentally arises from approximating functional operations using numerical methods on a finite-sized grid. Lu et al. (2021a) quantified the discretization error in sampling from a Gaussian process and reconstructing functions via linear interpolation. For DeepONet, Lanthaler et al. (2022) analyzed the error from encoding $v$ using numerical inner products on a discrete grid. More recently, Lanthaler et al. (2024) analyzed discretization error in Fourier Neural Operators (FNOs) during evaluation. For a fixed FNO model F, they bounded the numerical error of evaluating $\mathrm{F}(v)$ on a grid of size $N^d$ instead of the exact evaluation.

**Zero-Shot Super-Resolution:** The ability of a learned operator, trained on data sampled at a coarse grid resolution, to generalize and produce accurate predictions when evaluated on finer grids, without requiring retraining.

A more relevant discretization error arises during the learning process itself rather than model evaluation. It occurs because the estimator is trained on a discrete grid of size $N$ but evaluated at full resolution as $N \to \infty$. Subedi & Tewari (2025a) quantified this error for learning the linear core of FNOs. However, a more practical analysis would consider training on a grid of size $N_1$ and evaluating on a different grid of size $N_2$. Such a general analysis would provide a rigorous foundation for understanding the multiresolution generalization, also known as zero-shot super-resolution, often observed in practice (Li et al. 2021, Section 5.4). Therefore, a key direction for future research is developing a general theory of multiresolution generalization.

The problem of learning from discretized data shares similarities with partial information settings studied in the bandit literature (Lattimore & Szepesvári 2020). Thus, techniques from bandit literature could help in understanding learning theoretic consequences of discretization. Additionally, often in practice, training data is available at multiple grid sizes. A natural way to study this could be through the lens of missing data problems, borrowing tools from the statistical literature (Little & Rubin 2019).

## 5.4. Statistical Error

There is a substantial body of work studying statistical error when $\mathcal{F}$ is a class of linear operators. In the FDA literature, $\mathcal{V}$ is often assumed to be $L^2([0,1])$, the output space $\mathcal{W}$ is typically $\mathbb{R}$, and $\mathcal{F}$ is generally a class of integral transforms. For foundational results in this setting, we refer readers to the seminal works of Hall & Horowitz (2007) and Yuan & Cai (2010), as well as the references therein. A non-technical overview of these results is available in review articles such as Wang et al. (2016) and Morris (2015). These results have been extended to settings where the response is function-valued, say $\mathcal{W} = L^2([0,1])$, in works like Yao et al. (2005), Crambes & Mas (2013), Benatia et al. (2017), and Imaizumi & Kato (2018). Further generalizations to arbitrary Hilbert spaces $\mathcal{V}$ and $\mathcal{W}$ have been developed in (Hörmann & Kidziński 2015, Reimherr 2015, Mollenhauer et al. 2022, de Hoop et al. 2023). All of the works mentioned above generally establish prediction consistency results with associated rates of estimation under the additive noise model $w = \mathrm{F}(v) + \delta$. In contrast, Tabaghi et al. (2019) and Subedi & Tewari (2025a) bound excess risk for learning

linear operators in potentially fully misspecified (agnostic) settings. It is important to note that this list is far from exhaustive and represents only a subset of works from a mature and extensive body of literature on linear operator learning.

Compared to linear operator learning, the statistical theory for nonlinear operator classes $\mathcal{F}$ is relatively underdeveloped. Early works in this area include works on additive and polynomial models (Li & Marx 2008, Yao & Müller 2010). Kadri et al. (2016) extended the theory to the estimation of operators when $\mathcal{F}$ is a reproducing kernel Hilbert space (RKHS) of operators. Additionally, Batlle et al. (2024) developed a kernel-based framework for operator learning and established statistical guarantees for scenarios where data is observed as finite-dimensional vectors through a fixed measurement map. Nelsen & Stuart (2021) generalized the random feature model of Rahimi & Recht (2007) to learning operators, and comprehensive statistical analysis for RFM was later provided by Lanthaler & Nelsen (2023).

Building on the early work of Mhaskar & Hahm (1997), recent work by Kovachki et al. (2024a) provides a statistical analysis for non-parametric operator classes such as Lipschitz operators. However, their findings primarily highlight the hardness of the problem, as they show that the minimax error bound decays very slowly, at a rate of $\sim (\log n)^{-1}$. A related analysis by Liu et al. (2024) also reports similarly pessimistic rates for Lipschitz operators estimation. These results suggest that Lipschitz operators might be too big of a class to learn. Thus, an important future direction is to identify a smaller class $\mathcal{F}$ that captures operators commonly encountered in practice for which efficient statistical estimation is possible.

While most applied works in operator learning rely on neural network-based architectures, their statistical analyses remain limited. Lanthaler et al. (2022) and Gopalani et al. (2024) established generalization bounds for DeepONets, while Kim & Kang (2024) and Benitez et al. (2024) derived bounds for Fourier Neural Operators using Rademacher complexity estimates. However, these bounds typically rely on covering number analysis, leading to an exponential dependence on the parameter size or number of layers. Thus, the resulting bounds are too loose to provide any meaningful insights into the empirical success of these models. In future, shifting focus from deep architectures to tighter analyses of simpler models, such as single-layer networks, may yield deeper insights into the statistical properties of these neural network-based operator models.

## 5.5. Towards a General Statistical Theory of Operator Learning

Beyond analyzing specific operator classes, it would be of interest to develop a general statistical theory for operator learning. Historically, statistical learning theory has focused on developing such general frameworks, beginning with the Vapnik-Chervonenkis (VC) theory introduced by Vapnik & Chervonenkis (1971) and further developed through tools from empirical process theory by Dudley (1978), Talagrand (2005), and others. A key challenge in developing a similar theory for operator learning is identifying an appropriate complexity measure for the operator class $\mathcal{F}$ that effectively quantifies its complexity of learning. Modern statistical learning theory relies primarily on covering numbers to quantify such complexity (van der Vaart & Wellner 1996, van de Geer 2000). Recent work by Reinhardt et al. (2024) has taken a step in this direction by providing estimation bounds for compact operator classes in terms of their covering number. However, covering numbers may not be the right tool in this setting as many fundamental operator classes are inherently

non-compact, making covering number-based bounds vacuous.

For example, consider a simple class of constant functions

$$\mathcal{F} = \{v \mapsto w \mid w \in \mathcal{W} \text{ and } \|w\|_{\mathcal{W}} \leq 1\},$$

which corresponds to the unit ball in the output space $\mathcal{W}$. Since the unit ball in an infinite-dimensional space is non-compact, the covering number of $\mathcal{F}$ is unbounded under any reasonable metric. Yet, learning this class is just a reframing of a mean estimation problem. Given $n$ i.i.d. samples $\{(v_i, w_i)\}_{i=1}^n$, the empirical mean $\widehat{\mathrm{F}}_n = \frac{1}{n}\sum_{i=1}^n w_i$ achieves the standard convergence rate of $\mathcal{E}_n(\widehat{\mathrm{F}}_n, \mathcal{F}, \mathcal{P}, \mathrm{G}) \lesssim n^{-1/2}$ when $\mathcal{W}$ is a separable Hilbert space and $\ell$ is the canonical norm of $\mathcal{W}$. This suggests that a new complexity measure other than covering number is needed—one that meaningfully characterizes learnability for operator classes, much like what VC dimension does for binary classification (Blumer et al. 1989). More precisely, given a class $\mathcal{F}$, is there a complexity measure function $\mathrm{C}_\gamma(\mathcal{F})$ such that $\mathcal{E}_n(\widehat{\mathrm{F}}_n, \mathcal{F}, \mathcal{P}, \mathrm{G}) \xrightarrow{n\to\infty} 0$ if and only if $\mathrm{C}_\gamma(\mathcal{F}) < \infty$ for every $\gamma > 0$? In other words, the measure $\mathrm{C}_\gamma(\mathcal{F})$ tells us when the risk-consistent estimation of $\mathcal{F}$ is possible. A promising direction could be to develop generalizations of the fat-shattering dimension, which has been used to characterize learnability in scalar-valued regression (Bartlett et al. 1996, Alon et al. 1997). The theory of scalar-valued regression, developed in terms of the fat-shattering dimension, suggests that the appropriate complexity measure $\mathrm{C}_\gamma(\mathcal{F})$ also provides a tight bound on the rate at which $\mathcal{E}_n(\widehat{\mathrm{F}}_n, \mathcal{F}, \mathcal{P}, \mathrm{G})$ converges to zero.

## 6. PDE-SPECIFIC OPERATOR LEARNING

Existing works in operator learning typically use neural networks as black-box estimators for solution operators, which makes them broadly applicable across various PDEs. However, these methods often neglect structural properties inherent to specific PDE classes, leading to suboptimal data efficiency. To address this, a few works have attempted to integrate mathematical or physical constraints into the learning process, often referred to as *physics-informed learning*. The central idea is to encode known physical constraints (e.g., boundary conditions, conservation laws) directly into the training objective as a regularization term. Specifically, a physics-informed approach minimizes a regularized loss function of the form

$$\widehat{\mathrm{F}} \in \operatorname*{arg\,min}_{\mathrm{F}\in\mathcal{F}} \left( \frac{\lambda_1}{n} \sum_{i=1}^n \ell(\mathrm{F}(v_i), w_i) + \lambda_2 R(S_n, \mathrm{F}) \right)$$

for some prespecified weights $\lambda_1, \lambda_2 > 0$. Here, $S_n$ is the training sample set, and $R : S_n \times \mathcal{F} \to [0, \infty]$ is a regularization functional encoding PDE priors such as boundary conditions (Li et al. 2024b) or variational formulations (Goswami et al. 2023)).

In our running example of the heat equation (1), suppose $u$ is integrable and vanishes on the boundary. It is well known that the total heat energy is conserved, meaning $\int_{\mathcal{X}} u_t(x)\,dx = \int_{\mathcal{X}} u_0(x)\,dx, \quad \forall t > 0$. A natural way to enforce this constraint in operator learning is through a regularization functional

$$R(S_n, \mathrm{F}) := \frac{1}{n} \sum_{i=1}^n \left| \int_{\mathcal{X}} \big(\mathrm{F}(v_i)(x) - v_i(x)\big)\,dx \right|.$$

This penalizes deviations from the heat conservation law to ensure that the learned operator respects the underlying physical principle.

The use of physics-based constraints in learning models for solving PDEs was formally introduced within the framework of Physics-Informed Neural Networks (PINNs) by Raissi et al. (2019). By restricting the search space from $\mathcal{F}$ to operators F that satisfy physical laws, the effective model complexity is reduced. This generally leads to improved sample efficiency.

While physics-informed learning is an important first step, a more effective approach could be to incorporate these constraints directly into the architecture design, leading to PDE-specific models. For example, consider the time-dependent Schrödinger equation,

$$\mathrm{i}\,\hbar \frac{d\psi_t}{dt} = \mathrm{H}\,\psi_t.$$

The development of surrogate models for Schrödinger's equation has been an active area of research since the 1970s, with many research communities still dedicated to this pursuit. Thus, an off-the-shelf neural network is unlikely to make a meaningful progress. However, a carefully designed operator learning approach tailored specifically to the Schrödinger equation itself could be a valuable addition to the existing toolbox of approximate methods.

The structure of this PDE itself can be used to inform architecture design. For example, when the Hamiltonian H is time-independent, the solution operator takes the form $\mathrm{G} = \exp(-\mathrm{i}\,\frac{t}{\hbar}\,\mathrm{H})$, which is unitary, that is $\mathrm{G}^\dagger\,\mathrm{G} = \mathbf{I}$. Thus, a more data-efficient learning approach may involve parametrizing $\mathcal{F}$ such that every $\mathrm{F} \in \mathcal{F}$ is unitary. Extending existing parameterizations of unitary matrices, such as one by Jarlskog (2005), could provide a useful starting point.

Finally, we end by noting that a similar PDE-informed design philosophy underlies the Green function learning framework proposed by Boullé et al. (2022) and Gin et al. (2021), where the architecture is derived from the observation that solution operators of certain boundary value problems can be expressed as integral operators with Green's functions. Extending this type of principled approach of architectural design to general nonlinear PDEs is an important future direction.

## 7. FUTURE DIRECTIONS

While operator learning holds great potential, scaling it for real-world applications presents key challenges. We now outline some important research directions to address these issues.

### 7.1. Active Data Collection

It is unclear if the iid-based statistical model is the right framework to study operator learning for PDEs. This is because the learner can generate any training data by querying the numerical solver, and thus has no reason to be limited to iid samples from some source distribution. In fact, as generating training data requires computationally expensive numerical solvers, the learner *should* ideally generate data adaptively to ensure that the computational cost of training is justified by saving during evaluation. The model where the learner can adaptively select the data is referred to as active learning model in statistical learning theory. We will propose an active learning model and argue why this is the right model to study operator learning for surrogate modeling of PDE.

Given a sample size budget of $n$, the learner can pick *any* functions $v_1, v_2, \ldots, v_n \in \mathcal{V}$ and get the label $\mathrm{G}(v_i)$ for each $i \in [n]$. Note that this framework where the learner can request labels for any input is generally unrealistic in many problems. For example, for

human data, it may not be feasible to request a label for an individual with an arbitrary feature vector, as such a representative human may not exist in reality. However, this is perfectly realistic in operator learning because the PDE solver can provide a solution for any input function in the appropriate function space $\mathcal{V}$.

With this actively collected training data $(v_i, \mathrm{G}(v_i))_{i=1}^n$, the learner has to produce an estimate $\widehat{\mathrm{F}}_n$ such that

$$\mathcal{E}_n^{\text{active}}(\widehat{\mathrm{F}}_n, \mathcal{F}, \mathcal{P}, \mathrm{G}) := \sup_{\mu \in \mathcal{P}} \left( \mathop{\mathbb{E}}_{v_1, \ldots, v_n} \left[ \mathop{\mathbb{E}}_{v \sim \mu} \left[ \ell\big(\widehat{\mathrm{F}}_n(v), \mathrm{G}(v)\big) \right] \right] - \inf_{\mathrm{F} \in \mathcal{F}} \mathop{\mathbb{E}}_{v \sim \mu} \left[ \ell\big(\mathrm{F}(v), \mathrm{G}(v)\big) \right] \right)$$

vanishes to 0 as $n \to \infty$. The key distinction between $\mathcal{E}_n^{\text{active}}(\widehat{\mathrm{F}}_n, \mathcal{F}, \mathcal{P}, \mathrm{G})$ and the excess risk $\mathcal{E}_n(\widehat{\mathrm{F}}_n, \mathcal{F}, \mathcal{P}, \mathrm{G})$ defined in Equation (2) lies in the definition of $\widehat{\mathrm{F}}_n$. In Equation (2), $\widehat{\mathrm{F}}_n$ is constructed using labeled samples where the inputs $v_i$ are drawn independently from $\mu$. In contrast, for $\mathcal{E}_n^{\text{active}}(\widehat{\mathrm{F}}_n, \mathcal{F}, \mathcal{P}, \mathrm{G})$, the learner has the flexibility to select any $v_1, \ldots, v_n \in \mathcal{V}$. The expectation over $v_1, \ldots, v_n$ accounts for any randomness introduced by the learner in the data generation process.

This seemingly minor distinction can have significant implications for the guarantees that can be established in this setting. For example, when $\mathcal{F}$ is a class of linear operators, Subedi & Tewari (2025b) showed that fast convergence rates of $n^{-\beta}$ for $\beta \gg 1$ can be obtained for many natural families $\mathcal{P}$. This is in stark contrast to the i.i.d. setting, where rates faster than $n^{-1}$ are not possible. Therefore, this result highlights the substantial advantage of active data collection over i.i.d. sampling, at least for linear operator learning. Recent empirical work by Musekamp et al. (2024) also highlights the benefits of active data collection in operator learning. Li et al. (2024a) also showed empirically that an active learning strategy that jointly selects input functions and their resolution improves the data efficiency of FNOs.

The statistical benefits of active learning over passive sampling in traditional learning problems are well-established in the learning theory literature (Hanneke 2013, Settles 1994). Given this potential benefit, developing active data collection strategies and rigorously establishing their statistical benefits is an important direction for future research in operator learning, as emphasized by Azizzadenesheli (2024) in his ICML 2024 tutorial.

## 7.2. Uncertainty Quantification

For operator learning to be deployed in real-world applications, particularly in safety-critical domains, a rigorous framework of uncertainty quantification is essential. Several works have already explored this direction, mostly using Bayesian techniques. For example, Zou et al. (2025) uses Hamiltonian Monte Carlo to sample from the posterior distribution, using the mean for point prediction and the variance as an uncertainty estimate. Similarly, Magnani et al. (2022), Akhare et al. (2023), Guo et al. (2024) propose approximate Bayesian methods for uncertainty quantification, while Ma et al. (2024) proposes conformal prediction to provide distribution-free coverage guarantees.

Beyond ensuring model reliability, uncertainty quantification can actually be used to improve the operator learning pipeline in two ways. First, it allows uncertainty-guided active data collection, which can potentially improve sample efficiency by prioritizing high-uncertainty inputs for labeling. A standard Bayesian active learning strategy involves esti-

mating the empirical posterior covariance to define an acquisition objective

$$U_n(v) = \frac{1}{m}\sum_{i=1}^{m}\left\|\mathrm{F}_i(v) - \frac{1}{m}\sum_{j=1}^{m}\mathrm{F}_j(v)\right\|_{\mathcal{W}}^2,$$

where $\mathrm{F}_1, \ldots, \mathrm{F}_m$ are posterior samples from $n$ training data points. Then, the next input to label is selected as $v_{n+1} = \arg\max_{v\in\mathcal{V}} U_n(v)$, often from a predefined candidate set $S \subset \mathcal{V}$ of finite size for computational efficiency. This is particularly useful in settings where acquiring labeled data is costly, such as physical experiments, where retraining the model is much cheaper than running a new experiment. A similar approach to uncertainty-guided active learning using ensemble predictions was proposed by Musekamp et al. (2024).

Second, uncertainty quantification allows robust decision-making in downstream tasks where operator surrogates are integrated into optimization pipelines. For example, in design optimization (Rao 2019), the surrogate prediction $\widehat{w}$ is used to minimize a downstream objective function $J(\xi, w)$ over a feasible parameter space $\Xi$, subject to design constraints $C(\xi) \leq 0$. A robust optimization can account for uncertainty by solving a min-max problem,

$$\arg\min_{\xi\in\Xi} \max_{w\in B(\widehat{w})} J(\xi, w) \quad \text{subject to} \quad C(\xi) \leq 0,$$

where $B(\widehat{w})$ is an uncertainty set derived from Bayesian posterior samples or conformal prediction.

Thus, a proper uncertainty framework not only improves model reliability but also allows the development of more sample-efficient and robust systems. As such, developing reliable and scalable uncertainty quantification methods is a crucial future direction in operator learning.

## 7.3. Local Averaging and Ensemble Methods

Given a labeled dataset $\{(v_i, w_i)\}_{i=1}^n$, local averaging methods such as nearest neighbors or Nadaraya-Watson kernel smoothing define an estimator $\widehat{\mathrm{F}}_n$ as

$$\widehat{\mathrm{F}}_n(v) = \sum_{i=1}^{n} \widehat{\alpha}_i(v)\, w_i,$$

where $\widehat{\alpha}_i : \mathcal{V} \to [0,1]$ are weights satisfying $\sum_{i=1}^n \widehat{\alpha}_i(v) = 1$ for every $v \in \mathcal{V}$. This method outputs a weighted average of the training labels, with weights determined by the new input. Prior works have established the consistency of $k$-nearest neighbors for functional regression with both scalar response (Laloë 2008) and functional response (Lian 2011) as well as the consistency of Nadaraya-Watson estimators (Aspirot et al. 2009). However, their empirical performance in surrogate modeling of PDEs has not yet been studied. These methods may provide competitive alternatives to parametric models, particularly in data-scarce settings. Additionally, their ability to update efficiently makes them well-suited for settings where the data is acquired sequentially.

Ensemble methods, like local averaging, compute the weighted averages of multiple predictions. Formally, they define an estimator

$$\widehat{\mathrm{F}}_n(v) = \sum_{j=1}^{m} \widehat{\beta}_j\, \widehat{\Psi}_j(v), \quad \forall v \in \mathcal{V},$$

where $\widehat{\Psi}_j : \mathcal{V} \to \mathcal{W}$ are base predictors trained on all or part of the dataset $\{(v_i, w_i)\}_{i=1}^n$, and $\widehat{\beta}_j \in [0, \infty)$ are learned weights. Common ensemble techniques include bagging, boosting, and random forests.

Ensemble methods, especially boosting, have been shown to outperform neural networks in various applications, particularly when data is scarce, heavy-tailed, skewed, or highly variable (McElfresh et al. 2023). Given that operator learning for PDEs is a data-scarce settings and PDEs on complex geometries introduce data irregularities, ensemble methods may provide strong alternatives to neural operators. Functional regression literature already provide a foundation on these methods, with works on boosting (Ferraty & Vieu 2009, Tutz & Gertheiss 2010) and implementations like FDboost (Brockhaus et al. 2020), as well as works on bagging for functional covariates (Secchi et al. 2013, Kim & Lim 2022) and functional random forests (Möller et al. 2016, Rahman et al. 2019). While some methods handle functional targets (see (Brockhaus et al. 2020, Section 7)), most are designed for scalar-valued outputs. Extending these techniques to Banach space-valued targets is a useful starting point. Moreover, since existing functional ensemble methods are typically limited to small datasets, scaling them using computational techniques from operator learning (Kovachki et al. 2023) is another natural direction. Finally, rather than replacing neural operators, ensemble methods could also complement them as an ensemble of smaller neural operator models may outperform a single large one.

## 7.4. Frictionless Reproducibility

A key future direction for advancing operator learning is fostering what Donoho (2024) describes as frictionless reproducibility– a research environment where scientific findings, computational experiments, and learning models can be easily replicated, verified, and built upon with minimal effort.

A foundational step in this direction is the development of standard benchmark datasets. Lu et al. (2022) took an important step by releasing 16 benchmark datasets. Additionally, the `NeuralOperator` library (Kovachki et al. 2023, Kossaifi et al. 2024) has a few datasets for benchmarking. However, most existing datasets are on toy problems, which, while valuable for proof-of-concept validation, do not fully capture the complexity of real-world applications. Thus, there is a clear need to expand existing benchmarks to include large-scale datasets for applications such as climate modeling, materials science, and molecular dynamics, where operator learning is expected to have a major impact. Additionally, a single large-scale dataset could significantly accelerate progress: for example, consider the central role ImageNet (Deng et al. 2009) played in transforming computer vision.

Beyond datasets, another critical component of frictionless reproducibility is open-source implementations of operator learning models. Some progress has already been made in this direction with the development of libraries such as `DeepXDE` (Lu et al. 2021b) and `NeuralOperator` (Kossaifi et al. 2024, Kovachki et al. 2023). Additionally, many independent works have publicly released their models and datasets. However, these models and datasets are often scattered across different repositories, making it difficult for researchers to locate and systematically compare them. To address this, a centralized platform for hosting operator learning datasets and models would be of huge benefit to the community. Existing platforms such as Hugging Face could be used for sharing datasets, pre-trained models, and benchmarking results in a way that allows seamless collaboration and reproducibility.

### 7.5. Scaling Sample and Model Size

Recent works on scaling laws (Kaplan et al. 2020, Zhai et al. 2022) suggest that model performance continues to improve as sample size, model size, and computational resources increase. However, most existing works in operator learning are in a data-scarce setting, with models that are much smaller than those used in computer vision and language modeling. Thus, a key future direction is to explore how we can train a large-scale operator learning model on massive datasets. There are two key challenges in scaling operator learning methods. First, obtaining labeled data is expensive since obtaining each sample requires running a computationally expensive solver or conducting costly experiments. Second, widely used architectures such as DeepONet and FNOs do not scale as efficiently as transformer-based models, limiting the size of models that can be trained with available resources.

One approach to mitigating data limitations is training a foundation model on diverse PDE datasets and fine-tuning it for specific tasks, as explored by Subramanian et al. (2024). On the model scaling front, transformer-based architectures have been proposed for operator learning (Cao 2021, Hao et al. 2023, Li et al. 2023), but their quadratic computational cost with grid size limits scalability. Developing more efficient tokenization strategies for transformers to enable large-scale operator learning without excessive computational overhead remains an important future direction.

## 8. CONCLUSION

In this article, we reviewed recent developments in operator learning, emphasizing its statistical foundations and connections to functional data analysis (FDA). That said, the overall goal of the FDA differs slightly from that of operator learning. In FDA, the focus is on statistical inference, typically using RKHS-based frameworks. As a result, FDA methods do not always scale to large datasets. In contrast, operator learning primarily aims at prediction, with an emphasis on creating computationally efficient methods that can be used to train large models and handle large datasets. However, bridging these fields could be mutually beneficial. FDA's theoretical tools can be used for the analysis of operator learning methods, while methodological advances in operator learning can improve the scalability of FDA techniques.

Applied research has been the primary driver of advancements in operator learning, and this trend will likely continue. However, unlike other machine learning fields, there is a unique opportunity for theory to catch up more rapidly due to the structured nature of operator learning problems. Here, the ground truth operator is well-defined, and the learner typically has partial prior knowledge along with a strong oracle access to the operator via a numerical solver. This is in contrast to language or vision modeling, where a ground truth function may not even exist or the learner may only have limited oracle access through available samples. As discussed in Section 6, this structure in operator learning allows for designing PDE-specific architectures and methods, which may also introduce further mathematical regularity that enables easier theoretical analysis. Moreover, a key reason for the gap between empirical performance and theoretical lower bounds in modern machine learning is the reliance on the iid assumption, which often fails to reflect practical scenarios where practitioners actively select the mosy informative samples. While the iid model historically became standard as a reasonable tradeoff between practical relevance and analytical tractability, operator learning allows for greater flexibility in data acquisition. This opens the door to alternative learning models that better align with real-world

data collection while being theoretically as tractable as the iid framework. Studying such alternative learning models can potentially bridge the gap between theory and practice.

## DISCLOSURE STATEMENT

The authors are not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

## ACKNOWLEDGMENTS

We acknowledge the support of NSF via grant DMS-2413089.

## LITERATURE CITED

Aghili J, Franck E, Hild R, Michel-Dansac V, Vigon V. 2025. Accelerating the convergence of newton's method for nonlinear elliptic pdes using fourier neural operators. *Communications in Nonlinear Science and Numerical Simulation* 140:108434

Akhare D, Luo T, Wang JX. 2023. Diffhybrid-uq: uncertainty quantification for differentiable hybrid neural modeling. *arXiv:2401.00161*

Alon N, Ben-David S, Cesa-Bianchi N, Haussler D. 1997. Scale-sensitive dimensions, uniform convergence, and learnability. *Journal of the ACM (JACM)* 44(4):615–631

Anandkumar A. 2023. AI that connects the digital and physical worlds. TED Talk. Available at: `https://www.ted.com/talks/anima_anandkumar_ai_that_connects_the_digital_and_physical_worlds`

Ananthaswamy A. 2021. Latest neural nets solve world's hardest equations faster than ever before. *Quanta Magazine* Available at: `https://www.quantamagazine.org/latest-neural-nets-solve-worlds-hardest-equations-faster-than-ever-before-20210419/`

Aspirot L, Bertin K, Perera G. 2009. Asymptotic normality of the nadaraya–watson estimator for nonstationary functional data and applications to telecommunications. *Journal of Nonparametric Statistics* 21(5):535–551

Azizzadenesheli K. 2024. Neural operator learning. *International Conference on Machine Learning (Tutorial)*

Azizzadenesheli K, Kovachki N, Li Z, Liu-Schiaffini M, Kossaifi J, Anandkumar A. 2024. Neural operators for accelerating scientific simulations and design. *Nature Reviews Physics* 6(5):320–328

Bartlett PL, Long PM, Williamson RC. 1996. Fat-shattering and the learnability of real-valued functions. *Journal of Computer and System Sciences* 52(3):434–452

Batlle P, Darcy M, Hosseini B, Owhadi H. 2024. Kernel methods are competitive for operator learning. *Journal of Computational Physics* 496:112549

Benatia D, Carrasco M, Florens JP. 2017. Functional linear regression with functional response. *Journal of econometrics* 201(2):269–291

Benitez JAL, Furuya T, Faucher F, Kratsios A, Tricoche X, de Hoop MV. 2024. Out-of-distributional risk bounds for neural operators with applications to the helmholtz equation. *Journal of Computational Physics* :113168

Bhattacharya K, Hosseini B, Kovachki NB, Stuart AM. 2021. Model reduction and neural networks for parametric pdes. *The SMAI journal of computational mathematics* 7:121–157

Blumer A, Ehrenfeucht A, Haussler D, Warmuth MK. 1989. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)* 36(4):929–965

Boullé N, Earls CJ, Townsend A. 2022. Data-driven discovery of green's functions with human-understandable deep learning. *Scientific reports* 12(1):4824

Boullé N, Halikias D, Townsend A. 2023. Elliptic pde learning is provably data-efficient. *Proceedings of the National Academy of Sciences* 120(39):e2303904120

Boullé N, Townsend A. 2024. A mathematical guide to operator learning. vol. 25 of *Handbook of Numerical Analysis*. Elsevier, 83–125

Brault R, Heinonen M, Buc F. 2016. Random fourier features for operator-valued kernels, In *Asian Conference on Machine Learning*, pp. 110–125, PMLR

Brockhaus S, Rügamer D, Greven S. 2020. Boosting functional regression models with fdboost. *Journal of Statistical Software* 94:1–50

Cao S. 2021. Choose a transformer: Fourier or galerkin. *Advances in neural information processing systems* 34:24924–24940

Chen D, Hall P, Müller HG. 2011. Single and multiple index functional regression models with nonparametric link. *The Annals of Statistics* :1720–1747

Chen T, Chen H. 1995. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE transactions on neural networks* 6(4):911–917

Cheng AHD. 1984. Darcy's flow with variable permeability: A boundary integral solution. *Water Resources Research* 20(7):980–984

Crambes C, Mas A. 2013. Asymptotics of prediction in functional linear regression with functional outputs. *Bernoulli* 19(5B):2627–2651

de Hoop MV, Kovachki NB, Nelsen NH, Stuart AM. 2023. Convergence rates for learning linear operators from noisy data. *SIAM/ASA Journal on Uncertainty Quantification* 11(2):480–513

Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. 2009. Imagenet: A large-scale hierarchical image database, In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee

Dodziuk J. 1981. Eigenvalues of the laplacian and the heat equation. *The American Mathematical Monthly* 88(9):686–695

Donoho D. 2024. Data science at the singularity. *Harvard Data Science Review* 6(1)

Dudley RM. 1978. Central limit theorems for empirical measures. *The Annals of Probability* :899–929

ECMWF. 2025. Fourcastnet ML model: Wind and geopotential heights at various pressure levels. *European Centre for Medium-Range Weather Forecasts* Available at: `https://charts.ecmwf.int/products/fourcast_medium-uv-z`

Evans LC. 2022. Partial differential equations, vol. 19. American Mathematical Society

Ferraty F, Vieu P. 2009. Additive prediction and boosting for functional data. *Computational Statistics & Data Analysis* 53(4):1400–1413

Gin CR, Shea DE, Brunton SL, Kutz JN. 2021. Deepgreen: deep learning of green's functions for nonlinear boundary value problems. *Scientific reports* 11(1):21614

Gopakumar V, Pamela S, Zanisi L, Li Z, Gray A, et al. 2024. Plasma surrogate modelling using fourier neural operators. *Nuclear Fusion* 64(5):056025

Gopalani P, Karmakar S, Kumar D, Mukherjee A. 2024. Towards size-independent generalization bounds for deep operator nets. *Transactions on Machine Learning Research*

Goswami S, Bora A, Yu Y, Karniadakis GE. 2023. Physics-informed deep neural operator networks. In *Machine Learning in Modeling and Simulation: Methods and Applications*. Springer, 219–254

Guo L, Wu H, Wang Y, Zhou W, Zhou T. 2024. Ib-uq: Information bottleneck based uncertainty quantification for neural function regression and neural operator learning. *Journal of Computational Physics* 510:113089

Gupta G, Xiao X, Bogdan P. 2021. Multiwavelet-based operator learning for differential equations. *Advances in neural information processing systems* 34:24048–24062

Hall P, Horowitz JL. 2007. Methodology and convergence rates for functional linear regression. *Annals of statistics* 35(1):70–91

Hanneke S. 2013. A statistical theory of active learning. *Foundations and Trends in Machine Learn-*

*ing* :1–212

Hao Z, Wang Z, Su H, Ying C, Dong Y, et al. 2023. Gnot: A general neural operator transformer for operator learning, In *International Conference on Machine Learning*, pp. 12556–12569, PMLR

Harvey AC. 1978. Linear regression in the frequency domain. *International Economic Review* :507–512

Hörmann S, Kidziński Ł. 2015. A note on estimation in hilbertian linear models. *Scandinavian journal of statistics* 42(1):43–62

Hsing T, Eubank R. 2015. Theoretical foundations of functional data analysis, with an introduction to linear operators, vol. 997. John Wiley & Sons

Hunter JK. 2023. Notes on partial differential equations. Available at: `https://www.math.ucdavis.edu/~hunter/pdes/pde_notes.pdf`

Imaizumi M, Kato K. 2018. Pca-based estimation for functional linear regression with functional responses. *Journal of multivariate analysis* 163:15–36

Jarlskog C. 2005. A recursive parametrization of unitary matrices. *Journal of mathematical physics* 46(10)

Jumper J, Evans R, Pritzel A, Green T, Figurnov M, et al. 2021. Highly accurate protein structure prediction with alphafold. *nature* 596(7873):583–589

Kadri H, Duflos E, Preux P, Canu S, Davy M. 2010. Nonlinear functional regression: a functional rkhs approach, In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 374–380

Kadri H, Duflos E, Preux P, Canu S, Rakotomamonjy A, Audiffren J. 2016. Operator-valued kernels for learning from functional response data. *Journal of Machine Learning Research* 17(20):1–54

Kadri H, Rabaoui A, Preux P, Duflos E, Rakotomamonjy A. 2011. Functional regularized least squares classification with operator-valued kernels, In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pp. 993–1000

Kaplan J, McCandlish S, Henighan T, Brown TB, Chess B, et al. 2020. Scaling laws for neural language models. *arXiv:2001.08361*

Kim H, Lim Y. 2022. Bootstrap aggregated classification for sparse functional data. *Journal of Applied Statistics* 49(8):2052–2063

Kim T, Kang M. 2024. Bounding the rademacher complexity of fourier neural operators. *Machine Learning* 113(5):2467–2498

Kossaifi J, Kovachki N, Li Z, Pitt D, Liu-Schiaffini M, et al. 2024. A library for learning neural operators. *arXiv:2412.10354*

Kovachki N, Lanthaler S, Mishra S. 2021. On universal approximation and error bounds for fourier neural operators. *Journal of Machine Learning Research* 22(290):1–76

Kovachki N, Li Z, Liu B, Azizzadenesheli K, Bhattacharya K, et al. 2023. Neural operator: Learning maps between function spaces with applications to pdes. *Journal of Machine Learning Research* 24(89):1–97

Kovachki NB, Lanthaler S, Mhaskar H. 2024a. Data complexity estimates for operator learning. *arXiv:2405.15992*

Kovachki NB, Lanthaler S, Stuart AM. 2024b. Operator learning: Algorithms and analysis. *Handbook of Numerical Analysis* 25:419–467

Laloë T. 2008. A k-nearest neighbor approach for functional regression. *Statistics & probability letters* 78(10):1189–1193

Lanthaler S. 2023. Operator learning with PCA-Net: upper and lower complexity bounds. *Journal of Machine Learning Research* 24(318):1–67

Lanthaler S, Mishra S, Karniadakis GE. 2022. Error estimates for deeponets: A deep learning framework in infinite dimensions. *Transactions of Mathematics and Its Applications* 6(1):tnac001

Lanthaler S, Nelsen NH. 2023. Error bounds for learning with vector-valued random features, In *Advances in Neural Information Processing Systems*

Lanthaler S, Stuart AM, Trautner M. 2024. Discretization error of fourier neural operators.

*arXiv:2405.02221*

Lattimore T, Szepesvári C. 2020. Bandit algorithms. Cambridge University Press

Li B, Marx BD. 2008. Sharpening p-spline signal regression. *Statistical Modelling* 8(4):367–383

Li S, Yu X, Xing W, Kirby R, Narayan A, Zhe S. 2024a. Multi-resolution active learning of fourier neural operators, In *International Conference on Artificial Intelligence and Statistics*, pp. 2440–2448, PMLR

Li Z, Kovachki N, Azizzadenesheli K, Liu B, Bhattacharya K, et al. 2020a. Neural operator: Graph kernel network for partial differential equations. *arXiv:2003.03485*

Li Z, Kovachki N, Azizzadenesheli K, Liu B, Bhattacharya K, et al. 2021. Fourier neural operator for parametric partial differential equations. *International Conference on Learning Representations*

Li Z, Kovachki N, Azizzadenesheli K, Liu B, Stuart A, et al. 2020b. Multipole graph neural operator for parametric partial differential equations. *Advances in Neural Information Processing Systems* 33:6755–6766

Li Z, Meidani K, Farimani AB. 2023. Transformer for partial differential equations' operator learning. *Transactions on Machine Learning Research*

Li Z, Zheng H, Kovachki N, Jin D, Chen H, et al. 2024b. Physics-informed neural operator for learning partial differential equations. *ACM/JMS Journal of Data Science* 1(3):1–27

Lian H. 2011. Convergence of functional k-nearest neighbor regression estimate with functional responses. *Electronic Journal of Statistics* 5:31–40

Little RJ, Rubin DB. 2019. Statistical analysis with missing data, vol. 793. John Wiley & Sons

Liu H, Yang H, Chen M, Zhao T, Liao W. 2024. Deep nonparametric estimation of operators between infinite dimensional spaces. *Journal of Machine Learning Research* 25(24):1–67

Lord GJ, Powell CE, Shardlow T. 2014. An introduction to computational stochastic pdes, vol. 50. Cambridge University Press

Lu L, Jin P, Pang G, Zhang Z, Karniadakis GE. 2021a. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature machine intelligence* 3(3):218–229

Lu L, Meng X, Cai S, Mao Z, Goswami S, et al. 2022. A comprehensive and fair comparison of two neural operators (with practical extensions) based on fair data. *Computer Methods in Applied Mechanics and Engineering* 393:114778

Lu L, Meng X, Mao Z, Karniadakis GE. 2021b. DeepXDE: A deep learning library for solving differential equations. *SIAM Review* 63(1):208–228

Ma Z, Azizzadenesheli K, Anandkumar A. 2024. Calibrated uncertainty quantification for operator learning via conformal prediction. *arXiv:2402.01960*

Magnani E, Krämer N, Eschenhagen R, Rosasco L, Hennig P. 2022. Approximate bayesian neural operators: Uncertainty quantification for parametric pdes. *arXiv preprint arXiv:2208.01565*

Mallat S. 1999. A wavelet tour of signal processing. Elsevier

McElfresh D, Khandagale S, Valverde J, Prasad C V, Ramakrishnan G, et al. 2023. When do neural nets outperform boosted trees on tabular data? *Advances in Neural Information Processing Systems* 36:76336–76369

Mhaskar HN, Hahm N. 1997. Neural networks for functional approximation and system identification. *Neural Computation* 9(1):143–159

Micchelli CA, Pontil M. 2005. On learning vector-valued functions. *Neural computation* 17(1):177–204

Minh HQ. 2016. Operator-valued bochner theorem, fourier feature maps for operator-valued kernels, and vector-valued learning. *arXiv:1608.05639*

Mollenhauer M, Mücke N, Sullivan T. 2022. Learning linear operators: Infinite-dimensional regression as a well-behaved non-compact inverse problem. *arXiv:2211.08875*

Möller A, Tutz G, Gertheiss J. 2016. Random forests for functional covariates. *Journal of Chemometrics* 30(12):715–725

Mora C, Yousefpour A, Hosseinmardi S, Owhadi H, Bostanabad R. 2025. Operator learning with

gaussian processes. *Computer Methods in Applied Mechanics and Engineering* 434:117581
Morris JS. 2015. Functional regression. *Annual Review of Statistics and Its Application* 2(1):321–359
Musekamp D, Kalimuthu M, Holzmüller D, Takamoto M, Niepert M. 2024. Active learning for neural pde solvers. *arXiv:2408.01536*
Nelsen NH, Stuart AM. 2021. The random feature model for input-output maps between banach spaces. *SIAM Journal on Scientific Computing* 43(5):A3212–A3243
Nelsen NH, Stuart AM. 2024. Operator learning using random features: A tool for scientific computing. *SIAM Review* 66(3):535–571
Rahimi A, Recht B. 2007. Random features for large-scale kernel machines. *Advances in neural information processing systems* 20
Rahman R, Dhruba SR, Ghosh S, Pal R. 2019. Functional random forest with applications in dose-response predictions. *Scientific reports* 9(1):1628
Raissi M, Perdikaris P, Karniadakis GE. 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics* 378:686–707
Rao SS. 2019. Engineering optimization: theory and practice. John Wiley & Sons
Reimherr M. 2015. Functional regression with repeated eigenvalues. *Statistics & Probability Letters* 107:62–70
Reinhardt N, Wang S, Zech J. 2024. Statistical learning theory for neural operators. *arXiv:2412.17582*
Secchi P, Vantini S, Vitelli V. 2013. Bagging voronoi classifiers for clustering spatial functional data. *International journal of applied earth observation and geoinformation* 22:53–64
Settles B. 1994. Active learning literature survey. *Machine Learning* 15(2):201–221
Son H, Jang JW, Han WJ, Hwang HJ. 2021. Sobolev training for physics informed neural networks. *arXiv preprint arXiv:2101.08932*
Stepaniants G. 2023. Learning partial differential equations in reproducing kernel hilbert spaces. *Journal of Machine Learning Research* 24(86):1–72
Subedi U, Tewari A. 2025a. Controlling statistical, discretization, and truncation errors in learning fourier linear operators. *arXiv:2408.09004*
Subedi U, Tewari A. 2025b. On the benefits of active data collection in operator learning. *arXiv:2410.19725*
Subramanian S, Harrington P, Keutzer K, Bhimji W, Morozov D, et al. 2024. Towards foundation models for scientific machine learning: Characterizing scaling and transfer behavior. *Advances in Neural Information Processing Systems* 36
Tabaghi P, de Hoop M, Dokmanić I. 2019. Learning schatten–von neumann operators. *arXiv preprint arXiv:1901.10076*
Talagrand M. 2005. The generic chaining: upper and lower bounds of stochastic processes. Springer Science & Business Media
Tang Y, Kurths J, Lin W, Ott E, Kocarev L. 2020. Introduction to focus issue: When machine learning meets complex systems: Networks, chaos, and nonlinear dynamics. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 30(6)
The Nobel Committee. 2024. The Nobel Prize in Chemistry 2024. Accessed: 2025-02-09
Trefethen LN. 2019. Approximation theory and approximation practice, extended edition. SIAM
Tripura T, Chakraborty S. 2023. Wavelet neural operator for solving parametric partial differential equations in computational mechanics problems. *Computer Methods in Applied Mechanics and Engineering* 404:115783
Tutz G, Gertheiss J. 2010. Feature extraction in signal regression: a boosting technique for functional data regression. *Journal of Computational and Graphical Statistics* 19(1):154–174
Umetani N, Bickel B. 2018. Learning three-dimensional flow for interactive aerodynamic design. *ACM Transactions on Graphics (TOG)* 37(4):1–10

van de Geer S. 2000. Empirical process theory and applications. Cambridge University Press

van der Vaart AW, Wellner JA. 1996. Weak convergence and empirical processes, vol. 126. Springer New York

Vapnik V. 2000. The nature of statistical learning theory. Springer, 2nd ed.

Vapnik V, Chervonenkis AY. 1971. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications* 16(2):264–280

Wang H, Fu T, Du Y, Gao W, Huang K, et al. 2023. Scientific discovery in the age of artificial intelligence. *Nature* 620(7972):47–60

Wang JL, Chiou JM, Müller HG. 2016. Functional data analysis. *Annual Review of Statistics and its application* 3(1):257–295

Williams CK, Rasmussen CE. 2006. Gaussian processes for machine learning, vol. 2. MIT press Cambridge, MA

Yao F, Müller HG. 2010. Functional quadratic regression. *Biometrika* 97(1):49–64

Yao F, Müller HG, Wang JL. 2005. Functional linear regression analysis for longitudinal data. *The Annals of Statistics* 33(6):2873–2903

You H, Zhang Q, Ross CJ, Lee CH, Hsu MC, Yu Y. 2022a. A physics-guided neural operator learning approach to model biological tissues from digital image correlation measurements. *Journal of Biomechanical Engineering* 144(12):121012

You H, Zhang Q, Ross CJ, Lee CH, Yu Y. 2022b. Learning deep implicit fourier neural operators (ifnos) with applications to heterogeneous material modeling. *Computer Methods in Applied Mechanics and Engineering* 398:115296

Yuan M, Cai TT. 2010. A reproducing kernel hilbert space approach to functional linear regression. *Annals of statistics* 38(6):3412–3444

Zhai X, Kolesnikov A, Houlsby N, Beyer L. 2022. Scaling vision transformers, In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12104–12113

Zhang X, Wang L, Helwig J, Luo Y, Fu C, et al. 2023. Artificial intelligence for science in quantum, atomistic, and continuum systems. *arXiv preprint arXiv:2307.08423*

Zheng H, Nie W, Vahdat A, Azizzadenesheli K, Anandkumar A. 2023. Fast sampling of diffusion models via operator learning, In *International conference on machine learning*, pp. 42390–42402, PMLR

Zhou T, Wan X, Huang DZ, Li Z, Peng Z, et al. 2024. Ai-aided geometric design of anti-infection catheters. *Science Advances* 10(1):eadj1741

Zou Z, Meng X, Karniadakis GE. 2025. Uncertainty quantification for noisy inputs–outputs in physics-informed neural networks and neural operators. *Computer Methods in Applied Mechanics and Engineering* 433:117479