RIESZ NEURAL OPERATOR FOR SOLVING PARTIAL DIFFERENTIAL EQUATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Local non-stationarity is pivotal to solving partial differential equations (PDEs). However, in operator learning, the spatially local information inherent in the data is often overlooked. Even when explicitly modeled, it is usually collapsed into local superpositions within the model architecture, preventing full exploitation of local features in physical phenomena. To address this limitation, our paper proposes a novel Riesz Neural Operator (RNO) based on the spectral derivative representation. Since PDEs are fundamentally governed by local derivatives, RNO leverages the Riesz transform, a natural spectral representation of derivatives, to mix global spectral information with local directional variations. This approach allows the RNO to outperform existing operators in complex scenarios that require sensitivity to local detail. Our design bridges the gap between physical interpretability and local dynamics. Experimental results demonstrate that the RNO consistently achieves superior prediction accuracy and generalization performance compared to existing approaches across various benchmark PDE problems and complex real-world datasets, presenting superior non-linear reconstruction capability in model analysis.

1 INTRODUCTION

Accurately simulating physical systems governed by partial differential equations (PDEs) remains foundational across fluid dynamics (Herde et al., 2024), materials science (Li et al., 2024), and climate modeling (Kurth et al., 2023). Neural operators have recently emerged as powerful surrogates for PDE solution operators, offering significant reductions in compute time and broad adaptability. However, it is a complex task to capture and predict the non-linear and local changes due to physical information missing, transform without local emphasis, etc. This work leverages local spectral derivatives to improve the precision of PDE solutions.

Existing operator-learning frameworks, including DeepONet (Lu et al., 2021; Gu et al., 2025), the Fourier Neural Operator (FNO) (Li et al., 2020a), and the Laplace Neural Operator (LNO) (Cao et al., 2024a), leverage specific transforms to model PDE dynamics. Suppose we analogize the process of function learning to the performance of a musical score. In DeepONet, each partitioned ensemble (trunk function) is assigned a fixed section, which plays as required by the score. This design elegantly captures the global melody of function-to-function mappings. However, it lacks control over melodic variations and requires an excessively large ensemble for complex scores. FNO, by contrast, assigns finer-grained roles with varying intensities to all performers but struggles to capture melodic diversity and local harmonies. LNO augments FNO with a smoother, yielding greater stability at the cost of detailed melodies. These intriguing comparisons are presented in Figure 1(a). Transformer-based operator learners similarly apply global attention to inputs. As a result, physical locality remains underexploited in operator learning, leading to two major shortcomings: (i) Non-stationary information is largely encoded in local spatial features (Pacejka & Besselink, 1997). Since the temporal dimension is typically embedded implicitly during training, such non-stationarity is not explicitly represented (Barberi & Kruse, 2023). We therefore hypothesize that *once spatial locality is captured adequately, non-stationary temporal signals can be learned naturally*. To substantiate this hypothesis, we seek to reconstruct spatial locality more accurately. (ii) Discarding spatial locality forfeits fine-scale details, especially the small perturbations typical of PDE-governed phenomena. From a spectral perspective, these correspond to high-frequency content; neglecting locality therefore

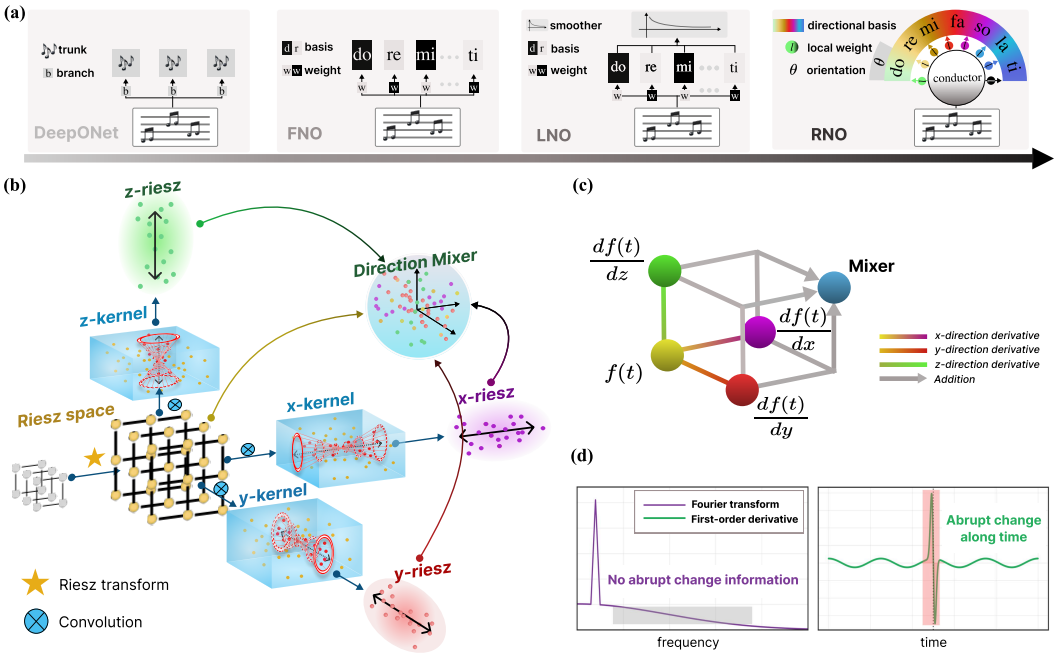

Figure 1: (a) Taking musical score processing as an example, we compare DeepONet, FNO, LNO, and RNO. (b) In the Riesz space, the Riesz transform yields direction-selective kernels (e.g., an x-kernel for the x-direction). Each kernel computes the spectral derivative along its associated direction. (c) The RNO pipeline is formalized mathematically. (d) In the Fourier spectrum (left), no localized change appears in the high-frequency band where a discontinuity is expected, whereas the first-order derivative (right) cleanly captures the emerging jump over time, demonstrating its local sensitivity.

suppresses high-frequency representations and weakens the modeling of strong nonlinearities. For further details on local necessity comparison, please refer to the test in A.1. In the experiments, phase plays a crucial role in the local spectral orientation. Although prior work addresses complex physical systems (Li et al., 2021; Wang et al., 2023), emphasizes local structure (Raonic et al., 2023; Wang et al., 2022) and uses derivative to optimize the loss(Cheng et al., 2025; Qiu et al., 2024), to our knowledge no operator-learning method that optimizes non-linear dynamics by mixing local derivatives. Concurrently, an increasing share of advances stems from incorporating established architectures (e.g., Transformer(Wu et al., 2023), mamba(Tiwari et al., 2025a), and graph network(Li et al., 2025)), whereas recent attempts to modify the neural operator itself have produced limited gains.

Inspired by the Taylor expansion—where higher-order terms encode subtle, local variations (Kanwal & Liu, 1989; Bruhn et al., 2005), therefore derivatives are naturally suited to capture locality Figure 1(c). Yet operator-learning pipelines dominated by global transforms (e.g., Fourier, Laplace) often underrepresent such information, and the energy from abrupt changes is difficult to capture (Alley et al., 2003). To address these bottlenecks, we propose the Riesz Neural Operator (RNO), which improves neural operators' ability to model complex dynamic variations. The Riesz transform in our RNO acts as the **conductor**'s gestures, injecting directional derivatives that govern local dynamics, while the direction mixer serves as **orchestration**, harmonizing different directional components. Analytically, Taylor expansion separates behavior: the DC (direct component) tracks slow trends, whereas abrupt or small-scale variations appear as higher-order terms (Hodaei et al., 2017; Bruhn et al., 2005). Building on this perspective, we apply the Riesz transform to extract first-order derivative information (direction and rate of change) (Unser & Van De Ville, 2009; Wadhwa et al., 2014; Tamada & Igarashi, 2017), and reproject these derivatives onto the original spectrum to emphasize salient components. The result is not only a faithful reproduction of the global score but also a nuanced capture of fine-scale, anisotropic variations — much like the layered expressiveness of a super symphony. The main contributions of this work are as follows:

- Motivated by the Taylor expansion, we encode fine-scale structure using local spectral derivatives. Integrating the Riesz transform sharpens sensitivity to dynamics while preserving physical interpretability, improving expressivity for non-linear phenomena.
- We are the first to explicitly embed the Riesz transform within operator learning, providing a direction-aware operator mixer to unify global basis information and local directional derivatives. Our method supports a principled composition for reconstructing physical dynamics.
- RNO achieves state-of-the-art performance in extensive benchmarks, covering various complex Navier–Stokes PDEs and the real-world climate ERA5 dataset. Experiments indicate that its low-level feature mixer affords a distinct advantage for modeling strong nonlinearities.

2 PRELIMINARIES

Neural Operator. Neural operator learning represents a novel paradigm intersecting scientific computing and machine learning, aimed at learning solution operators for PDEs. Owing to their full differentiability, neural operators permit direct parameter optimization, making them ideal for inverse design and a broad spectrum of inverse problem settings. Initially, neural operators originated from DeepONet(Lu et al., 2021). This paradigm later evolved into the classical FNO(Li et al., 2020a;b), which also performs function-to-function mappings in Fourier space and has spawned several FNO variants(Tran et al., 2021; Li et al., 2023). Recently, a study employed the Laplace transform as an alternative to the Fourier transform(Cao et al., 2024a) to address non-periodic characteristics in data, significantly enhancing the interpretability of neural operators. This highlights the significance of handling non-stationary data(Tiwari et al., 2025b). Additionally, other studies have concentrated on the capability of models to address complex phenomena(Li et al., 2021; Lanthaler et al., 2023; Xiao et al., 2024), strengthening the neural operators' ability to handle nonlinearity.

Meanwhile, due to the complex variability of real-world data and the higher standards required for capturing detailed variations, local properties within neural operators have also attracted attention(Raonic et al., 2023; Tripura & Chakraborty, 2022). Among these studies, the work on neural operators with localized integral and differential kernels emphasizes the necessity of local and differential properties. However, their differential kernels and local kernels are designed separately and introduced into the model as independent modules. Our approach leverages domain knowledge, integrates task-specific features with directional information, and thus possesses the capability to handle local signals, all while maintaining interpretability.

Riesz Methods. Spatial derivatives are widely used in deep learning. In Physics-Informed Neural Networks (PINNs) (Raissi et al., 2019), temporal/spatial derivatives construct PDE residuals, ensuring predictions both fit observations and satisfy governing laws. Derivative signals have likewise been incorporated into LSTM variants to better track state changes (DiPietro & Hager, 2020). By contrast, explicit frequency-domain operations are less common: most prior work analyzes networks' spectral behavior during training (Xu et al., 2019) rather than designing models centered on spectral directionality.

The Riesz transform maps spatial derivatives into the spectral domain while blending integral and differential effects: *the integral term preserves neural operator advantages, and the differential term injects local variation* (Tamada & Igarashi, 2017; Wadhwa et al., 2014). It generalizes the Hilbert transform to higher dimensions; the relationship is derived in A.4. Formally, the transform is a convolution with a Calderón–Zygmund kernel (de Francia et al., 1986), and its kernel-based processing is illustrated in Figure 1(b). A defining property is strong directional selectivity: the transform decomposes signals along specific spatial directions, enabling analysis of anisotropic structure (e.g., edges and textures) (Unser et al., 2009) and isolation of instantaneous phase. These properties make the Riesz transform well suited to data with complex, non-stationary variability.

3 METHODS

3.1 PROBLEM DEFINITION

In physics, it is widely recognized that PDEs characterize a function by constraining its local derivatives(Wu et al., 2023). That is, the evolution of a physical system is described through relations

among its partial derivatives:

$$\mathcal{F}\big(x, u(x), \partial_{x_1} u(x), \ldots, \partial_{x_n} u(x)\big) = 0, \tag{1}$$

where $\partial_{x_i} u(x)$ denotes the local derivative of u with respect to x_i. Physically, local derivatives quantify variations in space and time, whereas PDEs couple these local variations to govern the system's global dynamics. Nevertheless, the implications of this locality have been insufficiently exploited. We therefore seek a principled representation of this local derivative form. As the natural spectral representation of derivatives, the Riesz transform provides more than an engineering convenience: embedding it into neural operators offers a mathematically and physically consistent means of unifying global spectral efficiency with local dynamics. For a further relationship discussion of local derivative and PDEs with a much higher degree of mathematical rigor, we refer the readers to A.1.

3.2 Derivative Information in Physical Fields

Local dynamics via Taylor expansion. Adopting the Eulerian viewpoint commonly used in optical flow analysis(Bruhn et al., 2005), we describe temporal variation of a one–dimensional signal by the first–order Taylor series

$$I(x,t) \;\approx\; f(x) \;+\; \gamma(t)\,\frac{df(x)}{dx}, \tag{2}$$

where $I(x,t) = f(x+\gamma(t))$ and $\gamma(t)$ denotes the instantaneous displacement at time $t>0$ as shown in Figure 6. The term $\gamma(t)\,f'(x)$ therefore *encodes motion magnitude implicitly inside the spatial derivative*. Neglecting this factor, as in most vanilla neural operator blocks, effectively acts as a low-pass filter and can compromise fidelity in highly dynamic regimes.

Directional derivative. Beyond magnitude, first–order derivatives carry orientation information as described in A.1. The directional derivative of $u(\mathbf{x},t)$ along a unit vector $\mathbf{n} = [n_1, \ldots, n_d]^\mathsf{T}$ is

$$\nabla u \cdot \mathbf{n} \;=\; \sum_{i=1}^{d} \frac{\partial u}{\partial x_i}\, n_i. \tag{3}$$

Consequently, model predictions must depend not only on scalar changes but also on the direction in which those changes occur.

Spectral derivative. Taking the temporal derivative of $f(t)$ and transferring it to the Riesz space gives

$$\gamma(t)\,\frac{df(t)}{dt} \quad \overset{\mathscr{R}}{\longleftrightarrow} \quad \gamma(t)\,j\,k\,\mathscr{R}\big\{f(t)\big\}, \tag{4}$$

where k is the wave component and $\mathscr{R}$ denotes Riesz transform. In this representation, a spatial derivative manifests as a 90° phase shift accompanied by an amplitude scaling proportional to k. In 2D, $\mathscr{R}_x$ and $\mathscr{R}_y$ act as projectors onto directions defined by k's angles with the axes, allowing efficient computation of spatial (and higher-order) derivatives. This relation enables the Riesz transform to compute spatial derivatives efficiently and to incorporate higher-order quantities.

Dynamics in Riesz space. Let $h(\mathbf{q},t)$ be a scalar field with spatial coordinates $\mathbf{q} = [q_1, \ldots, q_n]^\mathsf{T} \in \mathbb{R}^n$ and time $t \in \mathbb{R}^+$. Its complex-valued spectrum is defined by the exponential kernel

$$\hat{h}(\mathbf{k},t) \;=\; \int_{\mathbb{R}^n} h(\mathbf{q},t)\, e^{-j\,\mathbf{kq}}\, d\mathbf{q}, \qquad \mathbf{k} = [k_1, \ldots, k_n]^\mathsf{T} \in \mathbb{C}^n, \tag{5}$$

where $j^2 = -1$. The magnitude $\|\mathbf{k}\| = (\sum_{i=1}^{n} k_i^2)^{1/2}$ controls spatial scale, while the vector direction of $\mathbf{k}$ encodes the dominant orientation of the corresponding spectral component. According to the definition of the directional derivative, we apply the n-D Riesz transform $\mathscr{R}$ to direction yields

$$\mathscr{R}\big[\nabla h(\mathbf{q},t)\big](\mathbf{k}) \;=\; \sum_{i=1}^{n} j\,h_i\,\frac{k_i}{\|\mathbf{k}\|}\,h(\mathbf{q},t)e^{-j\,\mathbf{kq}}d\mathbf{q}, \tag{6}$$

showing that differentiation becomes a multiplicative modulation by the directional factor $j\,h_i k_i/\|\mathbf{k}\|$. This highlights a key property of the Riesz framework: normalising by $\|\mathbf{k}\|$ preserves energy while amplifying orientation selectivity, enabling precise estimation of directional rate of change without additional scaling heuristics.

3.3 RIESZ NEURAL OPERATOR

Operator learning. To enhance representational capacity, the input is first projected into a higher-dimensional space by a shallow connection layer, yielding $q(t)$. The transformed $q(t)$ then passes through an integral-kernel layer and a linear layer, followed by an activation, to produce the high-dimensional output representation,

$$u(t) = \sigma\big(Wq(t) + (c * q)(t)\big). \quad \forall t \in D \tag{7}$$

Here, D is a bounded open set; W is a linear transformation; and $(c * q)(t) := \int_D c(t,s)\, q(s)\, \mathrm{d}s$ denotes the integral operator with kernel c. By Green's function theory (Li et al., 2020b), the kernel $c(t,s)$ can approximate solution operators for PDEs. As shown in Figure 8, rather than relying on a single global transform, we couple local and global components to more fully map inputs to outputs. Indeed, RNO preserves the compositional structure characteristic of neural operators. Its overall procedure can be decomposed into three modules,

$$\mathscr{F}_\theta = \mathscr{F}_{\theta_{\text{CoordToRiesz}}} \circ \mathscr{F}_{\sum_{i=1}^{M} \theta_i^R} \circ \mathscr{F}_{\theta_{\text{RieszToCoord}}}, \tag{8}$$

where $\circ$ denotes operator composition and $\mathscr{F}$ is a mapping between spaces. Specifically, $\mathscr{F}_{\theta_{\text{CoordToRiesz}}} : \mathcal{X} \to \mathcal{R}_\mathcal{X}$ lifts the input to the Riesz domain, and $\mathscr{F}_{\theta_{\text{RieszToCoord}}} : \mathcal{R}_\mathcal{Y} \to \mathcal{Y}$ reconstructs the final output. This modular factorization is generic for neural operators. In the Riesz domain, $\mathscr{F}_{\sum_{i=1}^{M} \theta_i^R}$ learns and mixes non-linear, direction-wise mappings, where M is the number of directional components. In practice, components are combined via analytic signals (1D) (Marple, 1999) or monogenic signals (multi-D) (Unser et al., 2009), providing a natural encoding of signal characteristics. Directional weights emphasize orientation-dependent variation. The enhanced Riesz representations are then mapped back by $\mathscr{F}_{\theta_{\text{RieszToCoord}}}$ to yield a spatial output with enhanced directional features.

Riesz conductor. Analogous to convolution performed in a generic frequency domain, the kernel $(c * q)$ is realised by multiplication in Riesz space. For m-dimensional data the i-th directional Riesz transform of q is

$$R_i(\mathbf{k}) \;=\; \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \frac{j\, k_i}{\|\mathbf{k}\|}\, e^{-j\,\mathbf{k}\cdot\mathbf{q}}\, d^m\mathbf{q}, \tag{9}$$

The spectral factor $j\, k_i/\|\mathbf{k}\|$ behaves as a scale-normalised first-order derivative, thus emphasising local variations while suppressing magnitude. Using the forward ($\mathscr{R}$) and inverse ($\mathscr{R}^{-1}$) Riesz transforms, the integral operator is

$$(c * q)(t) \;=\; \mathscr{R}^{-1}\big(\mathscr{R}(c) \;\cdot\; \mathscr{R}(q)\big)(t), \qquad \forall\, t \in D, \tag{10}$$

where "$\cdot$" denotes element-wise complex multiplication. The model overview of RNO is shown in Figure 2. In the Riesz conductor, the directional numbers are not chosen randomly but are instead based on the data's dimensionality. This approach aligns with the inherent characteristics of physical field data. For example, in two-dimensional data, two orthogonal directions are used, maximizing data utilization and optimizing results without introducing redundancy, the proof can be found in A.2. Each direction is also multiplied by a scaling factor ζ, which controls the contribution of each direction. To account for the possibility that incorporating heterogeneous perturbations may introduce data artifacts, we establish an upper bound on ζ,

$$\zeta \left(\sum_{i=1}^{d} \omega_i \gamma_i(t) \right) < \frac{\pi}{6}, \tag{11}$$

where ω_i denotes the frequency along the i-th direction. The upper bound on ζ follows from the aggregate total variation across directions; see A.3 for the full derivation. The resulting spectral enhancement is illustrated in Figure 2: the Riesz integral kernel exhibits increased energy in high-frequency detail bands, mitigating the high-frequency issues reported during training (Xu et al., 2019).

Local orchestration. We introduce a direction-aware mixer in the Riesz domain that orchestrates global–local features across spectral scales. $\mathscr{F}_{\theta_{\text{CoordToRiesz}}}$ aggregates *global* Riesz-spectrum statistics, while directional branches capture *local* anisotropic variation; their fusion yields a monogenic-style representation that preserves coherence and fine directional detail, beyond magnitude-only operators. Details of the construction are provided in A.5. Data layout and formulas appear in Figure 1. In 1-D, Cartesian channels are not meaningful, wavelength conditions mixing to fuse forward/backward components (implementation details are in A.6).

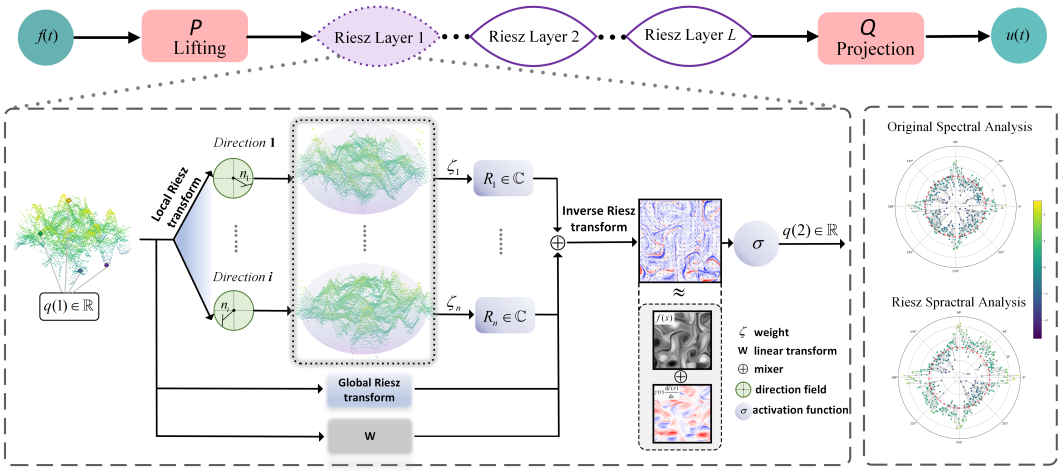

Figure 2: Model Overview of RNO. **Top:** The full RNO preserves the canonical neural-operator scaffold. **Bottom left:** Riesz layer 1. The input $q(1)$ is processed in parallel by (i) direction-wise local transforms, weighted by ζ, to extract orthogonal local features R; (ii) a global transform that stabilizes the representation; (iii) a linear map W. The mixed output passes through an inverse Riesz transform to reinforce spatial locality, followed by an activation σ to produce $q(2)$. **Bottom right:** Spectral comparison before and after the Riesz integral operator. Energy in the high-frequency band f (red circle denotes spectral mean) is evaluated under a fixed reference: larger magnitudes appear as outward expansion, smaller as inward contraction.

4 Experiments

Benchmark. RNO is evaluated on five diverse PDE benchmarks (Cao et al., 2024a)—Duffing, Beam, Diffusion, Reaction–Diffusion, and Brusselator(spanning 1-D 3-D, details in A.8). To assess performance on complex and abrupt dynamics, we further test 2-D Navier Stokes equations at Reynolds numbers 40, 500, and 5000 (Li et al., 2021), where higher Reynolds numbers induce richer rotational structure, probing sensitivity to directional capture. We then compare RNO with classical baselines on the real-world ERA5 dataset (Hersbach et al., 2020). These tasks constitute canonical PDE application scenarios and reflect practical challenges in solving such equations.

Baseline. We compare RNO with seven well-established neural network models: a classic model, U-Net(Ronneberger et al., 2015); two transformer-based operator learning models, Galerkin Transformer(Cao, 2021) and LSM(Wu et al., 2023); and five classical and powerful neural operator models: DeepOnet(Lu et al., 2021), FNO(Li et al., 2020a), F-FNO(Tran et al., 2023), ONO(Xiao et al., 2024) and LNO(Cao et al., 2024a). Each model excels in its respective domain. The comparison enables a comprehensive evaluation of RNO's effectiveness.

Implementation details. To ensure fairness, we fix evaluation metrics and training epochs across methods and use Adam (Kingma & Ba, 2014) optimizer. Metrics: relative ℓ_2 error for the general PDE and ERA5 benchmarks, and mean-squared error (MSE) for Navier Stokes equations. All neural operator baselines use a four-layer feature stack with sine activations. The learning rate is 10^{-3}. In RNO, the number of directions equals the data dimensionality. All experiments are implemented in PyTorch (Paszke et al., 2019) and run on 2 NVIDIA GTX 3090 GPUs.

4.1 Main Results

Performance across diverse PDEs. To assess generality, we evaluate RNO on five canonical PDE systems: Duffing, Beam, Diffusion, Reaction diffusion, and Brusselator, spanning mechanical, diffusive, and reactive dynamics. RNO attains the lowest prediction error on all benchmarks (Table 1); on Beam it reduces error by 51.6% relative to the second-best method, This substantial margin underscores RNO's ability to capture structural response behavior with high precision. Similarly, on the Duffing system, a classical non-linear oscillator known for its chaotic dynamics, RNO achieves a

Table 1: Test relative ℓ_2 error (↓) on five PDE benchmarks. Lower is better. The best (**bold**) and second-best (underline) results are highlighted. Promotions indicate relative gain of RNO over the second-best.

Model	Relative ℓ_2 (↓)				
	Duffing	Beam	Diffusion	Reaction–Diff.	Brusselator
U-NET(2015)	0.3835	0.0522	0.0212	0.1049	0.2557
GALERKIN(2021)	0.4239	0.0971	0.0319	0.1983	0.1883
DEEPONET(2021)	0.6289	0.9790	0.1356	0.7969	0.1910
FNO(2020a)	0.4536	0.0821	0.0229	0.1214	0.1827
F-FNO(2023)	0.2782	0.0515	0.0143	0.1003	0.1719
LSM(2023)	0.1699	0.0456	0.0145	0.0996	0.1628
LNO(2024a)	0.3325	0.0452	0.0081	0.1355	0.1858
ONO(2024)	0.3491	0.0519	0.0137	0.0989	0.1545
RNO	**0.1663**	**0.0219**	**0.0079**	**0.0899**	**0.1317**
Promotion	2.1%	51.6%	2.5%	10.3%	14.7%

* Top 4 ranking methods of PDE benchmarks: RNO (ours), LNO (2024), ONO (2024), LSM (2023).

2.1% improvement, indicating its capacity to model sensitive dependence on initial conditions and inherent nonlinearity effectively.

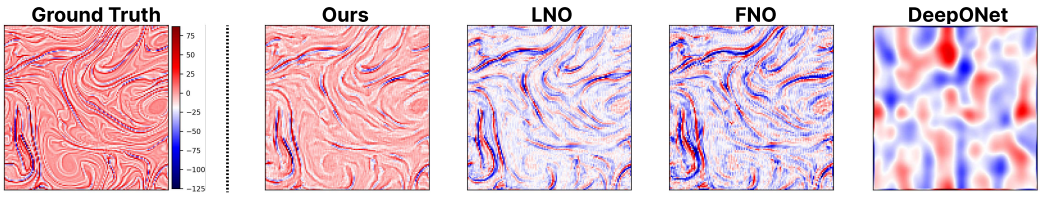

Figure 3: Comparison of high-frequency components with the Ground Truth.

On more diffusion-dominated tasks, such as Diffusion and Reaction-Diffusion(Reac-Diff.), RNO still outperforms prior approaches. While these systems are generally more stable and easier to approximate, RNO delivers 10.3% and 2.5% reductions in error, respectively, demonstrating its strength not only in complex non-linear settings but also in capturing fine-scale spatial dynamics in smoother regimes. Finally, for the Brusselator, a prototypical model for nonlinear chemical oscillations and pattern formation, RNO achieves a 14.7% reduction in relative error. This result is particularly significant as it highlights RNO's capability to handle stiff systems with intricate feedback mechanisms and spatiotemporal instabilities.

Performance on complex tasks. To evaluate RNO under highly non-linear dynamics, we train on two-dimensional incompressible Navier Stokes snapshots at three Reynolds numbers $Re \in \{40, 500, 5000\}$. As Re increases, the inertial range broadens and turbulent interactions intensify, rendering prediction progressively harder. Mean-squared error (MSE) for RNO, LNO, FNO and DeepONet is reported in Table 2. RNO achieves the lowest MSE across all Re; at $Re = 5000$ it attains MSE $= 0.9121$, substantially outperforming FNO and LNO. These results indicate that RNO's recurrent updates enhance stability as flow complexity grows. To assess fine-scale reconstruction, we isolate the high-frequency band of predicted velocity fields (Figure 3). RNO most faithfully recovers the high-wavenumber energy spectrum; LNO underestimates the spectral tail, whereas FNO exhibits ringing artefacts. DeepONet, however, produces a lot of noise in the

Table 2: Mean-squared error (↓) results of RNO, LNO, FNO, and DeeepONet models on NS datasets.

Datasets	DeepONet	FNO	LNO	RNO
$Re = 40$	0.0280	0.0078	0.0060	**0.0049**
$Re = 500$	3.4082	1.4251	1.2117	**0.4861**
$Re = 5000$	6.2721	2.9314	2.3139	**0.9121**

high-frequency components. These findings confirm that RNO captures subtle non-linear fluctuations more accurately than the alternatives.

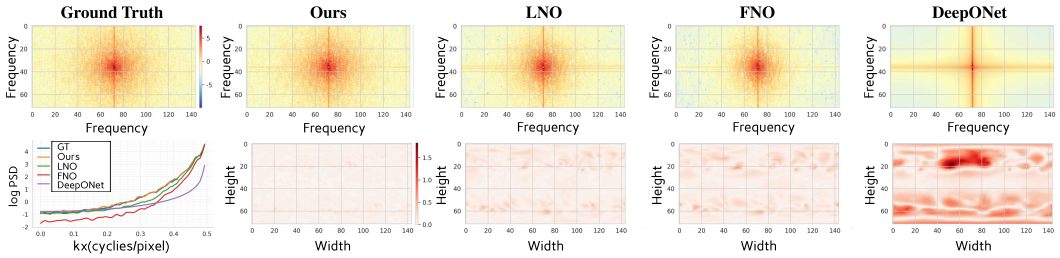

Figure 4: ERA5 benchmark at $t = 5$: Comparisons of 2-D PSD (top), spatial error (bottom right) and PSD along the x-axis for each method (bottom left).

We further evaluate on ERA5 using hourly 850-hPa geopotential-height fields (2012–2022 for training; 2023 for testing). As shown in Table 3, RNO significantly outperforms FNO , LNO, and DeepONet. RNO recovers both the inner low-frequency ring and the outer high-frequency core with smaller bias, as confirmed by PSD analysis (Figure 4). Across both synthetic turbulence and real-world weather, RNO preserves accuracy as dynamics intensify and faithfully reconstructs high-frequency content. It also transfers to operational-scale data without tuning, demonstrating its suitability for modeling and forecasting strongly non-linear spatiotemporal phenomena.

4.2 Model Analysis

Ablation study. To assess each module's contribution to the overall architecture, we evaluate four designs on two benchmarks (Duffing and Reaction–Diffusion): global-only (g) without mixer (o), local-only (l) without mixer, global+local (g+l) without mixer, and global+local (g+l) with mixer (w). From Table 4 we draw two clear observations: (i) on both datasets, either the global or the local module in isolation underperforms their combination; and (ii) the proposed mixer further facilitates effective information integration within the network, yielding additional gains.

Table 3: Relative ℓ_2 error (↓) results of RNO, LNO, FNO, and DeeepONet on ERA5.

Dataset	DeepONet	FNO	LNO	RNO
ERA5	0.0912	0.0093	0.0062	**0.0022**

Influence of orthogonalization. In practice, we set the number of directions equal to the data dimensionality to exploit the input's natural orthogonality. To isolate its effect, we vary the direction set and evaluate on three benchmarks (Beam, Reaction–Diffusion, and the Brusselator). We sweep the number of directions from 1 to 5, defining each by a uniform partition of the angular domain. We compare RNO with LNO, FNO, and DeepONet to quantify the role of orthogonal representations; in 2-D, two directions align with the native coordinate axes and are exactly orthogonal. Analogously in 3-D, the three canonical axes are orthogonal.

Table 4: Relative ℓ_2 error (↓) results of ablation study.

Datasets	$g(o)$	$l(o)$	g+l (o)	g+l (w)
Duffing	0.2098	0.2262	0.1801	**0.1663**
Reac–Diff.	0.0953	0.1189	0.0903	**0.0899**

As shown in Figure 5, RNO achieves its best performance with exactly two directions and remains competitive(typically second-best), when the number of directions is an integer multiple of the data dimension (e.g., 4). With too few directions (e.g., 1), we observe information loss and occasional instability; with too many (e.g., 3 or 5), performance does not improve because the additional, non-orthogonal directions are redundant. Even under suboptimal counts, RNO outperforms LNO, FNO, and DeepONet on most metrics. This highlights that orthogonality is not just a heuristic choice, but a principled design element in neural operators.

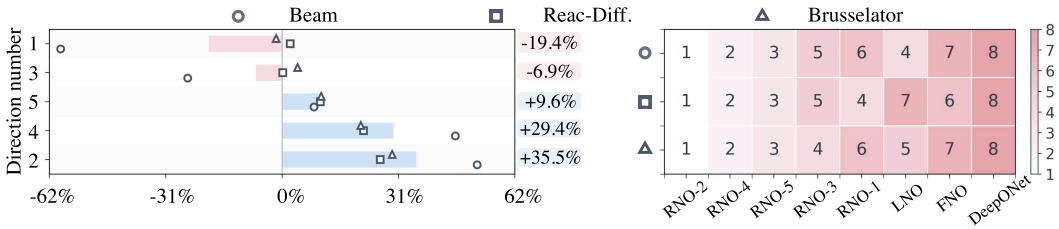

Figure 5: Comparison of relative ℓ_2 errors (left) and rank (right) for RNO, LNO, FNO, and DeepONet on the Beam, Reaction–Diffusion, and Brusselator benchmarks as the direction number varies from 1 to 5. RNO-n indicates that the RNO uses n directions.

Influence of activation function. In prior experiments, all methods employed the same sin activation to ensure parity. Prior work identifies activation functions as the primary source of nonlinearity in neural operators (Tran et al., 2023; Lu et al., 2021; Li et al., 2020b). To assess RNO's intrinsic capacity for non-linear phenomena, we benchmark it under multiple activations (sin, GELU, ReLU, Leaky ReLU, Sigmoid, Tanh, and none) on two representative tasks: Duffing and Reaction–Diffusion.

Table 5: Relative ℓ_2 error (↓) with different activation function on Duffing and Reac-Diffusion benchmarks. Lower is better. The best (**bold**) and second-best (underline) results are highlighted.

Dataset	Model	Relative ℓ_2 (↓)						
		sin	gelu	relu	leaky_relu	sigmoid	tanh	None
Duffing	**RNO**	**0.1663**	**0.1962**	**0.1441**	**0.1481**	**0.2089**	**0.1582**	**0.3215**
	LNO(2024a)	0.3325	0.4010	0.4527	0.4828	0.4211	0.3662	0.8121
	FNO(2020a)	0.4536	0.4725	0.4555	0.4612	0.4913	0.4343	0.9296
	DeepONet(2021)	0.6289	0.5972	0.4278	0.4717	0.9993	0.7042	0.9787
	Promotion	+50.0%	+51.1%	+68.2%	+67.9%	+50.4%	+56.8%	+60.4%
Reac–Diff.	**RNO**	**0.0899**	**0.1011**	**0.1027**	**0.0973**	0.1323	0.1482	**0.1017**
	LNO(2024a)	0.1355	0.1277	0.1068	0.1111	**0.1147**	**0.1069**	0.2915
	FNO(2020a)	0.1214	0.1443	0.1337	0.1500	0.1322	0.1424	0.3511
	DeepONet(2021)	0.7969	0.2742	0.2739	0.2754	1.0000	0.4337	0.6967
	Promotion	+25.9%	+20.8%	+3.8%	+12.4%	−15.3%	−38.6%	+65.1%

As shown in Table 5, on Duffing RNO attains the lowest error under every activation; ReLU performs best, while sin is slightly suboptimal. Even with no activation, RNO surpasses FNO, LNO, and DeepONet by a large margin, indicating a stronger intrinsic capacity for non-linear dynamics. On Reaction diffusion, the pattern differs: with Sigmoid or Tanh, RNO no longer exceeds FNO or LNO. Removing the activation improves RNO, approaching the optimum in Table 5. We attribute this behavior to redundant nonlinearities introduced by Sigmoid/Tanh, which exceed the data's representational needs and destabilize gradient propagation. In contrast, eliminating the activation severely degrades LNO, FNO, and DeepONet, underscoring RNO's superior capacity to internalize nonlinear representations.

Conclusion

We presented the Riesz Neural Operator (RNO), which bridges global spectral modeling with local derivative dynamics through Riesz transform. By jointly modeling local daymics (differentiation) and the operator theory (integration), we provide an integrated formulation that gives an intuitive explanation of how RNO enhances the interpretability. Evaluations on diverse PDEs, Navier–Stokes flows, and ERA5 data confirm consistent gains and robustness. Beyond accuracy, RNO demonstrates that embedding spectral derivatives into operator learning provides a natural and principled path toward more expressive models, suggesting broad potential for advancing scientific machine learning.

ETHICS STATEMENT

We adhere to the ICLR Code of Ethics (https://iclr.cc/public/CodeOfEthics). This work involves no human participants, personally identifiable information, or sensitive content.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of benchmarks in A.8. Experimental settings are summarized in 4. The datasets used in this study are publicly available: the diverse PDE benchmarks at (https://github.com/qianyingcao/Laplace-Neural-Operator); Navier–Stokes with $Re \in \{40, 500, 5000\}$ at (https://github.com/neuraloperator/neuraloperator); and ERA5 at (https://cds.climate.copernicus.eu/). All results are averages over 3 runs with different random seeds. Definitions of the evaluation metrics are provided in A.7.

REFERENCES

Richard B Alley, Jochem Marotzke, William D Nordhaus, Jonathan T Overpeck, Dorothy M Peteet, Roger A Pielke Jr, Raymond T Pierrehumbert, Peter B Rhines, Thomas F Stocker, Lynne D Talley, et al. Abrupt climate change. *science*, 299(5615):2005–2010, 2003.

Luca Barberi and Karsten Kruse. Localized states in active fluids. *Physical Review Letters*, 131(23): 238401, 2023.

Andrés Bruhn, Joachim Weickert, and Christoph Schnörr. Lucas/kanade meets horn/schunck: Combining local and global optic flow methods. *International journal of computer vision*, 61:211–231, 2005.

Qianying Cao, Somdatta Goswami, and George Em Karniadakis. Laplace neural operator for solving differential equations. *Nature Machine Intelligence*, 6(6):631–640, 2024a.

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

Shuhao Cao, Francesco Brarda, Ruipeng Li, and Yuanzhe Xi. Spectral-refiner: Accurate fine-tuning of spatiotemporal fourier neural operator for turbulent flows. *arXiv preprint arXiv:2405.17211*, 2024b.

Ze Cheng, Zhuoyu Li, Xiaoqiang Wang, Jianing Huang, Zhizhou Zhang, Zhongkai Hao, and Hang Su. Accelerating pde-constrained optimization by the derivative of neural operators. *arXiv preprint arXiv:2506.13120*, 2025.

Vaclav Cizek. Discrete hilbert transform. *IEEE Transactions on Audio and Electroacoustics*, 18(4): 340–343, 1970.

JoséL Rubio de Francia, Francisco J Ruiz, and JoséL Torrea. Calderón-zygmund theory for operator-valued kernels. *Advances in Mathematics*, 62(1):7–48, 1986.

Robert DiPietro and Gregory D Hager. Deep learning: Rnns and lstm. In *Handbook of medical image computing and computer assisted intervention*, pp. 503–519. Elsevier, 2020.

Jianghang Gu, Ling Wen, Yuntian Chen, and Shiyi Chen. An explainable operator approximation framework under the guideline of green's function. *Journal of Computational Physics*, pp. 114244, 2025.

Massimo Guiggiani and Paolo Casalini. Direct computation of cauchy principal value integrals in advanced boundary elements. *International Journal for Numerical Methods in Engineering*, 24(9): 1711–1720, 1987.

Maximilian Herde, Bogdan Raonic, Tobias Rohner, Roger Käppeli, Roberto Molinaro, Emmanuel de Bézenac, and Siddhartha Mishra. Poseidon: Efficient foundation models for pdes. *Advances in Neural Information Processing Systems*, 37:72525–72624, 2024.

Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, et al. The era5 global reanalysis. *Quarterly journal of the royal meteorological society*, 146(730):1999–2049, 2020.

Hossein Hodaei, Absar U Hassan, Steffen Wittek, Hipolito Garcia-Gracia, Ramy El-Ganainy, Demetrios N Christodoulides, and Mercedeh Khajavikhan. Enhanced sensitivity at higher-order exceptional points. *Nature*, 548(7666):187–191, 2017.

RP Kanwal and KC Liu. A taylor expansion approach for solving integral equations. *International Journal of Mathematical Education in Science and Technology*, 20(3):411–414, 1989.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Thorsten Kurth, Shashank Subramanian, Peter Harrington, Jaideep Pathak, Morteza Mardani, David Hall, Andrea Miele, Karthik Kashinath, and Anima Anandkumar. Fourcastnet: Accelerating global high-resolution weather forecasting using adaptive fourier neural operators. In *Proceedings of the platform for advanced scientific computing conference*, pp. 1–11, 2023.

Samuel Lanthaler, Zongyi Li, and Andrew M Stuart. Nonlocality and nonlinearity implies universality in operator learning. *arXiv preprint arXiv:2304.13221*, 2023.

Zhihao Li, Haoze Song, Di Xiao, Zhilu Lai, and Wei Wang. Harnessing scale and physics: A multi-graph neural operator framework for pdes on arbitrary geometries. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 1*, pp. 729–740, 2025.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *arXiv preprint arXiv:2010.08895*, 2020a.

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*, 2020b.

Zongyi Li, Miguel Liu-Schiaffini, Nikola Kovachki, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Learning dissipative dynamics in chaotic systems. *arXiv preprint arXiv:2106.06898*, 2021.

Zongyi Li, Nikola Kovachki, Chris Choy, Boyi Li, Jean Kossaifi, Shourya Otta, Mohammad Amin Nabian, Maximilian Stadler, Christian Hundt, Kamyar Azizzadenesheli, et al. Geometry-informed neural operator for large-scale 3d pdes. *Advances in Neural Information Processing Systems*, 36: 35836–35854, 2023.

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. *ACM/JMS Journal of Data Science*, 1(3):1–27, 2024.

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.

Lawrence Marple. Computing the discrete-time" analytic" signal via fft. *IEEE Transactions on signal processing*, 47(9):2600–2603, 1999.

Sophocles J Orfanidis. *Introduction to signal processing*. Prentice-Hall, Inc., 1995.

Thomas J Osler. Leibniz rule for fractional derivatives generalized and an application to infinite series. *SIAM Journal on Applied Mathematics*, 18(3):658–674, 1970.

HB Pacejka and IJM Besselink. Magic formula tyre model with transient properties. *Vehicle system dynamics*, 27(S1):234–249, 1997.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

Yuan Qiu, Nolan Bridges, and Peng Chen. Derivative-enhanced deep operator network. *Advances in Neural Information Processing Systems*, 37:20945–20981, 2024.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. 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, 2019.

Bogdan Raonic, Roberto Molinaro, Tim De Ryck, Tobias Rohner, Francesca Bartolucci, Rima Alaifari, Siddhartha Mishra, and Emmanuel de Bézenac. Convolutional neural operators for robust and accurate learning of pdes. *Advances in Neural Information Processing Systems*, 36: 77187–77200, 2023.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pp. 234–241. Springer, 2015.

Jonathan D Smith, Kamyar Azizzadenesheli, and Zachary E Ross. Eikonet: Solving the eikonal equation with deep neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 59 (12):10685–10696, 2020.

Atsushi Tamada and Michihiro Igarashi. Revealing chiral cell motility by 3d riesz transform-differential interference contrast microscopy and computational kinematic analysis. *Nature Communications*, 8(1):2194, 2017.

Karn Tiwari, Niladri Dutta, NM Krishnan, et al. Latent mamba operator for partial differential equations. *arXiv preprint arXiv:2505.19105*, 2025a.

Karn Tiwari, NM Krishnan, et al. Cono: Complex neural operator for continous dynamical physical systems. *APL Machine Learning*, 3(2), 2025b.

Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized fourier neural operators. *arXiv preprint arXiv:2111.13802*, 2021.

Alasdair Tran, Alexander Mathews, Lexing Xie, and Cheng Soon Ong. Factorized fourier neural operators. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tmIiMPl4IPa.

Tapas Tripura and Souvik Chakraborty. Wavelet neural operator: a neural operator for parametric partial differential equations. *arXiv preprint arXiv:2205.02191*, 2022.

Michael Unser and Dimitri Van De Ville. Wavelet steerability and the higher-order riesz transform. *IEEE Transactions on Image Processing*, 19(3):636–652, 2009.

Michael Unser, Daniel Sage, and Dimitri Van De Ville. Multiresolution monogenic signal analysis using the riesz–laplace wavelet transform. *IEEE Transactions on Image Processing*, 18(11): 2402–2418, 2009.

Neal Wadhwa, Michael Rubinstein, Frédo Durand, and William T Freeman. Riesz pyramids for fast phase-based video magnification. In *2014 IEEE International Conference on Computational Photography (ICCP)*, pp. 1–10. IEEE, 2014.

Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, et al. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, 2023.

Rui Wang, Yihe Dong, Sercan Ö Arik, and Rose Yu. Koopman neural forecaster for time series with temporal distribution shifts. *arXiv preprint arXiv:2210.03675*, 2022.

Steven Weinberg. Eikonal method in magnetohydrodynamics. *Physical Review*, 126(6):1899, 1962.

Haixu Wu, Tengge Hu, Huakun Luo, Jianmin Wang, and Mingsheng Long. Solving high-dimensional pdes with latent spectral models. *arXiv preprint arXiv:2301.12664*, 2023.

Zipeng Xiao, Zhongkai Hao, Bokai Lin, Zhijie Deng, and Hang Su. Improved operator learning by orthogonal attention. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 54288–54299, 2024.

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.

A Appendix

A.1 Why local direction is important for PDEs?

Local derivatives are inherently directional, with directionality encoded by the phase gradient $\nabla\phi$. In the high-frequency regime, PDEs couple to the data through their principal symbol evaluated at $\nabla\phi$. The Riesz transform provides rotation-equivariant estimates of local phase and orientation, thereby supplying neural networks with precisely the quantities that govern PDE propagation. We next analyze each step in detail.

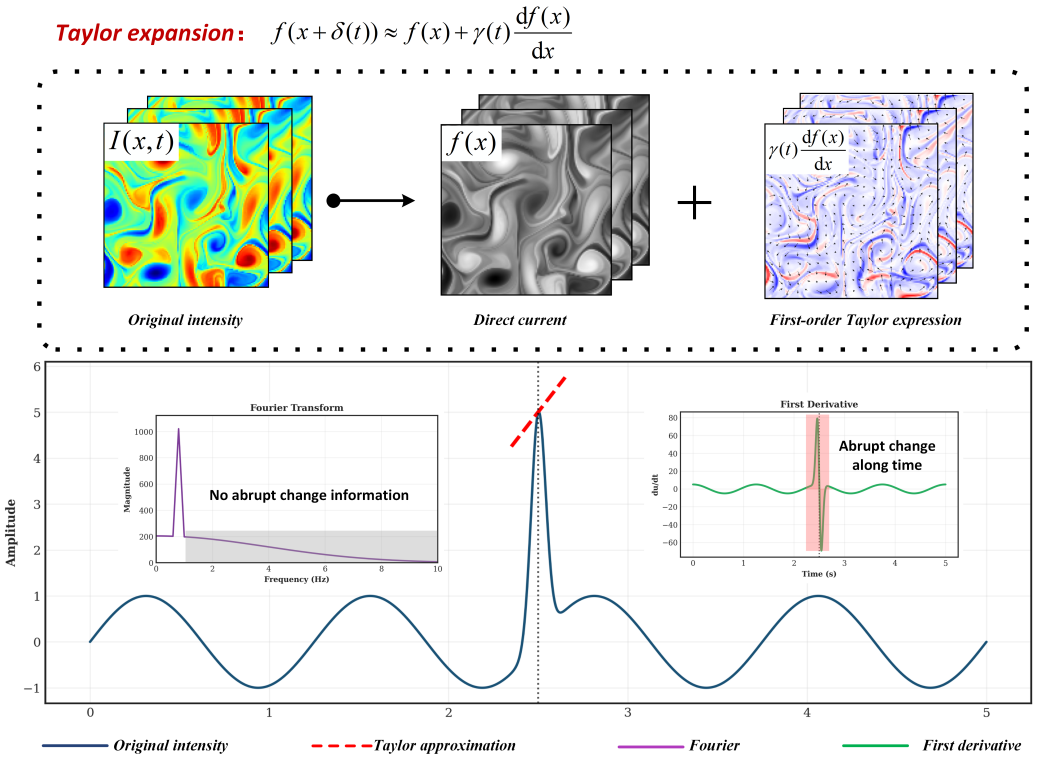

Figure 6: The specific structure of Taylor expansion and its difference from Fourier transform.

1. Local derivatives are directional

Given $u : \mathbb{R}^d \to \mathbb{C}$, the directional derivative along a unit vector $\mathbf{n}$ is

$$\partial_{\mathbf{n}} u(\mathbf{x}) = \mathbf{n} \cdot \nabla u(\mathbf{x}).$$

Write $u = Ae^{i\phi}$ with $A \geq 0$, $\phi \in \mathbb{R}$ wherever $u \neq 0$. Then

$$\partial_{\mathbf{n}} u = e^{i\phi}(\partial_{\mathbf{n}} A + iA\,\partial_{\mathbf{n}}\phi) = u\,(\partial_{\mathbf{n}} \ln A + i\,\partial_{\mathbf{n}}\phi)\,. \tag{12}$$

Consequently, the normalized directional derivative decomposes neatly into amplitude and phase components:

$$\partial_{\mathbf{n}}\phi = \mathrm{Im}\Big(\frac{\partial_{\mathbf{n}} u}{u}\Big), \qquad \partial_{\mathbf{n}} \ln A = \mathrm{Re}\Big(\frac{\partial_{\mathbf{n}} u}{u}\Big)\,. \tag{13}$$

Accordingly, local derivatives are inherently orientation dependent, with $\nabla\phi$ prescribing that orientation.

2. Directionality is encoded by the phase

From $\nabla u = u(\nabla \ln A + i\,\nabla\phi)$ we obtain

$$\nabla\phi = \mathrm{Im}\Big(\frac{\nabla u}{u}\Big)\,, \qquad \nabla \ln A = \mathrm{Re}\Big(\frac{\nabla u}{u}\Big)\,. \tag{14}$$

The vector $\mathbf{k}(\mathbf{x}) \coloneqq \nabla\phi(\mathbf{x})$ is the local wavevector and

$$\mathbf{n}(\mathbf{x}) \coloneqq \frac{\nabla\phi(\mathbf{x})}{\|\nabla\phi(\mathbf{x})\|}$$

is the *local orientation*. In the high-frequency regime, where amplitude varies slowly compared with phase,

$$\|\nabla u\| \approx A\,\|\nabla\phi\|, \quad \partial_{\mathbf{n}} u \approx iA\,(\partial_{\mathbf{n}}\phi)\,e^{i\phi},$$

so the derivative is dominated by directional phase change.

3. Why PDEs care: principal symbol and high-frequency link

Consider an m-th order linear differential operator

$$P(x,\partial) = \sum_{|\alpha|\le m} a_\alpha(x)\,\partial^\alpha, \qquad p_{\mathrm{pr}}(x,\xi) = \sum_{|\alpha|=m} a_\alpha(x)\,(i\xi)^\alpha$$

with principal symbol p_{pr}. Insert a ansatz $u(x) = A(x)e^{i\omega\phi(x)}$ ($\omega \gg 1$). By repeated Leibniz(Osler, 1970),

$$P\big(Ae^{i\omega\phi}\big) = e^{i\omega\phi}\Big[\omega^m\, p_{\mathrm{pr}}\big(x,\nabla\phi\big)\,A \;+\; \mathcal{O}(\omega^{m-1})\Big]. \tag{15}$$

At leading order, the principal symbol ties the dynamics to the phase gradient $\nabla\phi$. Asymptotic solvability at order ω^m entails the eikonal equation(Smith et al., 2020),

$$p_{\mathrm{pr}}\big(x,\nabla\phi(x)\big) = 0\,. \tag{16}$$

Hence, rays propagate along the phase gradient $\nabla\phi$. For the transport operator $P = \partial_t + \mathbf{a}(x)\cdot\nabla_x$, the principal symbol is $p_{\mathrm{pr}}(x,\tau,\xi) = i\big(\tau + \mathbf{a}\cdot\xi\big)$. Imposing the eikonal condition (Weinberg, 1962) $p_{\mathrm{pr}}(x,\partial_t\phi,\nabla_x\phi) = 0$ gives $\tau + \mathbf{a}\cdot\xi = 0$ with $\xi = \nabla_x\phi$, so the spatial propagation direction is aligned with $\nabla_x\phi$.

4. Riesz transform: extracting phase and orientation

The d-D Riesz transform $\mathcal{R}u = (\mathcal{R}_1 u, \ldots, \mathcal{R}_d u)$ is defined by

$$\widehat{\mathcal{R}_j u}(\omega) = -i\,\frac{\omega_j}{\|\omega\|}\,\hat{u}(\omega).$$

Plane wave benchmark. For $u(x) = \cos(\mathbf{k}\cdot x)$,

$$\mathcal{R}u(x) = \sin(\mathbf{k}\cdot x)\,\frac{\mathbf{k}}{\|\mathbf{k}\|}.$$

Hence $\mathcal{R}u$ points in the wave direction $\mathbf{k}/\|\mathbf{k}\|$ and is $\pi/2$ out of phase.

High frequency. Let $u(x) = A(x)\cos\big(\omega\,\phi(x)\big)$ with smooth A, ϕ and $\omega \gg 1$. Locally near x_0, $\phi(x) \approx \phi(x_0) + \nabla\phi(x_0)\cdot(x - x_0)$, i.e. a plane wave with $\mathbf{k} = \omega\nabla\phi(x_0)$. Then

$$\mathcal{R}u(x_0) = A(x_0)\,\sin\big(\omega\phi(x_0)\big)\,\frac{\nabla\phi(x_0)}{\|\nabla\phi(x_0)\|} \;+\; \mathcal{O}\!\left(\frac{\|\nabla A\| + \|\nabla^2\phi\|}{\omega}\right). \tag{17}$$

Therefore the orientation is

$$\mathbf{n}(x) = \frac{\mathcal{R}u(x)}{\|\mathcal{R}u(x)\|} \;\approx\; \frac{\nabla\phi(x)}{\|\nabla\phi(x)\|}\,,$$

and a rotation-invariant *local phase* can be defined by the monogenic signal (Unser et al., 2009),

$$\rho(x) = \sqrt{u(x)^2 + \|\mathcal{R}u(x)\|^2}, \qquad \theta(x) = \operatorname{atan2}\big(\|\mathcal{R}u(x)\|,\; u(x)\big),$$

so that locally $u \approx \rho\cos\theta$, $\|\mathcal{R}u\| \approx \rho\sin\theta$.

5. The rationality of using Riesz transformation for neural operator

- **Aligns with principal symbols.** By equation 15–equation 16, the leading-order response of the PDE depends on $\nabla\phi$ through $p_{\mathrm{pr}}(x, \nabla\phi)$. Riesz features extract the orientation $\mathbf{n} \approx \nabla\phi/\|\nabla\phi\|$ and the quadrature phase θ, thereby granting the network direct access to the directional variables relevant to the PDE.
- **Rotation covariance.** Since $\mathcal{R}$ is rotation-covariant, $(u, \mathcal{R}u)$ provides features that transform predictably under rotations. This property is essential for modeling anisotropic diffusion, advection, and wave propagation without relearning all orientations.
- **Phase emphasis and contrast robustness.** Phase (via θ) remains invariant under multiplicative contrast changes $u \mapsto c\,u$, while the derivatives in equation 13 accentuate $\partial_{\mathbf{n}}\phi$. Consequently, learning is guided toward geometry and propagation rather than raw amplitude.
- **Differentiable and lightweight.** $\mathcal{R}$ is linear with a fixed multiplier $-i\xi/\|\xi\|$, and its discrete realizations are convolutional and backpropagation-friendly. It can serve either as a fixed front-end or as a parallel stream complementing learned filters.

6. Influence of phase and orientation

To highlight how phase governs directionality, we conduct an ablation on the Navier–Stokes dataset. Under an identical network architecture, we compare two input parameterizations (cf. Figure 7(a)): (i) complex features that retain both magnitude and phase, $A(x)e^{i\phi(x)}$; and (ii) magnitude-only features with the phase suppressed, $A(x)e^{i*0}$. All other training and evaluation settings are held fixed.

As shown in Figure 7(b)-(d), removing phase yields markedly sparser fine-scale structures and substantial misalignment of local feature orientations. In rollout regimes (Cao et al., 2024b) (e.g., NO), such local directional errors accumulate over time. The effect is amplified for more complex data with more numerous localized structures, resulting in larger long-horizon errors, as evidenced by Table 2. Moreover, the Riesz transform precisely enhances the directional sensitivity of phase control.

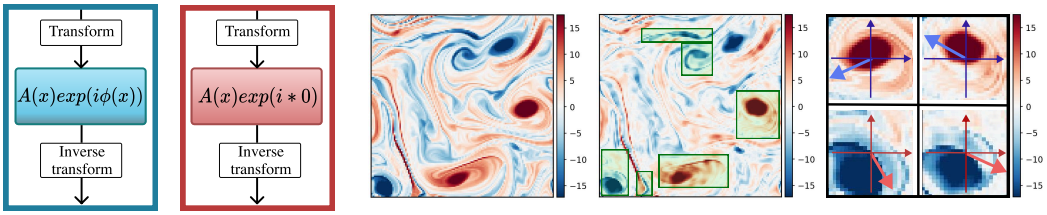

Figure 7: (a) Two feature structures; (b) results using magnitude–phase features; (c) results using magnitude-only features; (d) orientation comparison on selected details.

A.2 Orthogonality

Inner product. Let $n \geq 2$ and wave-vector $\mathbf{k} = (k_1, \ldots, k_n) \in \mathbb{R}^n$ and magnitude $\|\mathbf{k}\| = (\sum_{i=1}^n k_i^2)^{1/2}$. For any distinct indices $i \neq j$ consider the L_2 inner product

$$\langle R_i, R_j \rangle \;=\; \int_{\mathbb{R}^n} R_i(\mathbf{k})\,\overline{R_j(\mathbf{k})}\;d\mathbf{k}. \tag{18}$$

Then setting $E(\mathbf{k}) := |F(\mathbf{k})|^2$ (an even function in every k_ℓ) yields

$$\langle R_i, R_j \rangle \;=\; \int_{\mathbb{R}^n} \frac{k_i k_j}{\|\mathbf{k}\|^2}\,E(\mathbf{k})\;d\mathbf{k}.$$

Odd–even symmetry argument. Fix $i \neq j$. Under the sign change $k_i \mapsto -k_i$ the factor $k_i k_j$ flips sign while both $\|\mathbf{k}\|^2$ and $E(\mathbf{k})$ remain unchanged; hence the integrand is *odd* with respect to the k_i axis. Because the integration domain $\mathbb{R}^n$ is symmetric, the integral vanishes:

$$\langle \mathcal{R}_i, \mathcal{R}_j \rangle = 0, \qquad \forall\, i \neq j \tag{19}$$

Spatial-domain equivalence. By Parseval's theorem (Orfanidis, 1995), the frequency-domain result implies spatial orthogonality: $\langle \mathcal{R}_i f, \mathcal{R}_j f \rangle_{\mathbf{x}} = 0$ for $i \neq j$, where $\mathcal{R}_i$ denotes the real-space Riesz transform in direction i. Thus the vector of Riesz components $\boldsymbol{\mathcal{R}} f = (\mathcal{R}_1 f, \ldots, \mathcal{R}_d f)$ forms an orthonormal frame in $L_2(\mathbb{R}^n)$. Consequently, neural operators that process the Riesz vector component-wise inherit stable, non-redundant latent representations in any spatial dimension d.

A.3 Bound

First-order approximation. The first-order Taylor expansion provides an approximation to the actual intensity function $f(x)$ and its variations, which is reflected in the Riesz transform within the spatial domain (Wadhwa et al., 2014). For rapidly varying data, the accuracy of the first-order Taylor expansion decreases as ζ increases. Large values of $\zeta\gamma(t)$ may introduce artifacts. To ensure the approximation remains valid, we derive an upper bound for ζ.

Assuming the first-order Taylor expansion holds, the modified intensity is:

$$f(x + \zeta\gamma(t)) \approx f(x) + \zeta\gamma(t)\frac{df(x)}{dx}. \tag{20}$$

The Riesz transform introduces a 90-degree phase shift. If $f(x) = \cos(\omega x)$, we have:

$$\cos(\omega x + \zeta\omega\gamma(t)) \approx \cos(\omega x) - \zeta\omega\gamma(t)\sin(\omega x). \tag{21}$$

Using the trigonometric identity:

$$\cos(\omega x + \zeta\omega\gamma(t)) = \cos(\omega x)\cos(\zeta\omega\gamma(t)) - \sin(\omega x)\sin(\zeta\omega\gamma(t)), \tag{22}$$

we require $\zeta\omega\gamma(t)$ to be small for $\cos(\zeta\omega\gamma(t)) \approx 1$ and $\sin(\zeta\omega\gamma(t)) \approx \zeta\omega\gamma(t)$. Assuming a 5% error tolerance, the second-order term should be less than 5% of the first-order term, yielding the condition:

$$\zeta\gamma(t) < \frac{\lambda}{12}, \tag{23}$$

where $\lambda = \frac{2\pi}{\omega}$ is the spatial wavelength. A larger wavelength allows for a larger scaling factor ζ; smaller wavelengths require a smaller ζ to avoid artifacts. While this bound on ζ is theoretical, the wavelength of the data must be determined based on the signal and network's characteristics.

Multidimensional case. For a multi-dimensional setting, let $x = (x_1, x_2, \ldots, x_d)$ represent the spatial coordinates. The first-order Taylor expansion of $f(x)$ becomes:

$$f(x + \zeta\gamma(t)) \approx f(x) + \zeta\sum_{i=1}^{d}\gamma_i(t)\frac{\partial f(x)}{\partial x_i}. \tag{24}$$

Here, $\gamma_i(t)$ is the scaling factor in the i-th direction. The scaling factor ζ still amplifies the signal's variations across all dimensions. The error condition becomes:

$$\zeta\left(\sum_{i=1}^{d}\omega_i\gamma_i(t)\right) < \frac{\pi}{6}, \tag{25}$$

where ω_i is the frequency in the i-th direction. The bound provided here is not an exact limit but rather a theoretical upper bound. Since it depends on the angular frequency of the data, the boundary is not fixed. Accordingly, in the mixer A.5 we replace it with a learnable parameter while constraining its range.

A.4 Relationship between the Hilbert and Riesz transforms

In one dimension, the Hilbert transform $\mathcal{H}f$ (Cizek, 1970) is the singular integral

$$\mathcal{H}f(x) = \frac{1}{\pi}\,\text{p.v.}\int_{-\infty}^{\infty}\frac{f(y)}{x-y}\,dy,$$

with Fourier representation

$$\mathcal{F}[\mathcal{H}f](\xi) = -i\,\text{sgn}(\xi)\,\hat{f}(\xi),$$

where "p.v." denotes the Cauchy principal value (Guiggiani & Casalini, 1987). In $\mathbb{R}^n$, the Riesz transform generalizes this construction: for $f \in \mathcal{S}(\mathbb{R}^n)$,

$$\mathcal{R}_j f(x) = \mathcal{F}^{-1}\left[-i\,\frac{\xi_j}{|\xi|}\,\hat{f}(\xi)\right](x), \qquad j = 1, \ldots, n,$$

so each $\mathcal{R}_j$ acts as a normalized directional derivative with a degree-zero Fourier multiplier.

For $n = 1$, taking $\xi_1 = \xi$ yields

$$\mathcal{R}_1 f(x) = \mathcal{F}^{-1}\left[-i\,\frac{\xi}{|\xi|}\,\hat{f}(\xi)\right](x) = \mathcal{F}^{-1}\left[-i\,\mathrm{sgn}(\xi)\,\hat{f}(\xi)\right](x),$$

recovering exactly the Hilbert transform: $\mathcal{R}_1 f = \mathcal{H} f$.

Collecting the components $\boldsymbol{\mathcal{R}} f = (\mathcal{R}_1 f, \ldots, \mathcal{R}_n f)$ gives a vector operator that is an L^2-isometry:

$$\sum_{j=1}^{n} \|\mathcal{R}_j f\|_{L^2}^2 = \|f\|_{L^2}^2.$$

Thus the Riesz transform is the canonical multi-dimensional analogue of the Hilbert transform, widely used to extract directional and phase information in higher-dimensional signal analysis. Making this connection explicit sharpens our understanding of how Riesz transforms operate on derivative-like quantities and underpins the RNO framework.

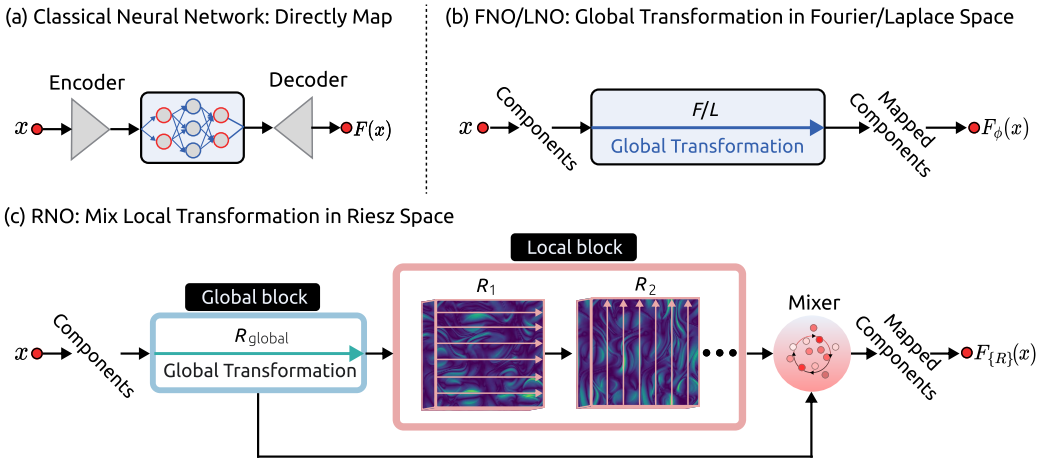

Figure 8: A comparison between RNO and simplified versions of some classical architectures, highlighting the global-local mixer of RNO.

A.5 Implementation of the direction mixer

The direction mixer essentially performs global–local integration of physical dynamics in the Riesz space. This differs from mixing in the spatial domain, since data representations in the Riesz space are inherently distinct, and can be regarded as a latent feature representation. Consequently, the challenge is to design a fusion mechanism that preserves data characteristics while naturally integrating physical properties. In the physical domain, the monogenic signal (Unser et al., 2009), a high-dimensional extension of analytic signals, offers such an example, as it naturally fuses orthogonal components across scales. For a monogenic signal, its general form in $\mathbb{R}^3$ can be expressed as

$$f(x) = x_1 + x_2 i + x_3 j. \tag{26}$$

Our implementation follows this formulation with several modifications:

$$Mixer = R_{\text{global}} + w_i R_i + w_j R_j, \tag{27}$$

where w denotes learnable, direction-specific parameters that assign spectral weights to the corresponding Riesz components. For n-dimensional data, the formulation naturally extends to n dimensions. To avoid excessive artifacts according to the derived bounds, the values are constrained within $(0, 2)$ according to bound in A.3. This structure enables a natural fusion of global and local information, which is key to the superior performance of the overall model.

A.6 Directionality in one dimension

In one spatial dimension, the Riesz transform reduces to the Hilbert transform (Cizek, 1970), the canonical operator for building analytic signals (Marple, 1999):

$$\mathcal{H}\{f\}(t) = \frac{1}{\pi}\,\mathrm{p.v.}\int_{-\infty}^{\infty} \frac{f(\tau)}{t-\tau}\,d\tau. \tag{28}$$

While a single spatial dimension lacks an intrinsic notion of orientation, the Hilbert transform nonetheless decomposes any real-valued signal into two oppositely directed components, distinguished by the sign of their temporal frequencies. Define

$$f_+(t) = \tfrac{1}{2}\big[f(t) + i\,\mathcal{H}\{f\}(t)\big], \tag{29}$$

$$f_-(t) = \tfrac{1}{2}\big[f(t) - i\,\mathcal{H}\{f\}(t)\big], \tag{30}$$

so that positive frequencies correspond to energy (or information) propagating away from the source, while negative frequencies correspond to energy returning toward it. The spectral imbalance between these two sectors thus reveals the net propagation direction. In fluid dynamics, spectra dominated by positive frequencies typically indicate pressure waves radiating from a point source. When solutions of the Navier–Stokes equations predict strong vortical feedback, analysis should instead focus on the negative-frequency band. In contrast, in purely diffusive processes where concentrations relax from high to low, the positive frequency component typically dominates.

A.7 Evaluation metrics

We evaluate model performance using two standard metrics in our work: the relative ℓ_2 error and the mean squared error (MSE). The relative ℓ_2 error measures the discrepancy normalized by the ground-truth energy and is defined as

$$\text{relative } \ell_2 \text{ error} = \frac{\sum_{i=1}^{M}\big(Y_i - \hat{Y}_i\big)^2}{\sum_{i=1}^{M}\big(Y_i\big)^2} = \frac{\|Y-\hat{Y}\|_2^2}{\|Y\|_2^2}, \tag{31}$$

The MSE quantifies the average squared prediction error:

$$\text{MSE} = \frac{1}{M}\sum_{i=1}^{M}\big(Y_i - \hat{Y}_i\big)^2. \tag{32}$$

Here Y_i denotes the ground-truth value, $\hat{Y}_i$ the model prediction, and M the number of samples/points.

A.8 Supplementary for benchmarks

A.8.1 Duffing Oscillator

A paradigmatic non-linear, damped, and driven oscillator, the Duffing system is governed by

$$m\ddot{x}(t) + c\dot{x}(t) + k_1 x(t) + k_3 x^3(t) = f(t), \tag{33}$$

where $x(t)$ is the displacement, $\dot{x}(t)$ and $\ddot{x}(t)$ denote velocity and acceleration, and the constants m, c, k_1, and k_3 represent the mass, viscous damping, linear stiffness, and cubic stiffness, respectively. The term $f(t)$ supplies the external forcing.

A.8.2 Euler–Bernoulli Beam

For a slender Euler–Bernoulli beam subject to transverse loading, the Euler–Lagrange formalism yields

$$EI\,\frac{\partial^4 y(x,t)}{\partial x^4} + \rho A\,\frac{\partial^2 y(x,t)}{\partial t^2} = f(x,t), \tag{34}$$

with $y(x,t)$ denoting the transverse deflection and $f(x,t)$ the applied load. Material and geometric properties enter through Young's modulus E, second moment of area I, density ρ, and cross-sectional area A.

A.8.3 DIFFUSION EQUATION

Pure diffusion in one dimension obeys

$$D\,\frac{\partial^2 y(x,t)}{\partial x^2} - \frac{\partial y(x,t)}{\partial t} = f(x,t), \tag{35}$$

where $y(x,t)$ is the scalar field, D the (constant) diffusion coefficient, and $f(x,t)$ a distributed source or sink.

A.8.4 REACTION–DIFFUSION SYSTEM

Coupling diffusion with non-linear kinetics leads to the generic reaction–diffusion form

$$D\,\frac{\partial^2 y(x,t)}{\partial x^2} + k\,y^2(x,t) - \frac{\partial y(x,t)}{\partial t} = f(x,t), \tag{36}$$

where k sets the quadratic reaction rate while $y(x,t)$ and $f(x,t)$ retain their usual interpretations.

A.8.5 BRUSSELATOR REACTION–DIFFUSION SYSTEM

The two-species Brusselator, describing an autocatalytic chemical network, is written as

$$\begin{aligned}\frac{\partial u}{\partial t} &= D_0\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + a + f(t) - (1+b)u + vu^2,\\ \frac{\partial v}{\partial t} &= D_1\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) + bu - vu^2,\end{aligned} \tag{37}$$

defined for $(x,y) \in [0,1]^2$ and $t \in [0,20]$. Here u and v are concentration fields, D_0 and D_1 their diffusion coefficients, and a, b the kinetic parameters, with $f(t)$ introducing a time-dependent feed term.

A.8.6 NAVIER–STOKES EQUATION

For an incompressible Newtonian fluid with velocity $\mathbf{u}(\mathbf{x},t)$ and pressure $p(\mathbf{x},t)$,

$$\partial_t \mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla p + \nu\Delta\mathbf{u} + \mathbf{f}, \qquad \nabla\cdot\mathbf{u} = 0,$$

where ν denotes the kinematic viscosity and $\mathbf{f}$ a body force. Upon nondimensionalization using the characteristic speed U and length L,

$$\partial_t \mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} = -\nabla p + \frac{1}{Re}\Delta\mathbf{u} + \mathbf{f}, \qquad \nabla\cdot\mathbf{u} = 0,$$

the sole control parameter is the Reynolds number $Re = UL/\nu$, which quantifies the ratio of inertial to viscous effects. Higher Reynolds numbers are accompanied by increasingly complex local structures, which is one of the challenges faced by current neural-network methods. Accordingly, using configurations spanning a range of Reynolds numbers in this experiment to validate the model's capacity to represent local nonlinearities is both reasonable and reliable. As shown in Figure 9, RNO retains excellent predictive performance despite increasing local complexity.

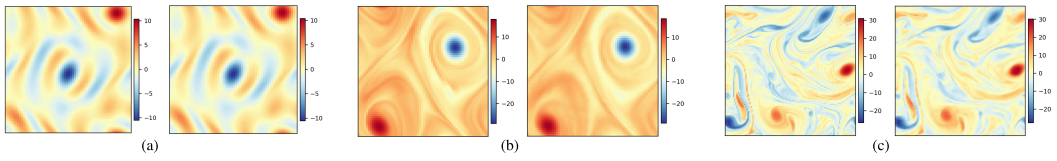

Figure 9: Comparison between the ground truth and RNO predictions across different Reynolds numbers. (a) $Re = 40$, (b) $Re = 500$, (c) $Re = 5000$.

DISCUSSION AND LIMINATION

RNO's key insight is a scale-wise repartitioning of the data. This reallocation expands the model's representational degrees of freedom. In neural-operator architectures, it provides a principled mechanism to surface local components that would otherwise be suppressed. Although these local components may spatially overlap with global features, such overlap does not degrade performance, as the original signals at those locations were underweighted. RNO explicitly amplifies these neglected local contributions and rebalances the component weights. From a Taylor-expansion perspective, these local terms admit further refinement, which we leave to future work.

RNO further models the coupling between the data's wave content and the neural-operator architecture, yielding a more complete justification of the design. Nonetheless, RNO has limitations: in some settings, control over local components lacks sufficient precision that is an issue we aim to address in subsequent work.

LLM USAGE STATEMENT: LANGUAGE POLISHING ONLY

We used a large language model (LLM) solely for language polishing to improve clarity and control manuscript length (e.g., correcting grammar and refining phrasing for greater precision.). The LLM was not used for problem formulation, methodology or experiment design, data processing or analysis, result interpretation, or drawing conclusions. All edits suggested by the LLM were reviewed and approved by the authors.