Nonlocal Attention Operator: Materializing Hidden Knowledge Towards Interpretable Physics Discovery

Yue Yu*
Department of Mathematics,
Lehigh University,
Bethlehem, PA 18015, USA
yuy214@lehigh.edu

Ning Liu
Global Engineering and
Materials, Inc.,
Princeton, NJ 08540, USA

Fei Lu
Department of Mathematics,
Johns Hopkins University,
Baltimore, MD 21218, USA

Tian Gao
IBM Research,
Yorktown Heights,
NY 10598, USA

Siavash Jafarzadeh
Department of Mathematics,
Lehigh University,
Bethlehem, PA 18015, USA

Stewart Silling
Center for Computing Research,
Sandia National Laboratories,
Albuquerque, NM 87123, USA

Abstract

Despite the recent popularity of attention-based neural architectures in core AI fields like natural language processing (NLP) and computer vision (CV), their potential in modeling complex physical systems remains under-explored. Learning problems in physical systems are often characterized as discovering operators that map between function spaces based on a few instances of function pairs. This task frequently presents a severely ill-posed PDE inverse problem. In this work, we propose a novel neural operator architecture based on the attention mechanism, which we coin Nonlocal Attention Operator (NAO), and explore its capability towards developing a foundation physical model. In particular, we show that the attention mechanism is equivalent to a double integral operator that enables nonlocal interactions among spatial tokens, with a data-dependent kernel characterizing the inverse mapping from data to the hidden parameter field of the underlying operator. As such, the attention mechanism extracts global prior information from training data generated by multiple systems, and suggests the exploratory space in the form of a nonlinear kernel map. Consequently, NAO can address ill-posedness and rank deficiency in inverse PDE problems by encoding regularization and achieving generalizability. We empirically demonstrate the advantages of NAO over baseline neural models in terms of generalizability to unseen data resolutions and system states. Our work not only suggests a novel neural operator architecture for learning interpretable foundation models of physical systems, but also offers a new perspective towards understanding the attention mechanism. Our code and data accompanying this paper are available at https://github.com/fishmoon1234/NAO.

1 Introduction

The interpretability of machine learning (ML) models has become increasingly important from the security and robustness standpoints [Rudin et al., 2022, Molnar, 2020]. This is particularly true in physics modeling problems that can affect human lives, where not only the accuracy but also the transparency of data-driven models are essential in making decisions [Coorey et al., 2022, Ferrari and Willcox, 2024]. Nevertheless, it remains challenging to discover the underlying physical system and the governing mechanism from data. Taking the material modeling task for instance, given that only the deformation field is observable, the goal of discovering the underlying material parame-

---
*Corresponding Author

ter field and mechanism presents an ill-posed unsupervised learning task. That means, even if an ML model can serve as a good surrogate to predict the corresponding loading field from a given deformation field, its inference of the material parameters can still drastically deteriorate.

To discover an interpretable mechanism for physical systems, a major challenge is to infer the governing laws of these systems that are often high- or infinite-dimensional, from data that are comprised of discrete measurements of continuous functions. Therefore, a data-driven surrogate model needs to learn not only the mapping between input and output function pairs, but also the mapping from given function pairs to the hidden state. From the PDE-based modeling standpoint, learning a surrogate model corresponds to a forward problem, whereas inferring the underlying

Figure 1: Illustration of NAO's architecture.

mechanism is an inverse problem. The latter is generally an enduring ill-posed problem, especially when the measurements are scarce. Unfortunately, such an ill-posedness issue may become even more severe in neural network models, due to the inherent bias of neural network approximations [Xu et al., 2019]. To tackle this challenge, many deep learning methods have recently been proposed as inverse PDE solvers [Fan and Ying, 2023, Molinaro et al., 2023, Jiang et al., 2022, Chen et al., 2023]. The central idea is to incorporate prior information into the learning scheme, in the form of governing PDEs [Yang et al., 2021, Li et al., 2021], regularizers [Dittmer et al., 2020, Obmann et al., 2020, Ding et al., 2022, Chen et al., 2023], or additional operator structures [Uhlmann, 2009, Lai et al., 2019, Yilmaz, 2001]. However, such prior information is often either unavailable or problem-specific in complex systems. As a result, these methods can only solve the inverse problem for a particular system, and one has to start from scratch when the system varies (e.g., when the material of the specimen undergoes degradation in a material modeling task).

In this work, we propose **Nonlocal Attention Operator (NAO)**, a novel attention-based neural operator architecture to simultaneously solve both forward and inverse modeling problems. Neural operators (NOs) [Li et al., 2020a,c] learn mappings between infinite-dimensional function spaces in the form of integral operators, hence they provide promising tools for the discovery of continuum physical laws by manifesting the mapping between spatial and/or spatiotemporal data; see You et al. [2022], Liu et al. [2024a,b, 2023], Ong et al. [2022], Cao [2021], Li et al. [2019, 2021], Goswami et al. [2022], Gupta et al. [2021] and references therein. However, most NOs focus on providing an efficient surrogate for the underlying physical system as a forward solver. They are often employed as black-box universal approximators but lack interpretability of the underlying physical laws. In contrast, the key innovation of NAO is that it introduces a kernel map based on the attention mechanism for simultaneous learning of the operator and the kernel map. As such, the kernel map automatically infers the context of the underlying physical system in an unsupervised manner. Intuitively, the attention mechanism extracts hidden knowledge from multiple systems by providing a function space of identifiability for the kernels, which acts as an automatic data-driven regularizer and endows the learned model's generalizability to new and unseen system states.

In this context, NAO learns a kernel map using the attention mechanism and simultaneously solves both the forward and inverse problems. The kernel map, whose parameters extract the global information about the kernel from multiple systems, efficiently infers resolution-invariant kernels from new datasets. As a consequence, NAO can achieve interpretability of the nonlocal operator and enable the discovery of hidden physical laws. **Our key contributions** include:

- We bridge the divide between inverse PDE modeling and physics discovery tasks, and present a method to simultaneously perform physics modeling (forward PDE) and mechanism discovery (inverse PDE).

- We propose a novel neural operator architecture NAO, based on the principle of contextual discovery from input/output function pairs through a kernel map constructed from multiple physical systems. As such, NAO is generalizable to new and unseen physical systems, and offers meaningful physical interpretation through the discovered kernel.

- We provide theoretical analysis to show that the attention mechanism in NAO acts to provide the space of identifiability for the kernels from the training data, which reveals its ability to resolve ill-posed inverse PDE problems.
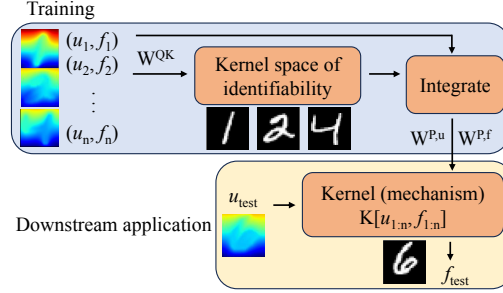
- We conduct experiments on zero-shot learning to new and unseen physical systems, demonstrating the generalizability of NAO in both forward and inverse PDE problems.

2 Background and related work

Our work resides at the intersection of operator learning, attention-based models, and forward and inverse problems of PDEs. The ultimate goal is to model multiple physical systems from data while simultaneously discovering the hidden mechanism.

Neural operator for hidden physics. Learning complex physical systems directly from data is ubiquitous in scientific and engineering applications [Ghaboussi et al., 1998, Liu et al., 2024c, Ghaboussi et al., 1991, Carleo et al., 2019, Karniadakis et al., 2021, Zhang et al., 2018, Cai et al., 2022, Pfau et al., 2020, He et al., 2021, Besnard et al., 2006]. In many applications, the underlying governing laws are unknown, hidden in data to be revealed by physical models. Ideally, these models should be *interpretable* for domain experts, who can then use these models to make further predictions and expand the understanding of the target physical system. Also, these models should be *resolution-invariant*. Neural operators are designed to learn mappings between infinite-dimensional function spaces [Li et al., 2020a,b,c, You et al., 2022, Ong et al., 2022, Cao, 2021, Liu et al., 2019, 2021, Goswami et al., 2022, Gupta et al., 2021]. As a result, NOs provide a promising tool for the discovery of physical laws by manifesting the mapping between spatial and/or spatio-temporal data.

Forward and inverse PDE problems. Most current NOs focus on providing an efficient surrogate for the underlying physical system as a forward PDE solver. They are often employed as black-box universal approximators without interpretability of the underlying physical laws. Conversely, several deep learning methods have been proposed as inverse PDE solvers [Fan and Ying, 2023, Molinaro et al., 2023, Jiang et al., 2022, Chen et al., 2023], aiming to reconstruct the parameters in the PDE from solution data. Compared to the forward problem, the inverse problem is typically more challenging due to its ill-posed nature. To tackle the ill-posedness, many NOs incorporate prior information, in the form of governing PDEs [Yang et al., 2021, Li et al., 2021], regularizers [Dittmer et al., 2020, Obmann et al., 2020, Ding et al., 2022, Chen et al., 2023], or additional operator structures [Uhlmann, 2009, Cai et al., 2019, Yilmaz, 2001]. To our knowledge, our NO architecture is the first that solves both the forward (prediction) and inverse (discovery) problems simultaneously.

Attention mechanism. Since 2017, the attention mechanism has become the backbone of state-of-the-art deep learning models on many core AI tasks like NLP and CV. By calculating the similarity among tokens, the attention mechanism captures long-range dependencies between tokens [Vaswani et al., 2017]. Then, the tokens are spatially mixed to obtain the layer output. Based on the choice of mixers, attention-based models can be divided into three main categories: discrete graph-based attentions [Child et al., 2019, Ho et al., 2019, Wang et al., 2020, Katharopoulos et al., 2020], MLP-based attentions [Tolstikhin et al., 2021, Touvron et al., 2022, Liu et al., 2021], and convolution-based attentions [Lee-Thorp et al., 2021, Rao et al., 2021, Guibas et al., 2021, Nekoozadeh et al., 2023]. While most attention models focus on discrete mixers, it is proposed in Guibas et al. [2021], Nekoozadeh et al. [2023], Tsai et al. [2019], Cao [2021], Wei and Zhang [2023] to frame token mixing as a kernel integration, with the goal of obtaining predictions independent of the input resolution.

Along the line of PDE-solving tasks, various attention mechanisms have been used to enlarge model capacity. To improve the accuracy of forward PDE solvers, Cao [2021] removes the softmax normalization in the attention mechanism and employs linear attention as a learnable kernel in NOs. Further developments include the Galerkin-type linear attention in an encoder-decoder architecture in OFormer [Li et al., 2022], a hierarchical transformer for learning multiscale problems [Liu et al., 2022], and a heterogeneous normalized attention with a geometric gating mechanism [Hao et al., 2023] to handle multiple input features. In particular, going beyond solving a single PDE, the foundation model feature of attention mechanisms has been applied towards solving multiple types of PDEs within a specified context in Yang and Osher [2024], Ye et al. [2024], Sun et al. [2024], Zhang [2024]. However, none of the existing work discovers hidden physics from data, nor do they discuss the connections between the attention mechanism and the inverse PDE problem.

3 Nonlocal Attention Operator

Consider multiple physical systems that are described by a class of operators mapping from input functions $u \in \mathbb{X}$ to output functions $f \in \mathbb{Y}$. Our goal is to learn the common physical law, in the form of operators $\mathcal{L}_K : \mathbb{X} \to \mathbb{Y}$ with system-dependent kernels K:

$$\mathcal{L}_K[u] + \epsilon = f. \tag{1}$$

3

Here $\mathbb{X}$ and $\mathbb{Y}$ are Banach spaces, ϵ denotes an additive noise describing the discrepancy between the ground-truth operator and the optimal surrogate operator, and K is a kernel function representing the nonlocal spatial interaction. As such, the kernel provides the knowledge of its corresponding system, while (1) offers a zero-shot prediction model for new and unseen systems.

To formulate the learning, we consider n_{train} training datasets from different systems, with each dataset containing d_u function pairs (u, f):

$$\mathcal{D}_{\text{tr}} = \{\{(u_i^\eta(x), f_i^\eta(x))\}_{i=1}^{d_u}\}_{\eta=1}^{n_{train}}. \tag{2}$$

In practice, the data of the input and output functions are on a spatial mesh $\{x_k\}_{k=1}^{n_x} \subset \Omega \subset \mathbb{R}^{d_x}$. The n_{train} models with kernels $\{K^\eta\}_{\eta=1}^{n_{train}}$ correspond to different material micro-structures or different parametric settings. As a demonstration, we consider models for heterogeneous materials with operators in the form

$$\mathcal{L}_K[u](x) = \int_\Omega K(x,y)g[u](y)dy, \; x \in \Omega, \tag{3}$$

where $g[u](y)$ is a functional of u determined by the operator; for example $g[u](y) = u(y)$ in Section 5.3. Our approach extends naturally to other forms of operators, such as those with radial interaction kernels in Section 5.1 and heterogeneous interaction in the form of $\mathcal{L}_K[u](x) = \int_\Omega K(x,y)g[u](x,y)dy$. Additionally, for simplicity, we consider scalar-valued functions u and f and note that the extension to vector-valued functions is trivial.

Remark: Such an operator learning problem arises in many applications in forward and inverse PDE-solving problems. The inference of the kernel K is an inverse problem, and the learning of the nonlocal operator is a forward problem. When considering a single physical system and taking K in (3) as an input-independent kernel, classical NOs can be obtained for forward PDE-solving tasks [Li et al., 2020c, Gupta et al., 2021] and governing law learning tasks [You et al., 2021, Jafarzadeh et al., 2024]. Different from existing work, we consider the operator learning across multiple systems.

3.1 Kernel map with attention mechanism

The key ingredient in NAO is a kernel map constructed using the attention mechanism. It maps from data pairs to an estimation of the underlying kernel. The kernel map

$$\{(u_i, f_i)\}_{i=1}^{d_u} \to K[\mathbf{u}_{1:d}, \mathbf{f}_{1:d}; \theta] \tag{4}$$

has parameters θ estimated from the training dataset (2). As such, it maps from the token $(\mathbf{u}_{1:d}, \mathbf{f}_{1:d})$ of the dataset $\{(u_i, f_i)\}_{i=1}^{d_u}$ to a kernel estimator, acting as an inverse PDE solver.

A major innovation of this kernel map is its dependence on both u and f through their tokens. Thus, our approach distinguishes itself from the forward problem-solving NOs in the related work section, where the attention depends only on u.

We first transfer the data $\{(u_i, f_i)\}_{i=1}^{d_u}$ to tokens $(\mathbf{u}_{1:d}, \mathbf{f}_{1:d})$ according to the operator in (3) by

$$\begin{aligned}
\mathbf{u}_{1:d} &= (\mathbf{u}_1, \ldots, \mathbf{u}_d) = \big(g[u_j](y_k)\big)_{1 \le j \le d, 1 \le k \le N} \in \mathbb{R}^{N \times d}, \\
\mathbf{f}_{1:d} &= (f_j(x_k))_{1 \le j \le d, 1 \le k \le N} \in \mathbb{R}^{N \times d},
\end{aligned} \tag{5}$$

where $d = d_u$ and $N = n_x$, assuming that $g[u]$ has a spatial mesh $\{y_k = x_k\}_{k=1}^N$.

Then, our discrete $(L+1)$-layer attention model for the inverse PDE problem writes:

$$\begin{aligned}
\boldsymbol{X}_{\text{in}} &= \boldsymbol{X}^{(0)} = (\boldsymbol{U}^{(0)}, \boldsymbol{F}^{(0)}) := (\mathbf{u}_{1:d}; \mathbf{f}_{1:d}) \in \mathbb{R}^{2N \times d}, \\
\boldsymbol{X}^{(l)} &= \text{Attn}[\boldsymbol{X}^{(l-1)}; \theta_l]\boldsymbol{X}^{(l-1)} + \boldsymbol{X}^{(l-1)} =: (\boldsymbol{U}^{(l)}, \boldsymbol{F}^{(l)}) \in \mathbb{R}^{2N \times d}, \quad 1 \le l < L, \\
\boldsymbol{X}_{\text{out}} &= \boldsymbol{X}^L = K[\mathbf{u}_{1:d}, \mathbf{f}_{1:d}; \theta]\mathbf{u}_{1:d} \approx \mathbf{f}_{1:d} \in \mathbb{R}^{N \times d}.
\end{aligned} \tag{6}$$

Here, $\theta_l = (\boldsymbol{W}_l^Q \in \mathbb{R}^{d \times d_k}, \boldsymbol{W}_l^K \in \mathbb{R}^{d \times d_k})$ and the attention function is

$$\text{Attn}[\boldsymbol{X}; \theta_l] = \sigma\left(\frac{1}{\sqrt{d_k}}\boldsymbol{X}\boldsymbol{W}_l^Q{\boldsymbol{W}_l^K}^\top \boldsymbol{X}^\top\right) \in \mathbb{R}^{2N \times 2N}.$$

The kernel map is defined as

$$\begin{aligned}
K[\mathbf{u}_{1:d}, \mathbf{f}_{1:d}; \theta] = {}& W^{P,u}\sigma\left(\frac{1}{\sqrt{d_k}}(\boldsymbol{U}^{(L-1)})^\top \boldsymbol{W}_L^Q(\boldsymbol{W}_L^K)^\top \boldsymbol{U}^{(L-1)}\right) \\
& + W^{P,f}\sigma\left(\frac{1}{\sqrt{d_k}}(\boldsymbol{F}^{(L-1)})^\top \boldsymbol{W}_L^Q(\boldsymbol{W}_L^K)^\top \boldsymbol{U}^{(L-1)}\right),
\end{aligned} \tag{7}$$

where $\theta := \{W^{P,u} \in \mathbb{R}^{N \times N}, W^{P,f} \in \mathbb{R}^{N \times N}, \{\boldsymbol{W}_l^Q, \boldsymbol{W}_l^K\}_{l=1}^L\}$ are learnable parameters. Here, we note that d_k only controls the rank bound when lifting each point-wise feature via $\boldsymbol{W}_l^Q(\boldsymbol{W}_l^K)^\top$. In the following, we denote $\boldsymbol{W}_l^{QK} := \frac{1}{\sqrt{d_k}}\boldsymbol{W}_l^Q(\boldsymbol{W}_l^K)^\top$, which characterizes the (trainable) interaction of d input function pair instances.

3.2 Nonlocal Attention Operator in continuum limit

As suggested by Cao [2021], we take the activation function σ as a linear operator. Then, noting that the matrix multiplication in (6) is a Riemann sum approximation of an integral (with a full derivation in Appendix A), we propose the nonlocal attention operator as the continuum limit of (6):

$$g_j^{(0)}(x) := g[u_j](x), \; f_j^{(0)}(x) := f_j(x),$$

$$\begin{pmatrix} g_j^{(l)}(x) \\ f_j^{(l)}(x) \end{pmatrix} = \int_\Omega K^{(l)}(x,y) \begin{pmatrix} g_j^{(l-1)}(y) \\ f_j^{(l-1)}(y) \end{pmatrix} dy + \begin{pmatrix} g_j^{(l-1)}(x) \\ f_j^{(l-1)}(x) \end{pmatrix}, \quad 1 \le l < L,$$

$$\mathcal{L}_{K[\mathbf{u}_{1:d}, \mathbf{f}_{1:d}; \theta]}[u](x) = \int_\Omega K[\mathbf{u}_{1:d}, \mathbf{f}_{1:d}; \theta](x,y) g[u](y) dy, \tag{8}$$

in which the integration is approximated by the Riemann sum in our implementation. Here,

$$K^{(l)}(x,y) := \begin{bmatrix} \sum_{\omega,\nu=1}^d \left(g_\omega^{(l-1)}(x)\boldsymbol{W}_l^{QK}[\omega,\nu]g_\nu^{(l-1)}(y)\right) & \sum_{\omega,\nu=1}^d \left(g_\omega^{(l-1)}(x)\boldsymbol{W}_l^{QK}[\omega,\nu]f_\nu^{(l-1)}(y)\right) \\ \sum_{\omega,\nu=1}^d \left(f_\omega^{(l-1)}(x)\boldsymbol{W}_l^{QK}[\omega,\nu]g_\nu^{(l-1)}(y)\right) & \sum_{\omega,\nu=1}^d \left(f_\omega^{(l-1)}(x)\boldsymbol{W}_l^{QK}[\omega,\nu]f_\nu^{(l-1)}(y)\right) \end{bmatrix}$$

$$K[\mathbf{u}_{1:d}, \mathbf{f}_{1:d}; \theta](x,y) := \sum_{\omega,\nu=1}^d \int_\Omega W^{P,u}(x,z) \left(g_\omega^{(L-1)}(z)W_L^{QK}[\omega,\nu]g_\nu^{(L-1)}(y)\right) dz$$

$$+ \sum_{\omega,\nu=1}^d \int_\Omega W^{P,f}(x,z) \left(f_\omega^{(L-1)}(z)W_L^{QK}[\omega,\nu]g_\nu^{(L-1)}(y)\right) dz.$$

We learn the parameters θ by minimizing the following mean squared error loss function:

$$\mathcal{E}(\theta) := \frac{1}{n_{train}} \sum_{\eta=1}^{n_{train}} \sum_{i=1}^d \int_\Omega \left| \mathcal{L}_{K[\mathbf{u}_{1:d}^\eta, \mathbf{f}_{1:d}^\eta; \theta]}[u_i^\eta](x) - f_i^\eta(x) \right|^2 dx. \tag{9}$$

The performance of the model is evaluated on test tasks with new and unseen kernels $\mathcal{D}_{test} = \{(u_i^{test}(x), f_i^{test}(x))\}_{i=1}^d$ based on the following two criteria:

$$\text{Operator (forward PDE) Error: } E_f := \frac{1}{d} \sum_{i=1}^d \frac{\left\| \mathcal{L}_{K[\mathbf{u}_{1:d}^{test}, \mathbf{f}_{1:d}^{test}; \theta]}[u_i^{test}](x) - f_i^{test}(x) \right\|_{L^2(\Omega)}}{\|f_i^{test}(x)\|_{L^2(\Omega)}}, \tag{10}$$

$$\text{Kernel (inverse PDE) Error: } E_K := \left| K[\mathbf{u}_{1:d}^{test}, \mathbf{f}_{1:d}^{test}; \theta] - K_{test} \right|_{L^2(\rho)} / |K_{test}|_{L^2(\rho)}. \tag{11}$$

Here, $L^2(\rho)$ is the empirical measure in the kernel space as defined in Lu et al [2023]. Note that θ are trained using multiple datasets. Intuitively speaking, the attention mechanism helps encode the prior information from other tasks for learning the nonlocal kernel, and leads to estimators significantly better than those using a single dataset. To formally understand this mechanism, we analyze a shallow 2-layer NAO in the next section.

4 Understanding the attention mechanism

To facilitate further understanding of the attention mechanism, we analyze the limit of the two-layer attention-parameterized kernel in (7) and the range of the kernel map, which falls in the space of identifiability for the kernels from the training data. We also connect the kernel map with the regularized estimators. For simplicity, we consider operators of the form

$$\mathcal{L}_K[u](x) = \int_0^\delta K(r)g[u](r,x)dr, \; x \in \Omega, \tag{12}$$

which is the radial nonlocal kernel in Sec.5.1.

4.1 Limit of the two-layer attention-parameterized kernel

We show that, as the number N and the spatial mesh n_x approach infinity, the limit of the two-layer attention-parameterized kernel is a double integral. Its proof is in Appendix B.1.

For simplicity, we assume that the dataset in (2) has $d_u = 1$ and $n_{train} = 1$ with a uniform mesh $\{x_j\}_{j=1}^{n_x}$. We define the tokens by

$$\mathbf{u}_{1:d} = (\mathbf{u}_1, \ldots, \mathbf{u}_d) = (g[u](r_k, x_j))_{1 \le j \le d, 1 \le k \le N} \in \mathbb{R}^{N \times d}, \; \mathbf{f}_{1:d} = (f(x_j))_{1 \le j \le d} \in \mathbb{R}^{1 \times d},$$
(13)

where $d = n_x$ and $\{r_k\}_{k=1}^N$ is the spatial mesh for K's independent variable $r \in [0, \delta]$.

Lemma 4.1. *Consider the two-layer attention model in (6)–(7) with bounded parameters. For each d and N, let $\{x_j\}_{j=1}^d$ and $\{r_k\}_{k=1}^N$ be uniform meshes of the compact sets Ω and $[0, \delta]$, and let $\{A_j\}_{j=1}^d$ be the resulting uniform partition of Ω. Assume that $g[u]$ in (12) is continuous on $[0, \delta] \times \Omega$. Then,*

$$\lim_{N \to \infty} \lim_{d \to \infty} \sum_{k=1}^N K[\mathbf{u}_{1:d}, \mathbf{f}_{1:d}; \theta](r_k) \mathbf{1}_{[r_{k-1}, r_k)}(r)(r_k - r_{k-1})$$

$$= K[u, f](r) := \int_0^\delta W^{P,u}(|r'|) \sigma \left(\int \int \left[g[u](r', x) W^{QK}(x, y) g[u](r, y) dx dy \right] \right) dr' \quad (14)$$

$$+ W^{P,f} \sigma \left(\int \int \left[f(x) W^{QK}(x, y) g[u](r, y) \right] dx dy \right),$$

where $W^{QK}(x, y) = \lim_{d \to \infty} \sum_{j,j'=1}^d W^{QK}[j, j'] \mathbf{1}_{A_j \times A_{j'}}(x, y)$ is the scaled $L^2(\Omega \times \Omega)$ limit of the parameter matrix $W^{QK}[j', j] = \sum_{l=1}^{d_k} W^Q[j, l] \cdot W^K[j', l]$ and $W^{P,u}(r) = \lim_{N \to \infty} \sum_{k=1}^N W^{P,u}[k] \mathbf{1}_{[r_{k-1}, r_k)}(r)$.

4.2 Space of identifiability for the kernels

For a given training dataset, we show that the function space in which the kernels can be identified is the closure of a data-adaptive reproducing kernel Hilbert space (RKHS). This space contains the range of the kernel map and hence provides the ground for analyzing the inverse problem.

Lemma 4.2 (Space of Identifiability). *Assume that the training data pairs in (2) are sampled from continuous functions $\{u_i^\eta\}_{i,\eta=1}^{d_u, n_{train}}$ with a compact support. Then, the function space the loss function in (9) has a unique minimizer $K(s) = K[\mathbf{u}_{1:d}^\eta, \mathbf{f}_{1:d}^\eta; \theta](s)$ is the closure of a data-adaptive RKHS H_G with a reproducing kernel $\bar{G}$ determined by the training data:*

$$\bar{G}(r, s) = [\rho'(r) \rho'(s)]^{-1} G(r, s),$$

where ρ' is the density of the empirical measure ρ defined by

$$\rho'(r) := \frac{1}{Z} \sum_{\eta=1}^{n_{train}} \sum_{i=1}^{d_u} \int_\Omega |g[u_i^\eta](r, x)| dx, \quad (15)$$

and the function G is defined by $G(r, s) := \frac{1}{n_{train} d} \sum_{\eta=1}^{n_{train}} \sum_{i=1}^{d_u} \int_\Omega g[u_i^\eta](r, x) g[u_i^\eta](s, x) dx$.

The above space is data-adaptive since the integral kernel $\bar{G}$ depends on data. It characterizes the information in the training data for estimating the nonlocal kernel $K(s) = K[\mathbf{u}_{1:d}^\eta, \mathbf{f}_{1:d}^\eta; \theta](s)$. In general, the more data, the larger the space is. On the other hand, note that the loss function's minimizer with respect to $K(s)$ is not the kernel map. The minimizer is a fixed estimator for the training dataset and does not provide any information for estimating the kernel from another dataset.

Comparison with regularized estimators. The kernel map solves the ill-posed inverse problem using prior information from the training dataset of multiple systems, which is not used in classical inverse problem solvers. To illustrate this mechanism, consider the extreme case of estimating the kernel in the nonlocal operator from a dataset consisting of only a single function pair (u, f). This inverse problem is severely ill-posed because of the small dataset and the need for deconvolution to estimate the kernel. Thus, regularization is necessary, where two main challenges present: (i) the selection of a proper regularization with limited prior information, and (ii) the prohibitive computational cost of solving the resulting large linear systems many times.

In contrast, our kernel map $K[\mathbf{u}_{1:d}, \mathbf{f}_{1:d}; \theta](s)$, with the parameter θ estimated from the training datasets, acts on the token $(\mathbf{u}_{1:d}, \mathbf{f}_{1:d})$ of (u, f) to provide an estimator. It passes the prior information about the kernel from the training dataset to the estimation for new datasets. Importantly,

it captures the nonlinear dependence of the estimator on the data (u, f). Computationally, it can be applied directly to multiple new datasets without solving the linear systems. In Section B.2, we further show that a regularized estimator depends nonlinearly on the data pair (u, f). In particular, similar to Lemma 4.1, there is an RKHS determined by the data pair (u, f). The regularized estimator suggests that the kernel map can involve a component quadratic in the feature $g[u]$, similar to the limit form of the attention model in Lemma 4.1.

5 Experiments

We assess the performance of NAO on a wide range of physics modeling and discovery datasets. Our evaluation focuses on several key aspects: 1) we demonstrate the merits of the continuous and linear attention mechanism, compare the performance with the baseline discrete attention model (denoted as Discrete-NAO), the softmax attention mechanism (denoted as Softmax-NAO), NAO with input on u only (denoted as NAO-u), the convolution-based attention mechanism (denoted as AFNO [Guibas et al., 2021]), and an MLP-based encoder architecture that maps the datum $[\boldsymbol{u}_{1:d}, \boldsymbol{f}_{1:d}]$ directly to a latent kernel (denoted as Autoencoder); 2) we measure the generalizability, in particular, the zero-shot prediction performance in modeling a new physical system with unseen governing equations, and across different resolutions; 3) we evaluate the data efficiency-accuracy trade-off in ill-posed inverse PDE learning tasks, as well as the interpretability of the learned kernels. In all experiments, the optimization is performed with the Adam optimizer. To conduct fair comparison across different methods, we tune the hyperparameters, including the learning rates, the decay rates, and the regularization parameters, to minimize the training loss. In all examples, we use 3-layer models, and parameterize the kernel network $W^{P,u}$ and $W^{P,f}$ with a 3-layer MLP with hidden dimensions $(32, 64)$ and LeakyReLU activation. Experiments are conducted on a single NVIDIA GeForce RTX 3090 GPU with 24 GB memory. Additional results and details on data generation and training strategies are provided in Appendix C.

5.1 Radial kernel learning

Table 1: Test errors and the number of trainable parameters for the radial kernel problem, where bold numbers highlight the best methods. The small operator errors and large kernel errors of discrete-NAO highlight the ill-posedness of the inverse problem. NAO overcomes the ill-posedness and yields resolution-invariant estimators.

Case	model	#param	Operator test error		Kernel test error	
			ID	OOD1	ID	OOD1
$d = 302, d_k{=}10$	Discrete-NAO	16526	**1.33%**	25.81%	29.02%	28.80%
	Softmax-NAO	18843	13.45%	66.06%	67.55%	85.80%
	AFNO	19605	22.62%	68.76%	-	-
	NAO	18843	1.48%	**8.10%**	**5.40%**	**10.02%**
	NAO-u	18842	13.68%	66.68%	20.46%	74.03%
	Autoencoder	16424	12.97%	1041.49%	22.56%	136.79%
$d = 302, d_k{=}5$	Discrete-NAO	10465	**1.63%**	15.80%	33.21%	30.39%
	NAO	12783	2.34%	**9.23%**	**6.87%**	**14.62%**
$d = 302, d_k{=}20$	Discrete-NAO	28645	1.35%	18.70%	35.49%	30.81%
	NAO	30963	**1.33%**	**9.12%**	**4.63%**	**9.14%**
$d = 100, d_k{=}10$	Discrete-NAO	8446	1.73%	14.92%	34.52%	35.20%
	NAO	10763	**1.07%**	**6.35%**	**7.41%**	**17.02%**
$d = 50, d_k{=}10$	Discrete-NAO	6446	2.29%	10.31%	41.80%	45.30%
	NAO	8763	**1.56%**	**7.19%**	**15.95%**	**29.47%**
$d = 30, d_k{=}10$	Discrete-NAO	5646	5.60%	11.31%	58.24%	64.23%
	NAO	7963	**2.94%**	**8.04%**	**22.65%**	**33.77%**

In this example, we consider the learning of nonlocal diffusion operators, in the form:

$$\mathcal{L}_{\gamma_\eta}[u](x) = \int_\Omega \gamma_\eta(|y - x|)[u(y) - u(x)]dy = f(x), \forall x \in \Omega. \tag{16}$$

Unlike a (local) differential operator, this operator depends on the function u nonlocally through the convolution of $u(y) - u(x)$, and the operator is characterized by a radial kernel γ_η. It finds broad physical applications in describing fracture mechanics [Silling, 2000], anomalous diffusion behaviors [Bucur et al., 2016], and the homogenization of multiscale systems [Du et al., 2020].
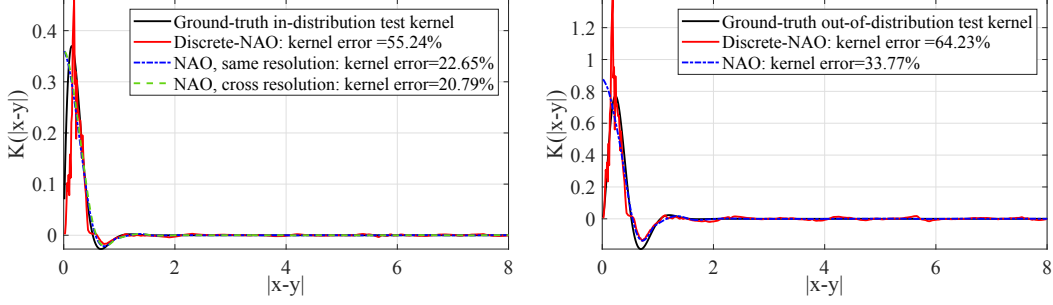
7

Figure 2: Results on radial kernel learning, when learning the test kernel from a small ($d = 30$) number of data pairs: test on an ID task (left), and test on an OOD task (right).

In this context, our goal is to learn the operator $\mathcal{L}$ as well as to discover the hidden mechanism, namely the kernel $K[\boldsymbol{u}_{1:d}, \boldsymbol{f}_{1:d}; \theta](x, y) = \gamma_\eta(|y - x|)$. In the form of the operator in (12), we have $K(r) = \gamma_\eta(r)$ and $g[u](r, x) = u(x + r) + u(x - r) - 2u(x)$ for $r \in [0, \delta]$.

To generate the training data, we consider 7 sine-type kernels

$$\gamma_\eta(|y - x|) := \exp(-\eta(|y - x|)) \sin(6|y - x|)\mathbf{1}_{[0,11]}(|y - x|), \ \eta = 1, 2, 3, 4, 6, 7, 8. \quad (17)$$

Here, η denotes task index. We generate 4530 data pairs $(g^\eta[u], f^\eta)$ with a fixed resolution $\Delta x = 0.0125$ for each task, where the loading function $\mathcal{L}_{\gamma_\eta}[u^\eta] = f^\eta$ is computed by the adaptive Gauss-Kronrod quadrature method. Then, we form a training sample of each task by taking d pairs from this task. When taking the token size $d = 302$, each task contains $\frac{4530}{302} = 15$ samples. We consider two test kernels: one following the same rule of (17) with $\eta = 5$ (denoted as the "in-distribution (ID) test" system), and the other following a different rule (denoted as the "out-of-distribution (OOD) test1" system):

$$\gamma_{ood1}(|y - x|) := |y - x|(11 - |y - x|) \exp(-5(|y - x|)) \sin(6|y - x|)\mathbf{1}_{[0,11]}(|y - x|). \quad (18)$$

Both the operator error (10) and the kernel error (11) are provided in Table 1. While the former measures the error of the learned forward PDE solver (i.e., learning a physical model), the latter demonstrates the capability of serving as an inverse PDE solver (i.e., physics discovery).

Ablation study. We first perform an ablation study on NAO, by comparing its performance with its variants (Discrete-NAO, Softmax-NAO, and NAO-u), AFNO, and Autoencoder, with a fixed token dimension $d = 302$, query-key feature size $d_k = 10$, and data resolution $\Delta x = 0.0125$. When comparing the operator errors, both Discrete-NAO and NAO serve as good surrogate models for the ID task with relative errors of 1.33% and 1.48%, respectively, while the other three baselines show $> 10\%$ errors. Therefore, we focus more on the comparison between Discrete-NAO and NAO. This gap becomes more pronounced in the OOD task: only NAO is able to provide a surrogate of $\mathcal{L}_{\gamma_{ood}}$ with $< 10\%$ error, who outperforms its discrete mixer counterpart by 68.62%, indicating that NAO learns a more generalizable mechanism. This argument is further affirmed when comparing the kernel errors, where NAO substantially outperforms all baselines by at least 81.39% in the ID test and 65.21% in the OOD test. This study verifies our analysis in Section 4: NAO learns the kernel map in the space of identifiability, and hence possesses advantages in solving the challenging ill-posed inverse problem. Additionally, we vary the query-key feature size from $d_k = 10$ to $d_k = 5$ and $d_k = 20$. Note that d_k determines the rank bound of W^{QK}, the matrix that characterizes the interaction between different data pairs. Discrete-NAO again performs well only in approximating the operator for the ID test, while NAO achieves consistent results in both tests and criteria, showing that it has successfully discovered the intrinsic low-dimension in the kernel space.

Alleviating ill-posedness. To further understand NAO's capability as an inverse PDE solver, we reduce the number of data pairs for each sample from $d = 302$ to $d = 30$, making it more ill-posed as an inverse PDE problem. NAO again outperforms its discrete mixer counterpart in all aspects. Interestingly, the errors in NAO increase almost monotonically, showing its robustness. For Discrete-NAO, the error also increases monotonically in the ID operator test, but there exists no consistent pattern in other test criteria. Figure 2 shows the learned test kernels in both the ID and OOD tasks. It shows that Discrete-NAO learns highly oscillatory kernels, while our continuous NAO only has a discrepancy near $|x - y| = 0$. Note that when $|x - y| = 0$, we have $u(y) - u(x) = 0$ in the ground-truth operator (16), and hence the kernel value at this point does not change the operator
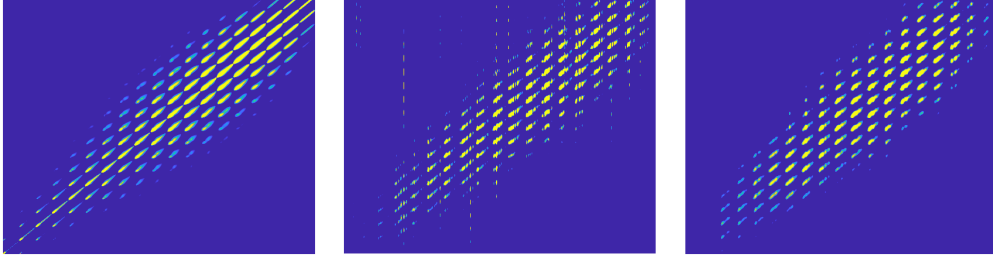
8

Figure 3: Kernel visualization in experiment 2, where the kernels correspond to the inverse of stiffness matrix: ground truth (left), test kernel from Discrete-NAO (middle), kernel from NAO (right).

value. That means, our data provides almost no information at this point. This again verifies our analysis: continuous NAO learns the kernel map structure from small data based on prior knowledge from other task datasets.

Cross-resolution. We test the NAO model trained with $\Delta x = 0.0125$ on a dataset corresponding to $\Delta x = 0.025$, and plot the results in Figure 2 Left. The predicted kernel is very similar to the one learned from the same resolution, and the error is also on-par (22.65% versus 20.79%).

5.2 Solution operator learning

We consider the modeling of 2D sub-surface flows through a porous medium with a heterogeneous permeability field. Following the settings in Li et al. [2020a], the high-fidelity synthetic simulation data for this example are described by the Darcy flow. Here, the physical domain is $\Omega = [0, 1]^2$, $b(\boldsymbol{x})$ is the permeability field, and the Darcy's equation has the form:

$$-\nabla \cdot (b(\boldsymbol{x})\nabla p(\boldsymbol{x})) = g(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega; \qquad p(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \partial\Omega. \tag{19}$$

In this context, we aim to learn the solution operator of Darcy's equation and compute the pressure field $p(\boldsymbol{x})$. We consider two study scenarios. 1) $g \to p$: each task has a fixed microstructure $b(\boldsymbol{x})$, and our goal is to learn the (linear) solution operator mapping from each loading field g to the corresponding solution field p. In this case, the kernel K acts as the Green's function of (19), and can be approximated by the inverse of the stiffness matrix. 2) $b \to p$: each task has a fixed loading field $g(\boldsymbol{x})$, and our goal is to learn the (nonlinear) solution operator mapping from the permeability field b to the corresponding solution field p.

Table 2: Test errors and the number of trainable parameters in solution operator learning.

Case	model	#param	Linear Operator: $g \to p$	Nonlinear Operator: $b \to p$
$d = 20$, d_k=20	Discrete-NAO	161991	8.61%	**10.84%**
900 samples	NAO	89778	**8.33%**	11.40%
$d = 50$, d_k=40	Discrete-NAO	662163	3.28%	5.61%
9000 samples	NAO	189234	**3.19%**	**5.28%**

We report the operator learning results in Table 2, where NAO slightly outperforms Discrete-NAO in most cases, using only 1/2 or 1/3 the number of trainable parameters. On the other hand, we also verify the kernel learning results by comparing the learned kernels in a test case with the ground-truth inverse of stiffness matrix in Figure 3. Although both Discrete-NAO and NAO capture the major pattern, the kernel from Discrete-NAO again shows a spurious oscillation in regions where the ground-truth kernel has zero value. On the other hand, by exploring the kernel map in the integrated knowledge space, the learned kernel from NAO does not exhibit such spurious modes.

To demonstrate the physical interpretability of the learned kernel, in the first row of Figure 4 we show the ground-truth microstructure $b(\boldsymbol{x})$, a test loading field instance $g(\boldsymbol{x})$, and the corresponding solution $p(\boldsymbol{x})$. By taking the summation of the kernel strength on each row, one can discover the interaction strength of each material point x with its neighbors. As this strength is related to the permeability field $b(x)$, the underlying microstructure can be recovered accordingly. In the bottom row of Figure 4, we demonstrate the discovered microstructure of this test task. We note that the discovered microstructure is smoothed out due to the continuous setting of our learned kernel (as shown in the bottom left plot), and a thresholding step is performed to discover the two-phase microstructure. The discovered microstructure (bottom right plot) matches well with the hidden ground-truth microstructure (left plot), except for regions near the domain boundary. This mismatch is due to
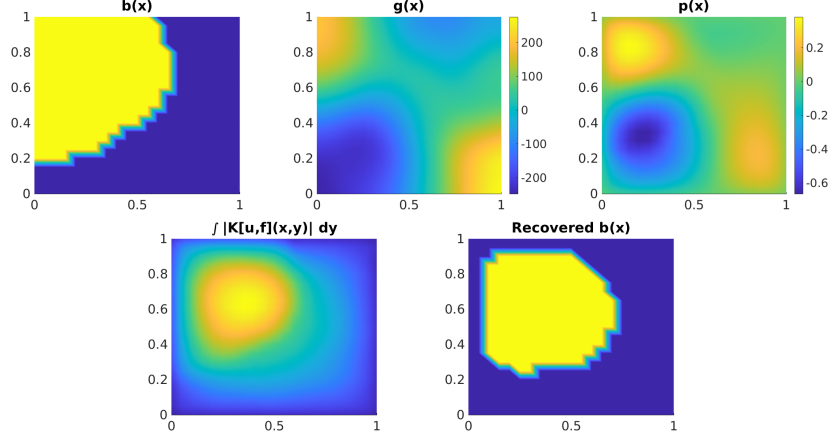
Figure 4: Demonstration of the generated data and the recovered microstructure from the learned kernel in Example 2. Top row: the ground-truth two-phase material microstructure from a test task (left), an exemplar loading field instance (middle), and the corresponding solution field instance (right). Bottom row: summation of the learned kernel for each line, corresponding to the total interaction of all material points (left), and the discovered two-phase material microstructure after thresholding (right).

the applied Dirichlet-type boundary condition ($p(x) = 0$ on $\partial\Omega$) in all samples, which leads to the measurement pairs $(p(x), g(x))$ containing no information near the domain boundary $\partial\Omega$ and makes it impossible to identify the kernel on boundaries.

5.3 Heterogeneous material learning

In this example, we investigate the learning of heterogeneous and nonlinear material responses using the Mechanical MNIST benchmark [Lejeune, 2020]. For training and testing, we take 500 heterogeneous material specimens, where each specimen is governed by a Neo-Hookean material with a varying modulus converted from the MNIST bitmap images. On each specimen, 200 loading/response data pairs are provided. Two generalization scenarios are considered. 1) We mix the data from all numbers and randomly take 10% of specimens for testing. This scenario corresponds to an ID test. 2) We leave all specimens corresponding to the number '9' for testing, and use the rest for training. This scenario corresponds to an OOD test. The corresponding results are listed in Table 3, where NAO again outperforms its discrete counterpart.

Table 3: Test errors and the number of trainable parameters in heterogeneous material learning.

Case	model	#param	ID test	OOD test
$d = 40$, d_k=40	Discrete-NAO	5,469,528	7.21%	7.95%
22500 samples	NAO	142,534	**6.57%**	**6.26%**
$d = 100$, d_k=100	Discrete-NAO	7,353,768	6.34%	6.01%
45000 samples	NAO	303,814	**4.75%**	**5.58%**

6 Conclusion

We propose Nonlocal Attention Operator (NAO), a novel NO architecture to simultaneously learn both the forward (modeling) and inverse (discovery) solvers in physical systems from data. In particular, NAO learns the function-to-function mapping based on an integral NO architecture and provides a surrogate forward solution predictor. In the meantime, the attention mechanism is crafted in building a kernel map from input-output function pairs to the system's function parameters, offering zero-shot generalizability to new and unseen physical systems. As such, the kernel map explores in the function space of identifiability, resolving the enduring ill-posedness in inverse PDE problems. In our empirical demonstrations, NAO outperforms all selected baselines on multiple datasets of inverse PDE problems and out-of-distribution generalizability tasks.

Broader Impacts: Beyond its merits in forward/inverse PDE modeling, our work represents an initial exploration in understanding the attention mechanism in physics modeling, and paves a theoretical path towards building a foundation model in scientific ML.

Limitations: Due to limited computational resource, our experiments focus on learning from a small to medium number (< 500) of similar physical systems. It would be beneficial to expand the coverage and enable learning across different types of physical systems.

Acknowledgments and Disclosure of Funding

References

Gilles Besnard, François Hild, and Stéphane Roux. finite-element displacement fields analysis from digital images: application to portevin–le châtelier bands. *Experimental mechanics*, 46(6):789–803, 2006.

Claudia Bucur, Enrico Valdinoci, et al. *Nonlocal diffusion and applications*, volume 20. Springer, 2016.

Shengze Cai, Zhiping Mao, Zhicheng Wang, Minglang Yin, and George Em Karniadakis. Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mechanica Sinica*, pages 1–12, 2022.

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

Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4):045002, 2019.

Ke Chen, Chunmei Wang, and Haizhao Yang. let data talk: data-regularized operator learning theory for inverse problems. *arXiv preprint arXiv:2310.09854*, 2023.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*, 2019.

Genevieve Coorey, Gemma A Figtree, David F Fletcher, Victoria J Snelson, Stephen Thomas Vernon, David Winlaw, Stuart M Grieve, Alistair McEwan, Jean Yee Hwa Yang, Pierre Qian, Kieran O'Brien, Jessica Orchard, Jinman Kim, Sanjay Patel, and Julie Redfern. The health digital twin to tackle cardiovascular diseaseA review of an emerging interdisciplinary field. *NPJ Digital Medicine*, 5:126, 2022.

Wen Ding, Kui Ren, and Lu Zhang. Coupling deep learning with full waveform inversion. *arXiv preprint arXiv:2203.01799*, 2022.

Sören Dittmer, Tobias Kluth, Peter Maass, and Daniel Otero Baguer. Regularization by architecture: A deep prior approach for inverse problems. *Journal of Mathematical Imaging and Vision*, 62:456–470, 2020.

Qiang Du, Bjorn Engquist, and Xiaochuan Tian. Multiscale modeling, homogenization and nonlocal effects: Mathematical and computational issues. *Contemporary mathematics*, 754:115–140, 2020.

Yuwei Fan and Lexing Ying. Solving traveltime tomography with deep learning. *Communications in Mathematics and Statistics*, 11(1):3–19, 2023.

Alberto Ferrari and Karen Willcox. Digital twins in mechanical and aerospace engineering. *Nature Computational Science*, 4(3):178–183, 2024.

J Ghaboussi, JH Garrett Jr, and Xiping Wu. Knowledge-based modeling of material behavior with neural networks. *Journal of engineering mechanics*, 117(1):132–153, 1991.

Jamshid Ghaboussi, David A Pecknold, Mingfu Zhang, and Rami M Haj-Ali. Autoprogressive training of neural network constitutive models. *International Journal for Numerical Methods in Engineering*, 42(1):105–126, 1998.

Somdatta Goswami, Aniruddha Bora, Yue Yu, and George Em Karniadakis. Physics-informed neural operators. *2022 arXiv preprint arXiv:2207.05748*, 2022.

John Guibas, Morteza Mardani, Zongyi Li, Andrew Tao, Anima Anandkumar, and Bryan Catanzaro. Adaptive fourier neural operators: Efficient token mixers for transformers. *arXiv preprint arXiv:2111.13587*, 2021.

Gaurav Gupta, Xiongye Xiao, and Paul Bogdan. Multiwavelet-based operator learning for differential equations. *Advances in neural information processing systems*, 34:24048–24062, 2021.

Zhongkai Hao, Zhengyi Wang, Hang Su, Chengyang Ying, Yinpeng Dong, Songming Liu, Ze Cheng, Jian Song, and Jun Zhu. Gnot: A general neural operator transformer for operator learning. In *International Conference on Machine Learning*, pages 12556–12569. PMLR, 2023.

Qizhi He, Devin W Laurence, Chung-Hao Lee, and Jiun-Shyan Chen. Manifold learning based data-driven modeling for soft biological tissues. *Journal of Biomechanics*, 117:110124, 2021.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

Siavash Jafarzadeh, Stewart Silling, Ning Liu, Zhongqiang Zhang, and Yue Yu. Peridynamic neural operators: A data-driven nonlocal constitutive model for complex material responses. *Computer Methods in Applied Mechanics and Engineering*, 425:116914, 2024.

Hanyang Jiang, Yuehaw Khoo, and Haizhao Yang. Reinforced inverse scattering. *arXiv preprint arXiv:2206.04186*, 2022.

George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020.

Ru-Yu Lai, Qin Li, and Gunther Uhlmann. Inverse problems for the stationary transport equation in the diffusion scaling. *SIAM Journal on Applied Mathematics*, 79(6):2340–2358, 2019.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms. *arXiv preprint arXiv:2105.03824*, 2021.

Emma Lejeune. Mechanical mnist: A benchmark dataset for mechanical metamodels. *Extreme Mechanics Letters*, 36:100659, 2020.

Zijie Li, Kazem Meidani, and Amir Barati Farimani. Transformer for partial differential equations' operator learning. *arXiv preprint arXiv:2205.13671*, 2022.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Graph kernel network for partial differential equations. *arXiv preprint arXiv:2003.03485*, 2020a.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Andrew Stuart, Kaushik Bhattacharya, and Anima Anandkumar. Multipole graph neural operator for parametric partial differential equations. *Advances in Neural Information Processing Systems*, 33:NeurIPS 2020, 2020b.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier NeuralOperator for Parametric Partial Differential Equations. In *International Conference on Learning Representations*, 2020c.

Zongyi Li, Hongkai Zheng, Nikola Kovachki, David Jin, Haoxuan Chen, Burigede Liu, Kamyar Azizzadenesheli, and Anima Anandkumar. Physics-informed neural operator for learning partial differential equations. *2021 arXiv preprint arXiv:2111.03794*, 2021.

Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. Pay attention to mlps. *Advances in neural information processing systems*, 34:9204–9215, 2021.

Ning Liu, Yue Yu, Huaiqian You, and Neeraj Tatikola. Ino: Invariant neural operators for learning complex physical systems with momentum conservation. In *International Conference on Artificial Intelligence and Statistics*, pages 6822–6838. PMLR, 2023.

Ning Liu, Yiming Fan, Xianyi Zeng, Milan Klöwer, LU ZHANG, and Yue Yu. Harnessing the power of neural operators with automatically encoded conservation laws. In *Forty-first International Conference on Machine Learning*, 2024a.

Ning Liu, Siavash Jafarzadeh, and Yue Yu. Domain agnostic fourier neural operators. *Advances in Neural Information Processing Systems*, 36, 2024b.

Ning Liu, Xuxiao Li, Manoj R Rajanna, Edward W Reutzel, Brady Sawyer, Prahalada Rao, Jim Lua, Nam Phan, and Yue Yu. Deep neural operator enabled digital twin modeling for additive manufacturing. *arXiv preprint arXiv:2405.09572*, 2024c.

Xinliang Liu, Bo Xu, and Lei Zhang. Ht-net: Hierarchical transformer based operator learning model for multiscale pdes. 2022.

Fei Lu, Quanjun Lang, and Qingci An. Data adaptive rkhs tikhonov regularization for learning kernels in operators. In *Mathematical and Scientific Machine Learning*, pages 158–172. PMLR, 2022.

Fei Lu, Qingci An, and Yue Yu. Nonparametric learning of kernels in nonlocal operators. *Journal of Peridynamics and Nonlocal Modeling*, pages 1–24, 2023.

Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *arXiv preprint arXiv:1910.03193*, 2019.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021.

Roberto Molinaro, Yunan Yang, Björn Engquist, and Siddhartha Mishra. Neural inverse operators for solving pde inverse problems. *arXiv preprint arXiv:2301.11167*, 2023.

Christoph Molnar. *Interpretable machine learning*. 2020.

Anahita Nekoozadeh, Mohammad Reza Ahmadzadeh, and Zahra Mardani. Multiscale attention via wavelet neural operators for vision transformers. *arXiv preprint arXiv:2303.12398*, 2023.

Daniel Obmann, Johannes Schwab, and Markus Haltmeier. Deep synthesis regularization of inverse problems. *arXiv preprint arXiv:2002.00155*, 2020.

Yong Zheng Ong, Zuowei Shen, and Haizhao Yang. IAE-Net: integral autoencoders for discretization-invariant learning. *arXiv preprint arXiv:2203.05142*, 2022.

David Pfau, James S Spencer, Alexander GDG Matthews, and W Matthew C Foulkes. Ab initio solution of the many-electron schrödinger equation with deep neural networks. *Physical Review Research*, 2(3):033429, 2020.

Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification. *Advances in neural information processing systems*, 34:980–993, 2021.

Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistic Surveys*, 16:1–85, 2022.

Stewart A Silling. Reformulation of elasticity theory for discontinuities and long-range forces. *Journal of the Mechanics and Physics of Solids*, 48(1):175–209, 2000.

Jingmin Sun, Yuxuan Liu, Zecheng Zhang, and Hayden Schaeffer. Towards a foundation model for partial differential equation: Multi-operator learning and extrapolation. *arXiv preprint arXiv:2404.12355*, 2024.

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021.

Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4):5314–5321, 2022.

Yao-Hung Hubert Tsai, Shaojie Bai, Makoto Yamada, Louis-Philippe Morency, and Ruslan Salakhutdinov. Transformer dissection: a unified understanding of transformer's attention via the lens of kernel. *arXiv preprint arXiv:1908.11775*, 2019.

Gunther Uhlmann. Electrical impedance tomography and calderón's problem. *Inverse problems*, 25 (12):123011, 2009.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.

Min Wei and Xuesong Zhang. Super-resolution neural operator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18247–18256, 2023.

Zhi-Qin John Xu, Yaoyu Zhang, Tao Luo, Yanyang Xiao, and Zheng Ma. Frequency principle: Fourier analysis sheds light on deep neural networks. *arXiv preprint arXiv:1901.06523*, 2019.

Liu Yang and Stanley J Osher. Pde generalization of in-context operator networks: A study on 1d scalar nonlinear conservation laws. *arXiv preprint arXiv:2401.07364*, 2024.

Liu Yang, Xuhui Meng, and George Em Karniadakis. B-pinns: Bayesian physics-informed neural networks for forward and inverse pde problems with noisy data. *Journal of Computational Physics*, 425:109913, 2021.

Zhanhong Ye, Xiang Huang, Leheng Chen, Hongsheng Liu, Zidong Wang, and Bin Dong. Pdeformer: Towards a foundation model for one-dimensional partial differential equations. *arXiv preprint arXiv:2402.12652*, 2024.

Özdoğan Yilmaz. *Seismic data analysis*, volume 1. Society of exploration geophysicists Tulsa, 2001.

Huaiqian You, Yue Yu, Nathaniel Trask, Mamikon Gulian, and Marta DElia. Data-driven learning of nonlocal physics from high-fidelity synthetic data. *Computer Methods in Applied Mechanics and Engineering*, 374:113553, 2021.

Huaiqian You, Yue Yu, Marta D'Elia, Tian Gao, and Stewart Silling. Nonlocal kernel network (NKN): A stable and resolution-independent deep neural network. *Journal of Computational Physics*, page arXiv preprint arXiv:2201.02217, 2022.

Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and E Weinan. Deep potential molecular dynamics: a scalable model with the accuracy of quantum mechanics. *Physical Review Letters*, 120(14):143001, 2018.

Zecheng Zhang. Modno: Multi operator learning with distributed neural operators. *arXiv preprint arXiv:2404.02892*, 2024.

A   Riemann sum approximation derivations

In this section, we show that the discrete attention operator (6) can be seen as the Riemann sum approximation of the nonlocal attention operator in (8), in the continuous limit. Without loss of generality, we consider a uniform discretization with grid size Δx. Denoting the $l-$th layer output as $\boldsymbol{U}^{(l)} = \left(g_j^{(l)}(x_k) \right)_{1 \leq j \leq d, 1 \leq k \leq N}$, $\boldsymbol{F}^{(l)} = \left(f_j^{(l)}(x_k) \right)_{1 \leq j \leq d, 1 \leq k \leq N}$, and $\frac{1}{\sqrt{d_k}} \boldsymbol{W}_l^Q \boldsymbol{W}_l^K$ as $\boldsymbol{W}_l^{QK} \in \mathbb{R}^{d \times d}$, for the $l-$th ($l < L$) layer update in (6) writes:

$$\boldsymbol{X}^{(l)}[\alpha, \beta] = \sum_{\gamma=1}^{2N} \sum_{\omega,\nu=1}^{d} \sum_{\lambda=1}^{d_k} \left(\frac{1}{\sqrt{d_k}} \boldsymbol{X}^{(l-1)}[\alpha, \omega] \boldsymbol{W}_l^Q[\omega, \lambda] \boldsymbol{W}_l^K[\nu, \lambda] \boldsymbol{X}^{(l-1)}[\gamma, \nu] \right) \boldsymbol{X}^{(l-1)}[\gamma, \beta] + \boldsymbol{X}^{(l-1)}[\alpha, \beta]$$

$$= \sum_{\gamma=1}^{2N} \sum_{\omega,\nu=1}^{d} \left(\boldsymbol{X}^{(l-1)}[\alpha, \omega] \boldsymbol{W}_l^{QK}[\omega, \nu] \boldsymbol{X}^{(l-1)}[\gamma, \nu] \right) \boldsymbol{X}^{(l-1)}[\gamma, \beta] + \boldsymbol{X}^{(l-1)}[\alpha, \beta].$$

That means, d_k only controls the rank bound when lifting each point-wise feature via W_l^{QK}, while W_l^{QK} characterizes the (trainable) interaction of d input function pair instances. Moreover, when $\alpha \leq N$, $\boldsymbol{X}^{(l)}[\alpha, \beta] = g_\beta^{(l)}(x_\alpha)$. When $\alpha > N$, $\boldsymbol{X}^{(l)}[\alpha, \beta] = f_\beta^{(l)}(x_{\alpha-N})$. Then,

$$g_\beta^{(l)}(x_\alpha) = \sum_{\gamma=1}^{N} \sum_{\omega,\nu=1}^{d} \left(\boldsymbol{X}^{(l-1)}[\alpha, \omega] \boldsymbol{W}_l^{QK}[\omega, \nu] \boldsymbol{X}^{(l-1)}[\gamma, \nu] \right) \boldsymbol{X}^{(l-1)}[\gamma, \beta]$$

$$+ \sum_{\gamma=N+1}^{2N} \sum_{\omega,\nu=1}^{d} \left(\boldsymbol{X}^{(l-1)}[\alpha, \omega] \boldsymbol{W}_l^{QK}[\omega, \nu] \boldsymbol{X}^{(l-1)}[\gamma, \nu] \right) \boldsymbol{X}^{(l-1)}[\gamma, \beta] + g_\beta^{(l-1)}(x_\alpha)$$

$$= \sum_{\gamma=1}^{N} \sum_{\omega,\nu=1}^{d} \left(g_\omega^{(l-1)}(x_\alpha) \boldsymbol{W}_l^{QK}[\omega, \nu] g_\nu^{(l-1)}(x_\gamma) \right) g_\beta^{(l-1)}(x_\gamma)$$

$$+ \sum_{\tilde{\gamma}=1}^{N} \sum_{\omega,\nu=1}^{d} \left(g_\omega^{(l-1)}(x_\alpha) \boldsymbol{W}_l^{QK}[\omega, \nu] f_\nu^{(l-1)}(x_{\tilde{\gamma}}) \right) f_\beta^{(l-1)}(x_{\tilde{\gamma}}) + g_\beta^{(l-1)}(x_\alpha).$$

Similarly,

$$f_\beta^{(l)}(x_\alpha) = \sum_{\gamma=1}^{N} \sum_{\omega,\nu=1}^{d} \left(\boldsymbol{X}^{(l-1)}[\alpha, \omega] \boldsymbol{W}_l^{QK}[\omega, \nu] \boldsymbol{X}^{(l-1)}[\gamma, \nu] \right) \boldsymbol{X}^{(l-1)}[\gamma, \beta]$$

$$+ \sum_{\gamma=N+1}^{2N} \sum_{\omega,\nu=1}^{d} \left(\boldsymbol{X}^{(l-1)}[\alpha, \omega] \boldsymbol{W}_l^{QK}[\omega, \nu] \boldsymbol{X}^{(l-1)}[\gamma, \nu] \right) \boldsymbol{X}^{(l-1)}[\gamma, \beta] + f_\beta^{(l-1)}(x_\alpha)$$

$$= \sum_{\gamma=1}^{N} \sum_{\omega,\nu=1}^{d} \left(f_\omega^{(l-1)}(x_\alpha) \boldsymbol{W}_l^{QK}[\omega, \nu] g_\nu^{(l-1)}(x_\gamma) \right) g_\beta^{(l-1)}(x_\gamma)$$

$$+ \sum_{\tilde{\gamma}=1}^{N} \sum_{\omega,\nu=1}^{d} \left(f_\omega^{(l-1)}(x_\alpha) \boldsymbol{W}_l^{QK}[\omega, \nu] f_\nu^{(l-1)}(x_{\tilde{\gamma}}) \right) f_\beta^{(l-1)}(x_{\tilde{\gamma}}) + f_\beta^{(l-1)}(x_\alpha).$$

With the Riemann sum approximation: $\int_\Omega p(x)dx \approx \Delta x^D \sum_{k=1}^{N_x} p(x_k)$, one can further reformulate above derivations as:

$$g_\beta^{(l)}(x) \approx \frac{1}{\Delta x^D} \int_\Omega \sum_{\omega,\nu=1}^{d} \left(g_\omega^{(l-1)}(x) \boldsymbol{W}_l^{QK}[\omega, \nu] g_\nu^{(l-1)}(y) \right) g_\beta^{(l-1)}(y) dy$$

$$+ \frac{1}{\Delta x^D} \int_\Omega \sum_{\omega,\nu=1}^{d} \left(g_\omega^{(l-1)}(x) \boldsymbol{W}_l^{QK}[\omega, \nu] f_\nu^{(l-1)}(y) \right) f_\beta^{(l-1)}(y) dy + g_\beta^{(l-1)}(x),$$

$$f_\beta^{(l)}(x) \approx \frac{1}{\Delta x^D} \int_\Omega \sum_{\omega,\nu=1}^{d} \left(f_\omega^{(l-1)}(x) \boldsymbol{W}_l^{QK}[\omega, \nu] g_\nu^{(l-1)}(y) \right) g_\beta^{(l-1)}(y) dy$$

$$+ \frac{1}{\Delta x^D} \int_\Omega \sum_{\omega,\nu=1}^{d} \left(f_\omega^{(l-1)}(x) \boldsymbol{W}_l^{QK}[\omega, \nu] f_\nu^{(l-1)}(y) \right) f_\beta^{(l-1)}(y) dy + f_\beta^{(l-1)}(x).$$

Therefore, the attention mechanism of each layer is in fact an integral operator after a rescaling:

$$\begin{pmatrix} g^{(l)}(x) \\ f^{(l)}(x) \end{pmatrix} = \int_\Omega K^{(l)}(x, y) \begin{pmatrix} g^{(l-1)}(y) \\ f^{(l-1)}(y) \end{pmatrix} dy + \begin{pmatrix} g^{(l-1)}(x) \\ f^{(l-1)}(x) \end{pmatrix}, \tag{20}$$

with the kernel $K^{(l)}(x, y)$ defined as:

$$\begin{bmatrix} \sum_{\omega,\nu=1}^d \left(g_\omega^{(l-1)}(x) \boldsymbol{W}_l^{QK}[\omega, \nu] g_\nu^{(l-1)}(y) \right) & \sum_{\omega,\nu=1}^d \left(g_\omega^{(l-1)}(x) \boldsymbol{W}_l^{QK}[\omega, \nu] f_\nu^{(l-1)}(y) \right) \\ \sum_{\omega,\nu=1}^d \left(f_\omega^{(l-1)}(x) \boldsymbol{W}_l^{QK}[\omega, \nu] g_\nu^{(l-1)}(y) \right) & \sum_{\omega,\nu=1}^d \left(f_\omega^{(l-1)}(x) \boldsymbol{W}_l^{QK}[\omega, \nu] f_\nu^{(l-1)}(y) \right) \end{bmatrix}. \tag{21}$$

For the $L-$th layer update, we denote the approximated value of $f_\beta(x_\alpha)$ as $\tilde{f}_\beta(x_\alpha) := \boldsymbol{X}_{\text{out}}[\alpha, \beta]$, then

$$\tilde{f}_\beta(x_\alpha) = \sum_{\gamma=1}^N K[\mathbf{u}_{1:d}, \mathbf{f}_{1:d}; \theta][\alpha, \gamma] g[u_\beta](x_\gamma)$$

$$= \sum_{\lambda,\gamma=1}^N \sum_{\omega,\nu=1}^d W^{P,u}[\alpha, \lambda] \left(\boldsymbol{U}^{(L-1)}[\lambda, \omega] W_L^{QK}[\omega, \nu] \boldsymbol{U}^{(L-1)}[\gamma, \nu] \right) g[u_\beta](x_\gamma)$$

$$+ \sum_{\lambda,\gamma=1}^N \sum_{\omega,\nu=1}^d W^{P,f}[\alpha, \lambda] \left(\boldsymbol{F}^{(L-1)}[\lambda, \omega] W_L^{QK}[\omega, \nu] \boldsymbol{U}^{(L-1)}[\gamma, \nu] \right) g[u_\beta](x_\gamma)$$

$$= \sum_{\lambda,\gamma=1}^N \sum_{\omega,\nu=1}^d W^{P,u}[\alpha, \lambda] \left(g_\omega^{(L-1)}(x_\lambda) W_L^{QK}[\omega, \nu] g_\nu^{(L-1)}(x_\gamma) \right) g[u_\beta](x_\gamma)$$

$$+ \sum_{\lambda,\gamma=1}^N \sum_{\omega,\nu=1}^d W^{P,f}[\alpha, \lambda] \left(f_\omega^{(L-1)}(x_\lambda) W_L^{QK}[\omega, \nu] g_\nu^{(L-1)}(x_\gamma) \right) g[u_\beta](x_\gamma)$$

$$\approx \frac{1}{\Delta x^{2D}} \int_\Omega \int_\Omega \sum_{\omega,\nu=1}^d W^{P,u}(x_\alpha, z) \left(g_\omega^{(L-1)}(z) W_L^{QK}[\omega, \nu] g_\nu^{(L-1)}(y) \right) dz \, g[u_\beta](y) dy$$

$$+ \frac{1}{\Delta x^{2D}} \int_\Omega \int_\Omega \sum_{\omega,\nu=1}^d W^{P,f}(x_\alpha, z) \left(f_\omega^{(L-1)}(z) W_L^{QK}[\omega, \nu] g_\nu^{(L-1)}(y) \right) dz \, g[u_\beta](y) dy.$$

Hence, a (rescaled) continuous limit of the kernel writes:

$$K[\mathbf{u}_{1:d}, \mathbf{f}_{1:d}; \theta](x, y) = \int_\Omega \sum_{\omega,\nu=1}^d W^{P,u}(x, z) \left(g_\omega^{(L-1)}(z) W_L^{QK}[\omega, \nu] g_\nu^{(L-1)}(y) \right) dz$$

$$+ \int_\Omega \sum_{\omega,\nu=1}^d W^{P,f}(x, z) \left(f_\omega^{(L-1)}(z) W_L^{QK}[\omega, \nu] g_\nu^{(L-1)}(y) \right) dz.$$

B Proofs and connection with regularized estimator

B.1 Proofs

Proof of Lemma 4.1. With $\{r_k\}_{k=1}^N$, we can write the attention-based kernel in (7) as

$$K[\mathbf{u}_{1:d}, \mathbf{f}_{1:d}; \theta](r_k) = \sum_{k'=1}^N W^{P,u}(r_{k'}) \sigma \left(\sum_{l=1}^{d_k} \sum_{j=1}^d \sum_{j=1}^d \mathbf{u}_j(r_{k'}) W^Q[j, l] \cdot W^K[j', l] \mathbf{u}_{j'}(r_k) \right)$$

$$+ W^{P,f} \sigma \left(\sum_{l=1}^{d_k} \sum_{j=1}^d \sum_{j'=1}^d [\mathbf{f}_j W^Q[j, l] \cdot W^K[j', l] \mathbf{u}_{j'}(r_k)] \right). \tag{22}$$

Denoting $W^{QK}[j',j] = \sum_{l=1}^{d_k} W^Q[j,l] \cdot W^K[j',l]$, we write the kernel in (22) as

$$K[\mathbf{u}_{1:d}, \mathbf{f}_{1:d}; \theta](r) = \sum_{r'=1}^{N} W^{P,u}(|r'|)\sigma\left(\sum_{j=1}^{d}\sum_{j'=1}^{d}\left[g[u](r', x_{i_j})W^{QK}[j,j']g[u](r, x_{i_{j'}})\right]\right)$$

$$+ W^{P,f}\sigma\left(\sum_{i=1}^{d}\sum_{j=1}^{d}\left[f(x_{i_{j'}})W^{QK}[j,j']g[u](r, x_{i_j})\right]\right).$$

Then, as $d \to \infty$ is achieved by refining the spatial mesh, viewing the summation in j as Riemann sum,

$$\lim_{d\to\infty}\sum_{j=1}^{d}\sum_{j'=1}^{d} g[u](r', x_{i_j})W^{QK}[j,j']g[u](r, x_{i_{j'}}) = \int\int g[u](r,x)W^{QK}(x,y)g[u](s,y)dxdy,$$

where the integral exists since $g[u](r,x)W^{QK}(x,y)$ is bounded. Sending also the number of tokens, N, to infinity, we obtain the limit attention model in (14). $\square$

Proof of Lemma 4.2. The proof is adapted from Lu et al. [2023, 2022]. Write $K(r) = K[\mathbf{u}_{1:d}^\eta, \mathbf{f}_{1:d}^\eta](r)$. Notice that the loss function in (9) can be expanded as

$$\mathcal{E}(K) = \frac{1}{n_{train}}\sum_{\eta=1}^{n_{train}}\sum_{i=1}^{d_u}\int_\Omega\left[\int_0^\delta K(r)g[u_i^\eta](r,x))dr - f_i^\eta(x)\right]^2 dx$$

$$= \frac{1}{n_{train}}\sum_{\eta=1}^{n_{train}}\sum_{i=1}^{d_u}\int_0^\delta\int_0^\delta K(s)K(r)\int_\Omega g[u_i^\eta](s,x))g[u_i^\eta](r,x))dxdrds$$

$$- \frac{2}{n_{train}}\sum_{\eta=1}^{n_{train}}\sum_{i=1}^{d_u}\int_0^\delta K(r)\int_\Omega g[u_i^\eta](r,x)f_i^\eta(x)dxdr + Const.$$

$$= \langle\mathcal{L}_{\bar{G}}K, K\rangle_{L_\rho^2} - 2\langle K, K^D\rangle_{L_\rho^2} + Const.,$$

where $\mathcal{L}_{\bar{G}} : L_\rho^2 \to L_\rho^2$ is the integral operator

$$\mathcal{L}_{\bar{G}}K(s) := \int_0^\delta K(r)\bar{G}(r,s)dr$$

and K^D is the Riesz representation of the bounded linear functional

$$\langle K, K^D\rangle_{L_\rho^2} = \frac{1}{n_{train}}\sum_{\eta=1}^{n_{train}}\sum_{i=1}^{d_u}\int_0^\delta K(r)\int_\Omega g[u_i^\eta](r,x)f_i^\eta(x)dxdr.$$

Thus, the quadratic loss function has a unique minimizer in $\text{Null}(\mathcal{L}_{\bar{G}})^\perp$.

Meanwhile, since the data pairs are continuous with compact support, the function $\bar{G}$ is a square-integrable reproducing kernel. Thus, the operator $\mathcal{L}_{\bar{G}}$ is compact and $H_G = \mathcal{L}_{\bar{G}}^{1/2}L_\rho^2$. Then, $\text{Null}(\mathcal{L}_{\bar{G}})^\perp = \overline{H_G}$, where the closure is with respect to L_ρ^2. $\square$

B.2 Connection with regularized estimator

Consider the inverse problem of estimating the nonlocal kernel K given a data pair (u, f). In the classical variational approach, one seeks the minimizer of the following loss function

$$\mathcal{E}(K) = \int_\Omega\left[\int_0^\delta K(r)g[u](r,x)dr - f(x)\right]^2 dx \tag{23}$$

The inverse problem is ill-posed in the sense that the minimizer can often be non-unique or sensitive to the noise or measurement error in data (u, f). Thus, regularization is crucial to produce a stable solution.

To connect with the attention-based model, we consider regularizing using an RKHS H_W with a square-integrable reproducing kernel W. One seeks an estimator in H_W by regularizing the loss with the $\|K\|_{H_W}^2$, and minimizes the regularized loss function

$$\mathcal{E}_{\lambda,W}(K) = \mathcal{E}(K) + \lambda \int_0^\infty \int_0^\infty K(s)K(r)W(r,s)drds. \tag{24}$$

The next lemma shows that the regularized estimator is a nonlinear function of the data pair (u, f), where the nonlinearity comes from the kernel G_u and the parameter λ_*.

Lemma B.1. *The regularized loss function in $\mathcal{E}_{\lambda,W}(K)$ in (24) is*

$$\mathcal{E}_{\lambda,W}(K) = \int_0^\infty \int_0^\infty K(r)K(s)[G_u(r,s) + \lambda W(r,s)]drds - 2\int_0^\infty K(r)K^{u,f}(r)dr + Const., \tag{25}$$

where G_u is defined in (27). Its minimizer is

$$\widehat{K} = (\mathcal{L}_{G_u} + \lambda_* \mathcal{L}_W)^{-1} K^{u,f}, \tag{26}$$

where $\mathcal{L}_{G_u}$ and $\mathcal{L}_W$ are integral operators with integral kernels G_u defined in (27) and W,

$$\mathcal{L}_{G_u}K(s) := \int_0^\infty K(r)G_u(r,s)dr, \quad \mathcal{L}_W K(s) := \int_0^\infty K(r)W(r,s)dr,$$

λ_* is the optimal hyper-parameter controlling the strength of regularization, and $K^{u,f}(r) = \int_\Omega g[u](r,x)f(x)dx$ is a function determined by the data (u, f).

When there is no prior information on the regularization, which happens often for the learning of the kernel, one can use the data-adaptive RKHS H_{G_u} with the reproducing kernel G_u determined by data:

$$G_u(r,s) = \int_\Omega g[u](r,x)g[u](s,x)dx. \tag{27}$$

Lu et al. [2023] shows that this regularizer can lead to consistent convergent estimators.

Remark B.2 (Discrete data and discrete inverse problem). In practice, the datasets are discrete. One can view the discrete inverse problem as a discretization of the continuous inverse problem. Assume that the integrands are compactly supported and when the integrals are approximated by Riemann sums, we can write loss function for discrete $\mathbf{K} = (K(r_1), \ldots, K(r_N))^\top \in \mathbb{R}^{N \times 1}$ as

$$\mathcal{E}_{\lambda,W}(K) \approx \sum_{k,k'=1}^{N,N} K(r_k)K(r_{k'})[G_u(r_k, r_{k'}) + \lambda W(r_k, r_{k'})](\Delta r)^2 - 2\sum_{k=1}^N K(r_k)K^{u,f}(r_k)\Delta r + Const..$$

$$= \mathbf{K}^\top [\mathbf{G}_u + \lambda \mathbf{W}]\mathbf{K} - 2\mathbf{K}^\top \mathbf{K}^{u,f} + Const., \tag{28}$$

where, recalling the definition of the token $\mathbf{u}_{1:d}$ in (5),

$$\mathbf{G}_u = (G_u(r_k, r_{k'}))_{1 \le k,k' \le K} = \left(\int_\Omega g[u](x, r_k)g[u](x, r_{k'})dx\right)_{1 \le k,k' \le K} \approx \mathbf{u}_{1:d}\mathbf{u}_{1:d}^\top,$$

$\mathbf{W} = (W(r_k, r_{k'}))_{1 \le k,k' \le K}$ and $\mathbf{K}^{u,f} = \left(\int_\Omega g[u](r_k, x)f(x)dx\right)_{1 \le k \le N}$.

The minimizer of this discrete loss function with the optimal hyper-parameter λ_* is

$$\widehat{\mathbf{K}} = (\mathbf{G}_u + \lambda_* \mathbf{W})^{-1}\mathbf{K}^{u,f}.$$

In particular, when taking $\mathbf{W} = \mathbf{G}_u$ and using the Neumann series $(\lambda_*^{-1}\mathbf{G}_u^2 + I)^{-1} = \sum_{k=0}^\infty (-1)^k \lambda_*^{-k}\mathbf{G}_u^{2k}$, we have

$$\widehat{\mathbf{K}} = (\lambda_*^{-1}\mathbf{G}_u^2 + I)^{-1}\lambda_*^{-1}\mathbf{G}_u\mathbf{K}^{u,f} = \lambda_*^{-1}\mathbf{G}_u\mathbf{K}^{u,f} - \lambda_*^{-2}\mathbf{G}_u^3\mathbf{K}^{u,f} + \sum_{k=2}^\infty (-1)^k \lambda_*^{-k}\mathbf{G}_u^{2k}\mathbf{G}_u\mathbf{K}^{u,f}.$$

In particular, λ_* depends on both $\mathbf{G}_u$ and $\mathbf{K}^{u,f}$. Hence, the estimator $\widehat{\mathbf{K}}$ is nonlinear in $\mathbf{G}_u$ and $\mathbf{K}^{u,f}$, and it is important to make the attention depend nonlinearly on the token $\mathbf{u}_{1:d}$, as in (7).

Proof of Lemma B.1. Since K is radial and noticing that

$$\int_{|\xi|=1} (u(x+r\xi) - u(x))d\xi = u(x+r) + u(x-r) - 2u(x) = g[u](r,x)$$

since $\xi \in \mathbb{R}^1$, we can write

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} K(|s|)K(|r|) \int_{\Omega} (u(x+s) - u(x))(u(x+r) - u(x))dxdrds$$

$$= \int_0^{\infty} \int_0^{\infty} K(r)K(s) \int_{|\xi'|=1} \int_{|\xi|=1} \int_{\Omega} (u(x+s\xi') - u(x))(u(x+r\xi) - u(x))dxd\xi d\xi'$$

$$= \int_0^{\infty} \int_0^{\infty} K(r)K(s) \int_{\Omega} g[u](r,x)g[u](s,x)dxdrds$$

$$= \int_0^{\infty} \int_0^{\infty} K(r)K(s)G_u(r,s)drds.$$

Meanwhile, by the Riesz representation theorem, there exists a function $K^{u,f} \in L^2(0,\infty)$ such that

$$\int_{-\infty}^{\infty} K(|r|) \int_{\Omega} (u(x+r) - u(x))f(x)dxdr$$

$$= \int_0^{\infty} K(r) \int_{|\xi|=1} \int_{\Omega} (u(x+r) - u(x))f(x)dxd\xi dr$$

$$= \int_0^{\infty} K(r) \int_{\Omega} g[u](r,x)f(x)dxdr = \int_0^{\infty} K(r)K^{u,f}(r)dr.$$

Combining these two equations, we can write the loss function as

$$\mathcal{E}(K) = \int_{\Omega} \left[\int_{-\infty}^{\infty} K(|s|)(u(x+s) - u(x))ds - f(x) \right]^2 dx$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} K(|s|)K(|r|) \int_{\Omega} (u(x+s) - u(x))(u(x+r) - u(x))dxdrds$$

$$- \int_{-\infty}^{\infty} K(|s|) \int_{\Omega} (u(x+s) - u(x))f(x)dxds + Const.$$

$$= \int_0^{\infty} \int_0^{\infty} K(r)K(s)G_u(r,s)drds - 2\int_0^{\infty} K(r)K^{u,f}(r)dr + Const..$$

Then, we can write the regularized loss function $\mathcal{E}_{\lambda,W}(K)$ as

$$\mathcal{E}_{\lambda,W}(K) = \int_0^{\infty} \int_0^{\infty} K(r)K(s)[G_u(r,s) + \lambda W(r,s)]drds - 2\int_0^{\infty} K(r)K^{u,f}(r)dr + Const.$$

$$= \langle (\mathcal{L}_{G_u} + \lambda\mathcal{L}_W)K, K\rangle_{L^2(0,\infty)} - 2\langle K, K^{u,f}\rangle_{L^2(0,\infty)} + Const.$$

Selecting the optimal hyper-parameter λ_*, which depends on both (u, f) and W, and setting the Fréchet derivative of $\mathcal{E}_{\lambda,W}$ over $L^2(0,\infty)$ to be zero, we obtain the regularized estimator in (26). □

C Data generation and additional discussion

C.1 Example 1: radial kernel learning

In all settings except the "single task" one, all kernels act on the same set of functions $\{u_i\}_{i=1,2}$ with $u_1 = \cos(x)\mathbf{1}_{[-\pi,\pi]}(x)$ and $u_2(x) = \sin(2x)\mathbf{1}_{[-\pi,\pi]}(x)$. In the "single task" setting, to create more diverse samples, the single kernel acts on a set of 14 functions: $u_k = \cos(kx)\mathbf{1}_{[-\pi,\pi]}(x)$, $k = 1, \cdots, 7$ and $u_k(x) = \sin(kx)\mathbf{1}_{[-\pi,\pi]}(x)$, $k = 8, \cdots, 14$. In the ground-truth model, the integral $\mathcal{L}_{\gamma_\eta}[u_i]$ is computed by the adaptive Gauss-Kronrod quadrature method, which is much more accurate than the Riemann sum integrator that we use in the learning stage. To create discrete datasets with different resolutions, for each $\Delta x \in 0.0125 \times \{1, 2\}$, we take values $\{u_i, f_i\}_{i=1}^N =$
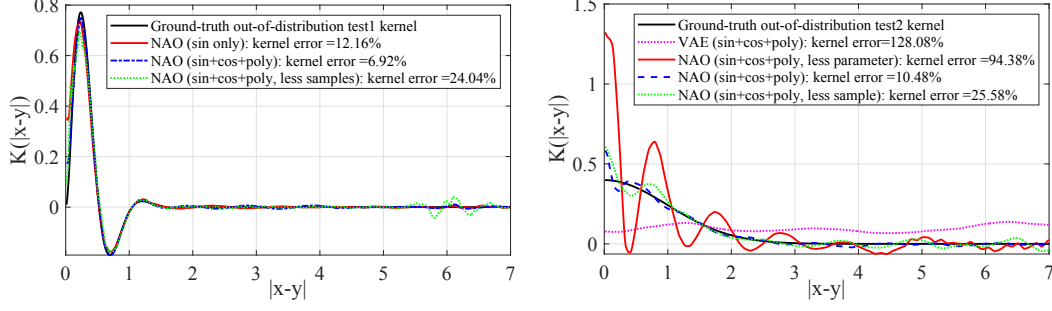
Figure 5: OOD test results on radial kernel learning, with diverse training tasks and $d = 302$. OOD1 (left): true kernel $\gamma(r) = r(11 - r)\exp(-5r)\sin(6r)\mathbf{1}_{[0,11]}(r)$; OOD2 (right): true kernel $\gamma(r) = \frac{\exp(-0.5r^2)}{\sqrt{2\pi}}$, a Gaussian kernel which is very different from all training tasks.

$\{u_i(x_j), f_i(x_j) : x_j \in [-40, 40], j = 1, \ldots, J\}_{i=1}^N$, where x_j is the grid point on the uniform mesh of size Δx. We form a training sample of each task by taking d pairs from this task. When taking the token size d and k function pairs, each task contains $\lfloor \frac{2265}{d} \rfloor \times k$ samples.

More diverse tasks. To further evaluate the generalization capability of NAO as a foundation model, we add another two types of kernels into the training dataset. The training dataset is now constructed based on 21 kernels of three groups, with 15 samples on each kernel:

- sine-type kernels: $\gamma_\eta^{\sin}(r) = \exp(-\eta r)\sin(6r)\mathbf{1}_{[0,11]}(r)$, $\eta = 1, 2, 3, 4, 6, 7, 8$.

- cosine-type kernels: $\gamma_\eta^{\cos}(r) = \frac{10-r}{20}\cos(\eta r)(10 - r)\mathbf{1}_{[0,10]}(r)$, $\eta = 0, 1, 2, 3, 4, 5, 6$.

- polynomial-type kernels: $\gamma_\eta^{poly}(r) = \exp(-0.1r)p_\eta\left(\frac{r-10}{10}\right)\mathbf{1}_{[0,10]}(r)$, $\eta = 1, 2, 3, 4, 5, 6, 7$, where p_η is the degree-η Legendre polynomial.

Based on this enriched dataset, besides the original "sine only" setting, three additional settings are considered to compare the generalizability across different settings. The results are reported in Table 4 and Figure 5. In addition to the original setting corresponding to all "sine" kernels, in part II of Table 4 (denoted as "sine+cosine+polyn"), we consider a "diverse task" setting, where all 315 samples are employed in training. In part III, we consider a "single task" setting, where only the first sine-type kernel, $\gamma(r) = \exp(-r)\sin(6r)\mathbf{1}_{[0,11]}(r)$, is considered as the training task, with 105 samples on this task. Lastly, in part IV we demonstrate a "fewer samples" setting, where the training dataset still consists of all 21 tasks but with only 5 samples on each task. For testing, besides the ID and OOD tasks in the ablation study, we add an additional OOD task with a Gaussian-type kernel $\gamma_{ood2}(r) = \frac{\exp(-0.5r^2)}{\sqrt{2\pi}}$, which is substantially different from all training tasks.

As shown in Table 4, considering diverse datasets helps in both OOD tests (the kernel error is reduced from 9.14% to 6.92% in OOD test 1, and from 329.66% to 10.48% in OOD test 2), but not for the ID test (the kernel error is slightly increased from 4.03% to 5.04%). We also note that since the "sine only" setting only sees systems with the same kernel frequency in training, it does not generalize to OOD test 2, where it becomes necessary to include more diverse training tasks. As the training tasks become more diverse, the intrinsic dimension of kernel space increases, requiring a larger rank size d_k of the weight matrices. When comparing the "fewer samples" setting with the "sine only" setting, the former exhibits better task diversity but fewer samples per task. One can see that the performance deteriorates on the ID test but improves on OOD test 2. This observation emphasizes the balance between the diversity of tasks and the number of training samples per task.

C.2 Example 2: solution operator learning

For one example, we generate the synthetic data based on the Darcy flow in a square domain of size 1×1 subjected to Dirichlet boundary conditions. The problem setting is: $-\nabla(b(x)\nabla p(x)) = g(x)$ subjected to $p(x) = 0$ on all boundaries. This equation describes the diffusion in heterogeneous fields, such as the subsurface flow of underground water in porous media. The heterogeneity is represented by the location-dependent conductivity $b(x)$. $p(x)$ is the source term, and the hydraulic

Table 4: Training and test for the radial kernel problem with more diverse tasks with fixed $d = 302$, where bold numbers highlight the best methods across different data settings. These results emphasize the balance between task diversity and the number of samples per task.

d_k	model	#param	Operator test error			Kernel test error		
			ID	OOD1	OOD2	ID	OOD1	OOD2
sine only: Train on sine dataset, 105 samples in total								
10	Discrete-NAO	16526	1.33%	25.81%	138.00%	29.02%	28.80%	97.14%
	NAO	18843	1.48%	8.10%	221.04%	5.40%	10.02%	420.60%
20	Discrete-NAO	28645	1.35%	18.70%	99.50%	35.49%	30.81%	101.08%
	NAO	30963	1.33%	9.12%	211.78%	4.63%	9.14%	329.66%
40	Discrete-NAO	52886	1.30%	31.37%	49.83%	38.89%	30.02%	129.84%
	NAO	55203	**0.67%**	7.34%	234.43%	**4.03%**	12.16%	1062.3%
	Autoencoder	16424	12.97%	1041.49%	698.72%	22.56%	136.79%	304.37%
sine+cosine+polyn: Train on diverse (sine, cosine and polynomial) dataset, 315 samples in total								
10	Discrete-NAO	16526	2.27%	13.02%	11.50%	10.41%	30.79%	77.80%
	NAO	18843	2.34%	14.37%	10.05%	7.26%	28.23%	94.38%
20	Discrete-NAO	28645	1.60%	6.03%	19.73%	21.83%	21.29%	18.97%
	NAO	30963	1.64%	3.25%	**3.58%**	5.45%	8.87%	15.82%
40	Discrete-NAO	52886	1.45%	5.49%	18.26%	20.07%	19.46%	18.44%
	NAO	55203	1.54%	**3.09%**	7.69%	5.04%	**6.92%**	**10.48%**
	Autoencoder	56486	12.67%	341.96%	211.61%	27.06%	52.43%	128.08%
Single task: Train on a single sine dataset, 105 samples in total								
10	NAO	18843	104.49%	104.37%	56.64%	100.31%	100.00%	94.98%
20	NAO	30963	116.89%	105.52%	85.40%	99.02%	99.55%	98.37%
40	NAO	55203	111.33%	104.12%	76.78%	101.61%	100.33%	95.90%
Fewer samples: Train on diverse (sine, cosine and polynomial) dataset, 105 samples in total								
10	NAO	18843	4.23%	15.34%	11.13%	10.11%	25.23%	97.63%
20	NAO	30963	4.15%	10.15%	9.59%	8.84%	23.08%	20.05%
40	NAO	55203	3.69%	11.67%	6.08%	9.19%	24.04%	25.58%

height $g(x)$ is the solution. For each data instance, we solve the equation on a 21×21 grid using an in-house finite difference code. We consider 500 random microstructures consisting of two distinct phases. For each microstructure, the square domain is randomly divided into two subdomains with different conductivity of either 12 or 3. Additionally, we consider 100 different $g(x)$ functions obtained via a Gaussian random field generator. For each microstructure, we solve the Darcy problem considering all 100 source terms, resulting in a dataset of $N = 10,000$ function pairs in the form of $\{p_i(x_j), g_i(x_j)\}_{i=1}^N$, and $j = 1, 2, \cdots, 441$ where x_j's are the discretization points on the square domain.

In operator learning settings, we note that the permutation of function pairs in each sample should not change the learned kernel, i.e., one should have $K[\mathbf{u}_{1:d}, \mathbf{f}_{1:d}] = K[\mathbf{u}_{\sigma(1:d)}, \mathbf{f}_{\sigma(1:d)}]$, where σ is the permutation operator. Hence, we augment the training dataset by permuting the function pairs in each task. Specifically, with 100 microstructures (tasks) and 100 function pairs per task, we randomly permut the function pairs and take 100 function pairs for 100 times per task. As a result, we can generate a total of 10000 samples (9000 for training and 1000 for testing) in the form of $\{\boldsymbol{u}_{1:100}^\eta, \boldsymbol{f}_{1:100}^\eta\}_{\eta=1}^{10000}$.

C.3 Example 3: heterogeneous material learning

In the Mechanical MNIST dataset, we generate a large set of heterogeneous material responses subjected to mechanical forces. This is similar to the approach in Lejeune [2020]. The material property (stiffness) of the heterogeneous medium is constructed assuming a linear scaling between 1 and 100, according to the gray-scale bitmap of the MNIST images, which results in a set of 2D square domains with properties that vary according to MNIST digit patterns. The problem setting for this data set is the equilibrium equation: $-\nabla \cdot P(x) = f(x)$ subjected to Dirichlet boundary condition of zero displacement and nonzero variable external forces: $f(x)$. $P(x) = \hat{P}(I + \nabla u(x))$ is the stress tensor and is a nonlinear function of displacement $u(x)$. The choice of material models determines the stress function, and here we employ the Neo-Hookean material model as in Lejeune

[2020]. From MNIST, we took 50 samples of each digit resulting in a total of 500 microstructures. We consider 200 different external forces $f(x)$ obtained via a Gaussian random field, and solve the problem for each pair of microstructure and external force, resulting in $N = 100,000$ samples. To solve each sample, we use FEniCS finite element package considering a 140×140 uniform mesh. We then downsample from the finite element nodes to get values for the solution $u(x)$ and external force $f(x)$ on a the coarser 29×29 equi-spaced grid. The resulting dataset is of the form: $\{u_i(x_j), f_i(x_j)\}_{i=1}^{N}$, and $j = 1, 2, ..., 841$ where x_j's are the equi-spaced sampled spatial points.

Similar to Example 2, we perform the permutation trick to augment the training data. In particular, with 500 microstructures (tasks) and 200 function pairs per task, we randomly permute the function pairs and take 100 function pairs for 100 times per task. As a result, we can generate a total of 50000 samples, where 45000 are used for training and 5000 for testing.

D   Computational complexity

Denote $(N, d, d_k, L, n_{train})$ as the size of the spatial mesh, the number of data pairs in each training model, the column width of the query/key weight matrices W^Q and W^K, the number of layers, and the number of training models. The number of trainable parameters in a discrete NAO is of size $O(L \times d \times d_k + N^2)$. For continuous-kernel NAO, taking a three-layer MLP as a dense net of size $(d, h_1, h_2, 1)$ for the trainable kernels $W^{P,u}$ and $W^{P,f}$ for example, its the corresponding number of trainable parameters is $O(d \times h_1 + h_1 \times h_2)$. Thus, the total number of trainable parameters for a continuous-kernel NAO is $O(L \times d \times d_k + d \times h_1 + h_1 \times h_2)$.

The computational complexity of NAO is quadratic in the length of the input and linear in the data size, with $O([d^2(3N + d_k) + 6N^2d]Ln_{train})$ flops in each epoch in the optimization. It is computed as follows for each layer and the data of each training model: the attention function takes $O(d^2d_k + 2Nd^2 + 4N^2d)$ flops, and the kernel map takes $O(d^2d_k + Nd^2 + 2N^2d)$ flops; thus, the total is $O(d^2(3N + d_k) + 6N^2d)$ flops. In inverse PDE problems, we generally have $d \ll N$, and hence the complexity of NAO is $O(N^2d)$ per layer per sample.

Compared with other methods, it is important to note that NAO solves both forward and ill-posed inverse problems using multi-model data. Thus, we don't compare it with methods that solve the problems for a single model data, for instance, the regularized estimator in Appendix B. Methods solving similar problems are the original attention model [Vaswani et al., 2017], convolution neural network (CNN), and graph neural network (GNN). As discussed in Vaswani et al. [2017], these models have a similar computational complexity, if not any higher. In particular, the complexity of the original attention model is $O(N^2d)$, and the complexity of CNN is $O(kNd^2)$ with k being the kernel size, and a full GNN is of complexity $O(N^2d^2)$.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We have clearly stated our claims in the abstract and introduction, including the key contributions that our proposed model NAO provides a novel input-dependent kernel for context extraction, integrates forward and inverse PDE predictions in one framework, and offers the capability in solving ill-posed inverse PDE problems.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the work is discussed in the Conclusion section.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: We have provided the full set of assumptions and corresponding complete proofs under each proposed Lemma with detailed derivations included in the Appendix A and B.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We have elaborated the detailed architectures of the proposed model in the main body of the paper as well as data generation details in Appendix C. The code and accompanying data have been released at `https://github.com/fishmoon1234/NAO`.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code and accompanying data have been released at `https://github.com/fishmoon1234/NAO`.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We briefly state the training and test strategies at the beginning of the Experiments section, with the full details provided in Appendix C.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Error bars are not reported due to the limited computational resource. Nevertheless, we have tuned the hyperparameters of all the considered models and reported extensive results in a large variety of settings, and they all support our claim.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The computing resources are reported in the first paragraph of the Experiments section. In particular, all the experiments are conducted on a single NVIDIA GeForce RTX 3090 GPU with 24 GB memory.

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in the paper conform with the NeurIPS Code of Ethics.

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The discussion on the societal impacts can be found in the Conclusion section.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the original papers that produced the code package or dataset where appropriate.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The code and accompanying data have been released at https://github.com/fishmoon1234/NAO, with instructions and pretrained models provided.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve crowdsourcing nor research with human subjects.