Computer Methods in Applied Mechanics and Engineering 463 (2027) 119388

Contents lists available at ScienceDirect

Comput. Methods Appl. Mech. Engrg.

journal homepage: www.elsevier.com/locate/cma

Learning mesh-free discrete differential operators with self-supervised graph neural networks

Lucas Gerken Starepravo [a,*], Georgios Fourtakas [a], Steven Lind [b], Ajay B. Harish [a], Tianning Tang [a], Jack R.C. King [a]

[a] School of Engineering, The University of Manchester, Manchester, UK
[b] School of Engineering, Cardiff University, Cardiff, UK

ARTICLE INFO

Dataset link: https://github.com/uom-complexfluids/nemdo

Keywords:
Mesh-free
Neural networks
Operator learning

ABSTRACT

Mesh-free numerical methods provide flexible discretisations for complex geometries; however, classical meshless discrete differential operators typically trade low computational cost for limited accuracy or high accuracy for substantial per-stencil computation. We introduce a parametrised framework for learning mesh-free discrete differential operators using neural networks trained via polynomial moment constraints derived from truncated Taylor expansions. The model maps local stencils relative positions directly to discrete operator weights. The current work demonstrates that neural networks can learn classical polynomial consistency while retaining robustness to irregular neighbourhood geometry. The learned operators depend only on local geometry, and can be reused across particle configurations and governing equations. The framework introduces an additional design axis absent from traditional discretisations: the learned operators have a capacity-dependent error floor, so a model may be optimised for accuracy, for computational cost, or a balance of the two. A trained model may be applied across resolutions, though accuracy cannot be refined below the limiting error set by model capacity. We evaluate the framework using standard numerical analysis diagnostics, showing improved accuracy over Smoothed Particle Hydrodynamics, and applicability is demonstrated by solving the Poisson's equation and the weakly compressible Navier–Stokes equations using the learned operators.

1. Introduction

Partial differential equations (PDEs) are foundational to the modelling of physical systems across science and engineering. Yet, analytical solutions are rarely available for realistic problems of interest. Thus, numerical methods are often used to approximate the governing equations.

Mesh-based numerical methods, such as the finite difference method (FDM) [1,2], finite element method (FEM) [3], finite volume method (FVM) [4,5], and spectral methods [6], have long been the standard tools for solving PDEs [7], given these methods can be made extremely accurate for simple geometries. Despite their maturity, these approaches face challenges regarding accuracy near interfaces [8] and the overhead associated with the mesh: this includes both the need for complex adaptation procedures [9] and initial generation phases that, for intricate geometries, may be more time-consuming than the simulation itself [10]. Mesh-free numerical methods have significant potential, as discretisation relies solely on local connectivity information between collocation

* Corresponding author.
E-mail address: lucas.gerkenstarepravo@postgrad.manchester.ac.uk (L.G. Starepravo).

https://doi.org/10.1016/j.cma.2026.119388
Received 25 March 2026; Received in revised form 14 August 2026; Accepted 2 September 2026
Available online 12 September 2026

points without requiring topological connectivity. Furthermore, collocations points can be positioned flexibly to satisfy resolution requirements and conform to complex geometries [7,9].

Developed in the 1970s, Smoothed Particle Hydrodynamics (SPH) [11] is perhaps the most widely used mesh-free numerical method. It is commonly employed in a Lagrangian formulation, where particles are advected with the underlying velocity field, and differential operators are approximated via radial kernel interpolation over neighbouring particles. SPH was originally developed for astrophysical problems [11], formulated with symmetries which guarantee exact global conservation, and it has since been applied more broadly to terrestrial fluid dynamics [12], in which settings global conservation is often sacrificed in favour of anti-symmetric formulations exhibiting improved local accuracy. However, the method remains formally zeroth order consistent, limiting accuracy it can achieve, particularly in complex turbulent flows [9,13]. This motivated the development of more accurate numerical methods that enforce polynomial consistency through the solution of a local or global linear system [14–17]. However, consistent mesh-free methods are typically implemented in an Eulerian framework [18,19], as they require solving dense local linear systems for each local neighbourhood to obtain weights for each stencil. In a Lagrangian setting, where the particle configuration evolves in time, these methods require solving a local linear system for every particle at every time step, leading to substantial computational overhead. As a result, mesh-free simulations commonly face a trade-off between classical SPH kernels, which are computationally efficient but inconsistent and exhibit poor convergence, and consistency-corrected methods, which offer improved accuracy at the expense of significantly higher computational complexity.

Recent years have seen growing interest in data-driven approaches to scientific computing. Prominent examples include physics-informed neural networks (PINNs) [20], neural operators [21], and learned discretisation methods [22]. While these represent significant advances, they share a common limitation: the learned model is tied to the training distribution — whether through dependence on solution data, prescribed boundary conditions, or specific grid structures — limiting reuse across different problems.

In the current work, we introduce Neural Mesh-Free Differential Operator (NeMDO), a self-supervised neural network framework that learns discrete *mesh-free* differential operators directly from irregular particle configurations.[1] Rather than learning PDE solutions or problem-specific closures, NeMDO constructs local operator weights by learning polynomial consistency constraints derived from Taylor expansions, yielding operators with a clear mathematical foundation. Our results show that neural networks can learn these consistency constraints and predict operator weights with structural properties analogous to those of classical mesh-free discretisations.

NeMDO is, to our knowledge, the first framework to learn approximately consistent mesh-free differential operators in a self-supervised manner directly from stencil geometry, without recourse to PDE data or per-stencil reconstruction at runtime. This yields a continuous cost–accuracy tradeoff between fixed-kernel methods such as SPH and exactly-solved reconstructions such as LABFM, with reconstruction cost amortised through offline training.

The resulting operators are strictly local, generalise across heterogeneous node configurations and resolutions, and are independent of any particular governing equation. We validate the learned operators using established numerical analysis tools, including convergence studies, modal response and stability analyses, and test the robustness of the framework through parametric studies. We further demonstrate applicability by solving the Poisson equation and the weakly compressible Navier–Stokes equations. The remainder of this paper is organised as follows. Section 2 provides the theoretical background on discrete differential operators and machine learning for PDEs. In Section 3, we detail traditional mesh-free operators, followed by a description of the proposed NeMDO methodology in Section 4. We then present a comprehensive series of experiments and discussions in Section 5. A parametric ablation study is provided in Sections 6, and 7 highlights the limitations of the current framework. Finally, Section 8 offers concluding remarks and summarises our key findings.

2. Background & related work

2.1. Classical numerical approximation of differential operators

Given the discretisation of a spatial domain Ω by a set of collocation points $\mathcal{P} := \{\mathbf{x}_i\}_{i=1}^{P} \subset \Omega \subset \mathbb{R}^d$, where $\mathbf{x}_i = (x_i, y_i)$ in the $d = 2$ case, a general local discrete approximation of a differential operator acting on ϕ can be written as

$$L^D(\phi(\mathbf{x}_i)) = \sum_{j \in \mathcal{N}_i} (\phi(\mathbf{x}_j) - \phi(\mathbf{x}_i)) w_{ji}^D, \tag{1}$$

where L^D denotes the discrete approximation of the differential operator D, $\mathcal{N}_i$ is the local support (or computational stencil) associated with node i, and w_{ji}^D are the corresponding stencil weights. This formulation encompasses a broad class of numerical methods given appropriate stencils and indexing—including finite difference method [2], finite element method [3], SPH [11], and mesh-free high-order methods such as the Local Anisotropic Basis Function Method (LABFM) [17]—with the specific discretisation determined by how the weights w_{ji}^D and neighbourhoods $\mathcal{N}_i$ are constructed.

[1] A preliminary version of this work was presented at the International Conference on Learning Representations (ICLR) 2026 AI&PDE workshop [23], which is non-archival. The present manuscript adds a significantly extended analysis and verification, containing: (i) demonstration of architectural flexibility by training both a multi-layer perceptron (MLP) baseline and a graph neural network (GNN), (ii) a method for directional derivatives allowing reuse of trained operators, (iii) analysis of the imaginary component of the modal response and of waves at different angles, (iv) extension of the method to Dirichlet boundary conditions, (v) application to the Poisson equation and analysis of the impact of different node distributions, (vi) an extended ablation with cross-evaluation across levels of particle disorder and rotational robustness, and (vii) analysis of point-cloud statistics.

The accuracy and utility of the discretised solution depends on the *convergence, consistency*, and *stability* of the numerical method. For a discretisation to be consistent, the discrete operator must reproduce the Taylor moments up to a prescribed order. In mesh-free methods, these consistency conditions are typically enforced by solving local reconstruction problems in each particle neighbourhood. Representative approaches include the reproducing kernel particle method [24], generalised moving least squares (GMLS) [25,26], the radial basis function-finite difference methods (RBF-FD) [27–29], and LABFM [17]. While solving these local linear systems enforces polynomial consistency and can yield high-order convergence, these methods introduce high computational cost when computing stencil weights. This computational overhead can be prohibitive in Lagrangian/Semi-Lagrangian frameworks, when the particles move during time integration, and the stencil weights must be recomputed at every time step.

In SPH, the stencil weights are obtained with a compact isotropic kernel [30]. The most used SPH kernel functions are the B-spline functions [31] and the Wendland functions [32], where the error of the integral approximation is $\mathcal{O}(h^2)$ in the limit of large h/s [33]—where h is the smoothing length of the kernel (i.e. the radius of the neighbouring region about point i) and s is the average collocation point spacing—with convergence of $\mathcal{O}(h)$ or lower in practice [34–36]. This discrepancy is largely attributed to the discretisation of the original SPH formulation, which assumes a continuum; furthermore, irregularities in collocation point distribution can lead to further deterioration in accuracy [13]. A consistency analysis of discrete SPH indicates that gradient operators are zeroth-order accurate in h on disordered particles, and certain Laplacian formulations may diverge as resolution increases. Nonetheless, convergence is often observed within a finite resolution window and for sufficiently well-behaved particle configurations and low wavenumbers [13]. The convergence behaviour further depends on the neighbour count $\mathcal{N}_i$, which is kernel-dependent. In practice, achieving stable and convergent SPH discretisations typically requires large support sizes, often involving tens to hundreds of neighbours per particle [37].

2.2. Machine learning for partial differential equations

Physics-Informed Neural Networks is a prominent paradigm in scientific machine learning, where neural networks are trained to approximate the solution field associated with a specific PDE instance. This approach embed the governing equations, boundary conditions, and initial conditions into the training objective, typically by penalising PDE residuals evaluated through automatic differentiation [20,38]. This formulation enables mesh-free, unsupervised training, and can achieve high accuracy for well-posed problems. However, PINNs are inherently instance-specific, as the learned model corresponds to a particular choice of governing equations and initial/boundary conditions, typically requiring retraining when these change.

Neural Operators constitute an alternative line of research that focuses on learning mappings between function spaces. Neural operator methods approximate the solution operator of a PDE from paired samples of input and output functions, enabling inference across varying discretisations and resolutions. Prominent examples include Deep Operator Networks (DeepONets) [39], Fourier Neural Operators (FNOs) [40], and graph-based neural operators (GNOs) [41]. These frameworks learn implicit representations of the underlying PDE dynamics from data, effectively learning the response map for a prescribed family of geometries [42], loads, and boundary conditions [43], and in some cases have demonstrated transfer across related PDEs; although the associated training and inference costs remain substantial [44].

Learning-based particle methods leverage the locality of differential operators, to learn local mesh-free PDE time integration from solutions obtained from traditional operators [45–49]. While effective for specific flow classes, these approaches are typically trained end-to-end on time-dependent solution data, tying the learned representations closely to the governing equations and regimes observed during training.

Learning Low-Level Numerical Operators: A complementary line of work focuses on learning numerical discretisations or low-level operator components. Rather than approximating solutions directly, these frameworks learn local numerical building blocks — such as derivative stencils or basis functions — while preserving the structure of established discretisation schemes. Examples include learning data-driven finite-difference stencils to enable resolution coarsening [22,50,51], extending these to finite-volume methods [52]. These approaches operate at a similar level of abstraction to the framework proposed in the current manuscript, predicting discretisation weights rather than solution fields. However, a key distinction is that prior work conditions the discretisation weights on local field values, tying the trained model to the spectral characteristics and functional forms present in the training data. NeMDO instead conditions weights solely on node positions, making the learned operator fully PDE-agnostic and reusable without retraining. While traditional neural operators and its variants can learn local differential operator actions from solution data [53], they do not explicitly construct reusable discrete operators as functions solely from stencil geometry. Despite these advancements, the ability of such models to approximate differential fields is often restricted to the spectral characteristics and functional forms present in the training distribution.

More recently, Choi et al. [54] proposed data-driven finite element methods (DD-FEM), which replaces classical polynomial bases with locally learned basis functions. This formulation enables reuse across geometries, mesh types, and boundary conditions, highlighting the potential of learning reusable numerical building blocks that remain compatible with classical discretisation principles. The framework presented in the present manuscript is closely aligned with this operator-level perspective, but targets the learning of mesh-free discrete differential operators rather than local basis functions in mesh-based discretisations. This positions NeMDO within a distinct category from solution surrogates and neural operators: rather than learning a response map for a prescribed problem family, NeMDO learns a reusable discretisation tool that can be applied across governing equations without retraining.

We now assess the present framework against the criteria imposed by Choi et al. [54] for foundation models in computational science. NeMDO is not *trained on a broad distribution of physical systems*: the loss contains Taylor moment conditions rather than

solution data from physical systems, so equation-independence arises from the learning formulation rather than from exposure to multiple physical systems. It does exhibit **wide generalisation**, given a single trained model is being applied here to two governing equations, across resolutions, and across point-cloud statistics ranging from perturbed Cartesian lattices to particle-shifted distributions. This requires **no retraining from scratch or structural modification**; the same weights are deployed throughout, with no fine-tuning. Of the desirable characteristics, the framework meets **scientific consistency**, polynomial consistency being imposed directly through the loss, and admits the classical truncation-error analysis associated with **mathematical rigour**.

Two of the stated criteria are not met. First, NeMDO is not data-driven in the sense intended: it is trained on stencil geometry against analytically derived constraints, with no simulation, experimental or observational data. Second, generalisation is incomplete along two axes fixed at training time. The stencil size $|\mathcal{N}|$ is currently fixed, and the spatial dimension is set by the choice of monomials in the loss; transfer to a different stencil size or to different number of spatial dimensions requires retraining, though in the latter case only the additional monomials are needed and no change to the formulation. Variable stencil size and the extension to three dimensions are discussed in Section 8. NeMDO thus occupies an unusual position relative to these criteria, achieving the generalisation across physical systems such models aim for, but by construction rather than by exposure.

3. Standard numerical methods

In this work, we use LABFM as an exemplar high-order mesh-free method, noting that it shares characteristics (structure of computational stencil, form of discrete operator, and local linear system) with other high-order mesh-free methods (e.g. generalised finite difference method [55], GMLS [25], RBF-FD [56]). For a comprehensive derivation of LABFM, we refer the reader to [17]. LABFM requires solving a linear system with rank equal to the number of Taylor monomials required to ensure polynomial consistency of order p, making it a computationally efficient consistent mesh-free baseline. A discrete operator is said to have polynomially consistent of order p if it reproduces the target derivative exactly for all polynomials of degree $\leq p$. Other high-order mesh-free methods, such as RBF-FD, require solving significantly larger linear systems [57] and consequently incur larger computational cost. Additionally, we use two standard SPH kernels as benchmark: the quintic spline [31] and the Wendland C2 kernel [32]. Both have been widely adopted in computational fluid dynamics (CFD)-oriented simulations [58,59]. In the current section, we explain SPH and LABFM in detail.

3.1. Smoothed particle hydrodynamics operators

For SPH gradient operators, one may choose the symmetric or anti-symmetric forms, with the former providing exact conservation and the latter providing greater accuracy. With the focus here on consistent operators, we consider only the anti-symmetric operator, which can be expressed in the form of Eq. (1), here instantiated for the x direction, with

$$w^x_{ji} = \partial_x W(\mathbf{x}_{ji}, h) V_j, \tag{2}$$

where the subscript indicates $(\cdot)_{ji} = (\cdot)_j - (\cdot)_i$, thus, $\mathbf{x}_{ji}$ denotes the relative position of the collocation point j with respect to point i, W is a smoothing kernel, V_j is the volume of the particle, and ∂_x denotes the differentiation with respect to x_i. When computing the Laplacian with SPH operators, we employ the Morris operator [60]:

$$w^{\Delta}_{ji} = \frac{-2\,\mathbf{x}_{ji}}{\|\mathbf{x}_{ji}\|} \cdot \nabla_i W_{ji} V_j \tag{3}$$

where $\|\cdot\|$ denotes the Euclidean distance, and the superscript Δ denotes the Laplacian.

3.1.1. Smoothing kernels

Quintic spline: The quintic spline kernel has support $r \in [0, 3]$ and its given by

$$W^{\mathrm{QS}}(r) = \sigma \begin{cases} (1-r)_+^5 - 6\left(\frac{2}{3} - r\right)_+^5 + 15\left(\frac{1}{3} - r\right)_+^5, & 0 \leq r < 1, \\ (1-r)_+^5 - 6\left(\frac{2}{3} - r\right)_+^5, & 1 \leq r < 2, \\ (1-r)_+^5, & 2 \leq r < 3, \\ 0, & r \geq 3, \end{cases} \tag{4}$$

where $r = \|\mathbf{x}_{ji}\|/h$, σ is a normalisation constant that depends on the system's dimension, in two dimensions $\sigma = \frac{7}{478\pi h^2}$ [61]. We set $h = 1.5$ s, which yields approximately 60–65 neighbours for the quintic spline.

Wendland C2: The Wendland C2 kernel has a support of $r \in [0, 2]$ and is defined as

$$W^{\mathrm{WC2}}(r) = \sigma(1-r)^4(1+4r). \tag{5}$$

In two dimensions, $\sigma = \frac{7}{\pi h^2}$. We set $h = 1.5$ s, which yields approximately 25–30 neighbours for the Wendland C2.

3.2. The Local Anisotropic Basis Function Method (LABFM)

In LABFM, a general discrete operator is defined to approximate differential operators as shown in Eq. (1). The weights w_{ji}^D are given by a weighted sum of anisotropic basis functions (ABFs),

$$w_{ji}^D = \mathbf{W_{ji}} \cdot \boldsymbol{\Psi}_i^D = \mathrm{W}_{ji}^1 \Psi_{i,1}^D + \mathrm{W}_{ji}^2 \Psi_{i,2}^D + \mathrm{W}_{ji}^3 \Psi_{i,3}^D + \cdots \tag{6}$$

in which the vector $\mathbf{W_{ji}}$ are the ABFs, and $\boldsymbol{\Psi}_i^D$ is a coefficient vector. The coefficient vector is obtained by solving the following linear system

$$\mathbf{A_i} \boldsymbol{\Psi}_i^D = \mathbf{M}^D, \tag{7}$$

where $\mathbf{M}^D$ is a vector that contains the coefficients of the Taylor expansion about point i in a computational stencil, we will refer to this vector as the "moments" of the stencil. Below, we show instances of the moment vector when approximating exemplar differential operators

$$\boldsymbol{C}^D = \begin{cases} [1,0,0,0,0,0,\ldots]^T & if \;\; D = x \\ [0,1,0,0,0,0,\ldots]^T & if \;\; D = y \\ [0,0,1,0,1,0,\ldots]^T & if \;\; D = \Delta, \end{cases} \tag{8}$$

where $D = x$ and $D = y$ denote first derivatives in the respective coordinate directions. For a given support region, $\mathbf{A_i}$ is given by

$$\mathbf{A_i} = \sum_{j \in \mathcal{N}_i} \mathbf{X_{ji}} \otimes \mathbf{W_{ji}}, \tag{9}$$

where $\mathbf{X_{ji}}$ is defined as the vector of Taylor monomials,

$$\mathbf{X}_{ji} := \left[x_{ji}, y_{ji}, \frac{x_{ji}^2}{2}, x_{ji} y_{ji}, \frac{y_{ji}^2}{2}, \frac{x_{ji}^3}{6}, \ldots \right]^T,$$

and the rank of $\mathbf{A_i}$ is given by,

$$\mathrm{rank}(\mathbf{A_i})_{2D} = \frac{p^2 + 3p}{2} \quad \text{and} \quad \mathrm{rank}(\mathbf{A_i})_{3D} = \sum_{m=2}^{m=p+1} \frac{m(m+1)}{2} \tag{10}$$

in 2 and 3 dimensions, respectively.

To construct the ABFs, bivariate Hermite polynomials are combined with a smoothing kernel. The qth element of $\mathbf{X}_{ji}$ is proportional to $x_{ji}^a y_{ji}^b$. Thus, the qth ABF is defined below in 2 dimensions

$$W_{ji}^q = \frac{\psi\left(\|\mathbf{x}_{ji}\|/h_i\right)}{\sqrt{2^{a+b}}} H_a\left(\frac{x_{ji}}{h_i\sqrt{2}}\right) H_b\left(\frac{y_{ji}}{h_i\sqrt{2}}\right) \tag{11}$$

where H_a is the ath order univariate Hermite polynomial (of the physicists kind), and ψ is an RBF. In this work, the Wendland C2 kernel is used, following [62].

4. Method

In this section, we introduce NeMDO (Neural Mesh-Free Differential Operator), a new approach for computing weights used in discrete differential operators in mesh-free simulations with disordered node distributions (see Appendix A). Rather than computing these weights with a smoothing kernel or by solving per-particle linear systems, we formulate operator construction as a learning problem defined over local particle neighbourhoods. We employ our framework to predict discrete operator weights based solely on the relative positions of neighbouring particles, enabling local mesh-free spatial differential approximations.

Learning Mesh-Free Discrete Differential Operators. Given a discretised domain with collocation points $\mathcal{P}$, we associate to each point $\mathbf{x}_i$ a local neighbourhood $\mathcal{N}_i := \{\mathbf{x}_j \in \mathcal{P} : \|\mathbf{x}_{ji}\| \leq d_i^{(N)}\}$, where $d_i^{(N)}$ is the distance to the Nth nearest neighbour of $\mathbf{x}_i$. Our objective is to construct a local discrete approximation L^D of a differential operator D acting on ϕ, such that the operator evaluation at $\mathbf{x}_i$ is approximated by a weighted sum over neighbouring samples

$$L^D(\phi(\mathbf{x}_i)) = \sum_{j \in \mathcal{N}_i} (\phi(\mathbf{x}_j) - \phi(\mathbf{x}_i)) w_{ji}^D(\{\mathbf{x}_{ji}\}_{j \in \mathcal{N}_i}; \theta), \tag{12}$$

where $w_{ji}^D(\{\mathbf{x}_{ji}\}_{j \in \mathcal{N}_i}; \theta)$ are the data-driven weights inferred by a model parametrised by θ, conditioned on the stencil geometry. For brevity, the explicit dependence on the stencil geometry is omitted in subsequent notation where it is understood from the context, i.e. $w_{ji}^D(\theta) \equiv w_{ji}^D(\{\mathbf{x}_{ji}\}_{j \in \mathcal{N}_i}; \theta)$. Note that the difference form of Eq. (12) automatically satisfies the zeroth-moment condition (i.e. the gradients of constants are zero). We map the neighbourhood geometry to operator weights using a learnt function shared across all stencils. The learnt mapping does not depend on the field samples $\phi(\mathbf{x}_i)$, the governing equation, or the global simulation domain. Instead, it defines a reusable local operator constructor that can be applied consistently across particle distributions, resolutions, and governing equations.

Local Operator Parametrisation: The NeMDO framework is agnostic to the choice of parametric function f_θ, accommodating arbitrary neural architectures ranging from standard multi-layer perceptrons to geometric graphs. To demonstrate this generality, we implement two architectures within the framework: (i) a multi-layer perceptron (MLP) baseline, and (ii) a message-passing GNN. As the GNN serves as the primary and default architecture for our framework, it is referred to simply as "NeMDO" throughout the text and figures, while the alternative configuration is denoted as the "MLP".

4.0.0.1. MLP baseline. The MLP baseline processes the entire neighbourhood simultaneously. Because the MLP operates on a fixed-length concatenated input, a canonical ordering of the neighbours is required: neighbours are sorted by Euclidean distance from the central particle. The normalised relative positions are concatenated into a single input vector,

$$\mathbf{z}_i = \left[\hat{\mathbf{x}}_{(1)i}^\top, \hat{\mathbf{x}}_{(2)i}^\top, \dots, \hat{\mathbf{x}}_{(N)i}^\top\right]^\top \in \mathbb{R}^{N\cdot d}, \tag{13}$$

where the hat denotes relative positions normalised by the furthest-neighbour Euclidean distance (thus $\|\hat{\mathbf{x}}_{(N)i}\| = 1$), and the parenthesised subscript denotes the distance ordering $\|\hat{\mathbf{x}}_{(1)i}\| \le \|\hat{\mathbf{x}}_{(2)i}\| \le \cdots \le \|\hat{\mathbf{x}}_{(N)i}\|$ over $\mathcal{N}_i$. Without this sorting, identical stencils whose neighbours happen to be enumerated in a different order would present as distinct inputs to the network. The MLP then maps this concatenated vector to the full set of operator weights,

$$\{\hat{w}_{ji}^D\}_{j\in\mathcal{N}_i} = \mathrm{MLP}_\theta(\mathbf{z}_i), \quad \mathrm{MLP}_\theta : \mathbb{R}^{N\cdot d} \to \mathbb{R}^N. \tag{14}$$

Each hidden layer takes the form $\mathbf{h}^l = \sigma\left(\mathbf{W}^l\mathbf{h}^{l-1} + \mathbf{b}^l\right)$, where $\mathbf{W}^l$ and $\mathbf{b}^l$ are learnable parameters and σ is the Sigmoid Linear Unit (SiLU) activation function. Unlike the GNN, the MLP relies on the explicit distance sorting rather than architectural invariance to handle neighbour permutations.

4.0.0.2. Message-passing GNN. For each local neighbourhood $\mathcal{N}_i$ a graph is defined $\mathcal{G}_i := (\mathcal{V}_i, \mathcal{E}_i)$, where the node set $\mathcal{V}_i \equiv \mathcal{N}_i$ consists of the particle i and its neighbours. The edge set $\mathcal{E}_i := \{(i,j),(j,i) : j \in \mathcal{N}_i \setminus \{i\}\}$ defines star-shaped, bidirectional edges between the central particle and its neighbours. In contrast to dense radius graph seen in particle-based learning approaches [46], this star-shaped construction yields linear complexity in the neighbourhood size, and the graph connectivity is directly constructed from the neighbourhood lists already computed in standard mesh-free methods, requiring no additional spatial searches or geometric preprocessing. Node attributes encode the normalised relative position.

The graphs $\mathcal{G}_i$ are processed by a shared parametric function f_θ, implemented by a graph neural network (GNN),

$$\{\hat{w}_{ji}^D(\theta)\}_{j\in\mathcal{N}_i} = f_\theta\left(\mathcal{G}_i, \{\hat{\mathbf{x}}_{ji}\}_{j\in\mathcal{N}_i}\right). \tag{15}$$

The relative positions are first embedded into a latent representation using a multi-layer perceptron (MLP),

$$\mathbf{v}_j^0 = \mathrm{MLP}_\theta^{\mathrm{Emb}}\left(\hat{\mathbf{x}}_{ji}\right), \quad \mathrm{MLP}_\theta^{\mathrm{Emb}} : \mathbb{R}^d \to \mathbb{R}^{F_h}, \quad \forall j \in \mathcal{N}_i, \tag{16}$$

where $\mathbf{v}_j^0$ are the latent node features that are passed to the graph layers, the number of dimensions of the system is given by d and the number of hidden features per node after encoding is denoted as F_h. Message passing and node update is then performed over the stencil graph, where we define L graph layers as

$$\mathbf{v}_j^l = \mathrm{MLP}_\theta^{l,\mathrm{Upd}}\left(\mathbf{v}_j^{l-1}, \oplus_{k\in\mathcal{K}_j} \mathrm{MLP}_\theta^{l,\mathrm{Msg}}\left(\mathbf{v}_k^{l-1}\right)\right), \quad l = 1,\dots,L, \quad \forall j \in \mathcal{N}_i, \tag{17}$$

where $\mathrm{MLP}_\theta^{l,\mathrm{Upd}}$ and $\mathrm{MLP}_\theta^{l,\mathrm{Msg}}$ are the update and message networks, respectively, at layer l; $\mathbf{v}_j^l$ denotes the node features of node j at graph layer l, $\mathcal{K}_j \subset \mathcal{N}_i$ is the set of nodes connected to node j, and $\oplus$ is a differentiable, permutation-invariant aggregation operator. In our implementation, $\oplus$ is realised as an attention-weighted aggregation based on cross-graph matching [63].

Lastly, a final MLP maps the latent node representations to normalised discrete operator weights,

$$\hat{w}_{ji} = \mathrm{MLP}_\theta^{\mathrm{Out}}(\mathbf{v}_j^L), \quad \mathrm{MLP}_\theta^{\mathrm{Out}} : \mathbb{R}^{F_h} \to \mathbb{R}, \quad \forall j \in \mathcal{N}_i \tag{18}$$

in which $\{\hat{w}_{ji}(\theta)\}_{j\in\mathcal{N}_i}$ are the weights associated with the target differential operator. A schematic of the framework can be seen in Fig. 1. All MLPs in the GNN use hyperbolic tangent activation functions.

Both the MLP baseline and GNN are trained using the adaptive moment estimation (Adam) optimiser. In the present work, we focus on demonstrating the potential of this form of method, and reserve an assessment of different optimisers for future work. The activation functions differ between the MLP baseline and the GNN as each was selected independently via a hyperparameter study: SiLU performed best for the MLP baseline and hyperbolic tangent for the GNN. Each architecture is therefore evaluated at its own optimal configuration rather than under an artificially matched activation that would handicap one of them.

Translation invariance is enforced by encoding geometry through relative position vectors, while permutation invariance in the GNN is achieved through symmetric aggregation operations over neighbouring nodes (which is inherently handled by the graph architecture). Scale robustness across particle resolutions is introduced by normalising relative distances with respect to the maximum distance between neighbouring particles and the central particle. The proposed architecture is not explicitly rotation-equivariant; instead, rotational robustness is obtained implicitly through training on neighbourhoods with diverse node configurations.

Self-Supervised Learning with Polynomial Consistency Constraints: To learn the parameters θ, we employ a training objective derived from polynomial consistency conditions imposed by a truncated Taylor expansion. Consequently, the method does not require labelled target weights for individual stencils. To formulate the loss function, let us first consider a scalar field in

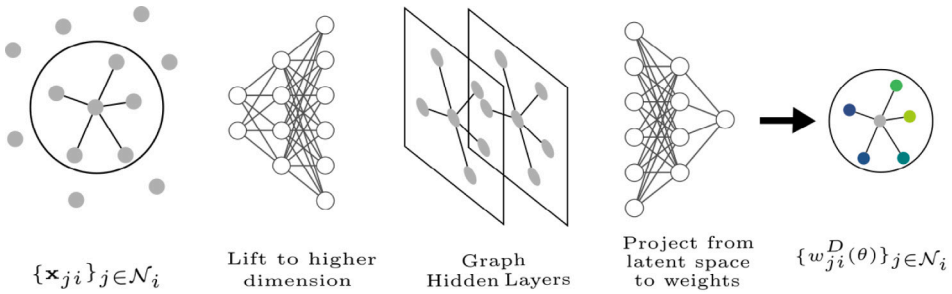


Fig. 1. We learn the mapping from relative position of particles within a local neighbourhood, to a local set of weights that approximate differential operators. The learned operator is physics-agnostic, and local to the computational stencil. The framework consists of a lifting the relative positions to a higher dimension with a neural network, followed by a stack of message-passing graph layers, and a final output head that maps latent representations to operator weights.

2 dimensions $\phi : \mathbb{R}^2 \to \mathbb{R}$ sampled on a discrete point cloud $\mathcal{P}$. For a given neighbourhood $\mathcal{N}_i$, where $\partial(\phi)|_i$ denotes the vector of partial derivatives of ϕ at particle i,

$$\partial(\phi)|_i := \left[\left.\frac{\partial\phi}{\partial x}\right|_i, \left.\frac{\partial\phi}{\partial y}\right|_i, \left.\frac{\partial^2\phi}{\partial x^2}\right|_i, \left.\frac{\partial^2\phi}{\partial x\partial y}\right|_i, \left.\frac{\partial^2\phi}{\partial y^2}\right|_i, \left.\frac{\partial^3\phi}{\partial x^3}\right|_i, \dots \right]^T,$$

and $\mathbf{X}_{ji}$ is the vector of Taylor monomials of the position of j relative to i, the multivariate Taylor expansion of ϕ about point i may be written compactly as

$$(\phi)_j = (\phi)_i + \mathbf{X}_{ji} \cdot \partial(\phi)|_i \tag{19}$$

The spatial dimension enters only through the monomials in $\mathbf{X}_{ji}$ and the corresponding entries in $\partial(\phi)|_i$: extending to three dimensions requires including the additional monomials $(z_{ji}, x_{ji}z_{ji}, y_{ji}z_{ji}, z^2_{ji}, \dots)$ and their associated derivatives, with no change to the architecture or training procedure. One can analyse the error in the discrete differential operator L^D by substituting the Taylor expansion into Eq. (1), obtaining

$$L^D_i(\phi) = \sum_{j\in\mathcal{N}_i} \mathbf{X}_{ji} \cdot \partial(\phi)|_i \, w^D_{ji} \tag{20}$$

For a target continuous differential operator D, polynomial consistency of order p requires that the discrete operator L^D_i reproduces the action of D on all monomials of total degree at most p, i.e.

$$L^D_i(\phi) = D(\phi)|_i, \quad \forall \text{ polynomials of degree} \le p. \tag{21}$$

For instance, to approximate $\frac{\partial\phi}{\partial x}$ with second-order polynomial consistency, it is required that the first-order discrete moments satisfy

$$\sum_{j\in\mathcal{N}_i} x_{ji} w^x_{ji} = 1, \quad \sum_{j\in\mathcal{N}_i} y_{ji} w^x_{ji} = 0 \tag{22}$$

with all remaining moments associated with the second-order approximation set to 0. When these conditions are satisfied, the resulting discrete operator approximates $\frac{\partial\phi}{\partial x}$ with second-order convergence for sufficiently smooth ϕ, yielding a truncation error of order s^2.

In our framework, we approximate polynomial consistency by minimising the error between predicted and target moments over all particles

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \left\| \sum_{j\in\mathcal{N}_i} \hat{\mathbf{X}}_{ji} \, \hat{w}^D_{ji}(\theta) - \mathbf{M}^D \right\|_2^2, \tag{23}$$

where $\hat{}$ indicates a normalised quantity. Thus, the proposed method does not require individual labels for each computational stencil. One can compute the training operator moments directly from the inputs and the predicted weights, and compute the loss based on the fixed target moments.

Data Generation: Training is performed on local particle neighbourhoods $\mathcal{N}_i$ sampled from synthetic disordered point cloud distributions. These are obtained by applying stochastic perturbations to a regular Cartesian grid with average spacing s_u. Specifically, each grid node is independently displaced by a noise term sampled from a uniform distribution $\mathcal{U}(-\frac{\epsilon s_u}{2}, \frac{\epsilon s_u}{2})$ per coordinate, where ϵ represents the non-dimensional noise intensity. This formulation ensures that the perturbation magnitude scales proportionally with the grid resolution, enabling a systematic analysis of operator sensitivity across different discretisation scales. We note that training is not restricted to this particular distribution family; the learned operators generalise to other point-cloud

statistics, as demonstrated in Section 5.6 using particle-shifted nodes [64–67]. There is a broad range of point-cloud generation methods available in the mesh-free literature [68,69]; for a comprehensive overview, we refer the reader to [70].

Unless stated otherwise, models are trained at a disturbance level of $\epsilon = 1.0$, which is significantly larger than the particle disorder typically encountered in practical mesh-free simulations. This choice ensures that the learned operators remain robust under severe geometric irregularity (as shown in Appendix A).

To expose the network to near-boundary geometry during training, we augment the dataset with truncated stencils that mimic the one-sided support typical of particles adjacent to a wall. For each raw stencil in the training set, a flat wall is placed at a random orientation $\varphi \sim \mathcal{U}(0, 2\pi)$ and a random normalised offset $\delta \sim \mathcal{U}(\delta_{\min}, \delta_{\max})$ from the central node, and all neighbours on the far side of the wall are discarded. Because the raw neighbour pool contains more than N nodes, discarded neighbours are replaced by more distant nodes from the same pool, preserving the stencil size. The surviving neighbours are re-normalised by their own maximum radius, matching the convention used for interior stencils. The effect of this augmentation on solution accuracy is assessed in Section 5.6.1.

Operator Rescaling and Reuse: For a target differential operator of order m, the predicted weights are rescaled by $(d_i^{(N)})^{-m}$ to recover the correct physical dimensions, reflecting the mth order spatial scaling of the operator—distinct from the formal order of accuracy p—and ensuring consistency across resolutions. The resulting weights are then used within standard mesh-free discretisations to approximate the action of the operator D on arbitrary fields.

Because operator construction depends only on local geometry, trained models can be reused across different resolutions, domains, and governing equations without retraining, provided the neighbourhood size $|\mathcal{N}_i|$ remains fixed. This enables learned operators to be deployed as drop-in numerical components within existing mesh-free solvers.

Directional Derivatives by Stencil Rotation: A single network is used for all directional derivatives. Let $\mathbf{n} = (\cos\vartheta, \sin\vartheta)$ denote the unit vector at angle ϑ to the x-axis, and let R_ϑ be the rotation satisfying $R_\vartheta \mathbf{n} = \mathbf{e}_x$, where $\mathbf{e}_x = (1, 0)$. Weights for the derivative along $\mathbf{n}$ are obtained by evaluating the same network on the rotated stencil,

$$\{\hat{w}_{ji}^{D,(\mathbf{n})}\}_{j\in\mathcal{N}_i} = f_\theta\Big(\{R_\vartheta \hat{\mathbf{x}}_{ji}\}_{j\in\mathcal{N}_i}\Big). \tag{24}$$

The same trained network can therefore predict weights for the derivative along any direction in the plane. The Laplacian is rotation invariant and requires no alignment.

This construction preserves accuracy, as shown below. Writing $w_{ji}^{(\mathbf{n})} \equiv \hat{w}_{ji}^{D,(\mathbf{n})}$, $\mathbf{x}_{ji} \equiv \hat{\mathbf{x}}_{ji}$ and $\mathbf{M}_\vartheta \equiv \mathbf{M}_{i,\vartheta}^D$ for brevity, let

$$\mathbf{M}_\vartheta \;=\; \sum_{j\in\mathcal{N}_i} w_{ji}^{(\mathbf{n})}\, \mathbf{x}_{ji} \tag{25}$$

denote the moment attained by the predicted weights on the original offsets. The same weights evaluated against the rotated offsets give $\sum_{j\in\mathcal{N}_i} w_{ji}^{(\mathbf{n})} R_\vartheta \mathbf{x}_{ji} = R_\vartheta \mathbf{M}_\vartheta$, since R_ϑ is constant. It is against these rotated offsets that the network enforces the conditions it was trained on. For a network trained on the first derivative in the x-direction,

$$R_\vartheta \mathbf{M}_\vartheta \;\approx\; \mathbf{M}^x. \tag{26}$$

The approximation in Eq. (26) indicates the network's deviation from exact consistency; the transformation $R_\vartheta \mathbf{M}_\vartheta$ is exact. Multiplying by R_ϑ^{T} therefore returns the physical frame, $\mathbf{M}_\vartheta \approx R_\vartheta^{\mathsf{T}} \mathbf{M}^x$, with the moment error transformed without change of magnitude, since R_ϑ is orthogonal. The accuracy attained at $\vartheta = 0$ is thus recovered exactly at every orientation.

5. Results & discussion

The present framework is assessed using a combination of qualitative analysis, polynomial reproduction residuals, derivative error analysis on a test function, modal response, stability analysis, parametric investigation and behaviour in PDEs. To assess the computational efficiency of the proposed framework, we report a wall-clock time analysis as a function of the L_2 error on a test function. Unless otherwise stated, all results are reported for a canonical configuration consisting of a first-order derivative or Laplacian operators, both with second-order consistency. Details on hyper-parameters and the training dataset for each specific model can be found in Appendix B. All experiments are conducted in two spatial dimensions.

5.1. Qualitative analysis of learned operator

To assess the geometric structure of the learned discrete operators, we perform a qualitative analysis of the predicted stencil weights across multiple noisy neighbourhood realisations. Fig. 2 visualises the learned kernel corresponding to two differential operators with the MLP baseline and NeMDO, where weights predicted for many independently perturbed neighbourhoods are overlaid in relative coordinates.

Despite significant stochastic variation in local particle configurations, the learned operators exhibit coherent structures. In Fig. 2, the x-derivative aggregated weight distribution displays approximate anti-symmetry in the direction of the derivative about the origin, i.e. $w(\mathbf{x}_{ji}) \approx -w(-\mathbf{x}_{ji})$, consistent with the structure of a first-derivative operator, while decaying in the transverse direction. This symmetry is not enforced explicitly but emerges purely from the polynomial moment constraints used during training. The overlaid kernels further reveal a decay in weight magnitude with increasing distance from the central particle. Notably, the learned weight magnitudes in NeMDO are lower than those of the MLP baseline. Spatially, the MLP's dominant weights cluster tightly around the origin, whereas NeMDO achieves a smoother distribution throughout the local stencil. A comprehensive investigation into how

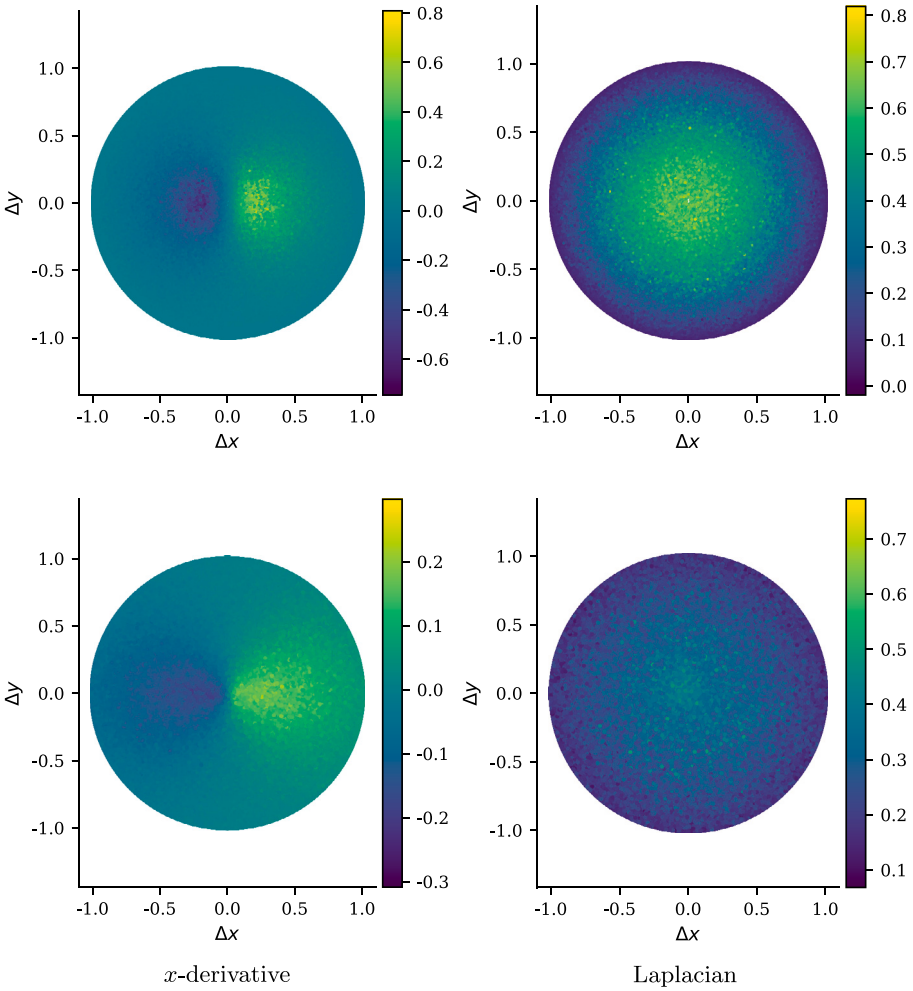


Fig. 2. Second-order normalised learned operators for MLP (top row) and NeMDO (bottom row), colours indicate weight value. Models were trained with particle disturbance of $\epsilon = 1.0$ and inferred with the same noise.

specific optimisation choices or activation functions, optimiser, and architecture influence these weight distributions remains an open avenue for future research, as a full parametric ablation falls outside the scope of the present study.

For the Laplacian operator, the learned kernels instead exhibit rotational invariance, with larger weights concentrated near the central particle and decreasing with distance. Although NeMDO and the MLP display comparable weight magnitudes here, the latter features dominant weights clustered more tightly around the centre. Together, these symmetry patterns and the consistent decay of weight magnitude away from the central particle mirror structural properties commonly observed in gradient and Laplacian kernels of classical mesh-free methods, including uncorrected SPH and polynomially corrected kernel formulations [17,37].

5.2. Polynomial consistency and convergence

We now verify that the learned operators satisfy the polynomial consistency conditions imposed in the training loss and that this translates into accurate differential approximations. For each operator type, we evaluate the learned stencil weights on monomials up to degree p. For brevity, we only show the first derivative results for the x direction, noting that results in the y direction are analogous.

Table 1 reports moment residuals for NeMDO, the MLP and SPH operators, averaged over independently perturbed neighbourhood realisations. For the gradient operator, NeMDO residuals are consistently on the order of 10^{-5} across all first- and second-order monomials, while the MLP are mostly of order 10^{-3}, and the SPH residuals are of order 10^{-2} for both smoothing kernels. For all operators, the Laplacian exhibits larger residuals compared to the derivative ones, with NeMDO on the order of 10^{-4}, the MLP with errors of order 10^{-3}, and the SPH operators with errors of order $10^{-1} - 10^{-2}$. The larger residuals for the Laplacian reflects the increased sensitivity of second-order derivatives to geometric perturbations, a behaviour also observed in classical high-order

Table 1
Moment residuals for NeMDO, MLP and SPH operators. Mean absolute error (MAE) and standard deviation are reported for each monomial moment, averaged over independently perturbed neighbourhood realisations. Residuals quantify the extent to which the learned discrete operators satisfy the imposed Taylor consistency constraints (smaller values are better). The superscripts in the learned operator identify the target operator.

Operator	Metric	x	y	$x^2/2$	xy	$y^2/2$
$\mathrm{NeMDO}^{x}_{p=2}$	MAE	8.84×10^{-5}	7.93×10^{-5}	5.77×10^{-5}	5.59×10^{-5}	4.74×10^{-5}
	St.d.	1.14×10^{-4}	1.04×10^{-4}	7.38×10^{-5}	7.35×10^{-5}	6.25×10^{-5}
$\mathrm{NeMDO}^{4}_{p=2}$	MAE	5.22×10^{-4}	5.09×10^{-4}	3.69×10^{-4}	4.09×10^{-4}	3.62×10^{-4}
	St.d.	6.91×10^{-4}	6.69×10^{-4}	4.94×10^{-4}	5.49×10^{-4}	4.77×10^{-4}
$\mathrm{MLP}^{x}_{p=2}$	MAE	9.30×10^{-4}	1.30×10^{-3}	1.93×10^{-3}	3.16×10^{-3}	1.94×10^{-3}
	St.d.	1.25×10^{-3}	1.75×10^{-3}	2.58×10^{-3}	4.34×10^{-3}	2.57×10^{-3}
$\mathrm{MLP}^{4}_{p=2}$	MAE	1.10×10^{-3}	1.07×10^{-3}	1.09×10^{-3}	1.19×10^{-3}	1.07×10^{-3}
	St.d.	1.52×10^{-3}	1.46×10^{-3}	1.46×10^{-3}	1.57×10^{-3}	1.47×10^{-3}
Wendland $C2^{x}$	MAE	5.14×10^{-2}	4.01×10^{-2}	2.70×10^{-2}	2.63×10^{-2}	1.36×10^{-2}
	St.d.	6.21×10^{-2}	5.01×10^{-2}	3.35×10^{-2}	3.29×10^{-2}	1.71×10^{-2}
Wendland $C2^{4}$	MAE	1.86×10^{-1}	1.83×10^{-1}	5.14×10^{-2}	8.02×10^{-2}	5.26×10^{-2}
	St.d.	2.31×10^{-1}	2.27×10^{-1}	6.21×10^{-2}	1.00×10^{-1}	6.40×10^{-2}
Quintic Spline^{x}	MAE	3.44×10^{-2}	2.42×10^{-2}	1.98×10^{-2}	2.09×10^{-2}	1.10×10^{-2}
	St.d.	4.24×10^{-2}	3.03×10^{-2}	2.46×10^{-2}	2.59×10^{-2}	1.38×10^{-2}
Quintic Spline^{4}	MAE	8.49×10^{-2}	7.99×10^{-2}	3.44×10^{-2}	4.85×10^{-2}	3.35×10^{-2}
	St.d.	1.05×10^{-1}	9.91×10^{-2}	4.24×10^{-2}	6.05×10^{-2}	4.15×10^{-2}

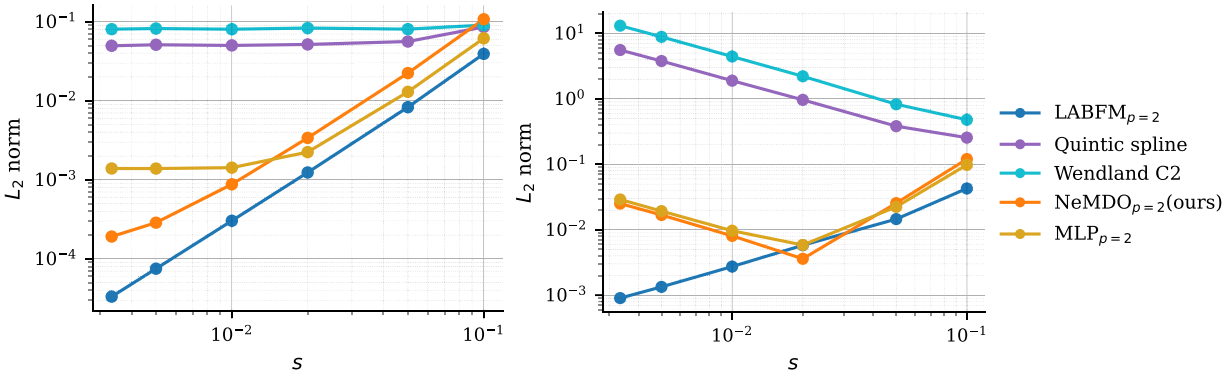


Fig. 3. Convergence of the discrete x-derivative (*left*) and Laplacian (*right*) on a smooth test function with $\epsilon = 0.5$, noting that NeMDO was trained with $\epsilon = 1.0$. Relative L_2 error versus particle spacing s for the learned NeMDO operator, a second-order LABFM operator, two uncorrected SPH kernels (quintic spline and Wendland C2), and MLP.

mesh-free and spectral discretisations [71]. Overall, these results indicate that the polynomial consistency constraints can be learnt with a self-supervised framework, generalise for unseen stencil arrangements, and significantly outperform SPH operators.

Next, we evaluate the learned operators on a smooth test function with known analytical derivatives. We considered a square domain defined by $(x, y) \in [-0.5, 0.5]^2$, and we define the test function as

$$\phi(\tilde{x}, \tilde{y}) = 1.0 + (\tilde{x}\tilde{y})^4 + \sum_{n=1}^{6}(\tilde{x}^n + \tilde{y}^n), \tag{27}$$

where $\tilde{x} = x - 0.1453$ and $\tilde{y} = y - 0.16401$. This choice of test function with pseudo-random offset ensures asymmetry in the function, to prevent the masking of errors, which could cancel for a symmetric function (e.g. Fourier modes) [17].

Relative L_2 errors between the predicted and exact derivatives are reported at varying average particle spacing s. The learned operators are compared against classical discretisations, including SPH kernels based with the quintic spline and Wendland C2, as well as a representative formally consistent method, LABFM.

Fig. 3 reports the convergence behaviour of the derivative and Laplacian. All convergence results are computed at a particle disorder level of $\epsilon = 0.5$ given SPH convergence is known to degrade significantly at higher disorder levels [13]. For the gradient, NeMDO$_{p=2}$ and the MLP$_{p=2}$ baseline closely follow the second-order LABFM$_{p=2}$ scheme, exhibiting a decay rate consistent with second-order convergence as $s \to 0$. Across all resolutions, the learnt and LABFM operators substantially outperform the SPH kernel. At lower resolutions, the MLP baseline achieves superior accuracy compared to the graph-based framework; however, it plateaus at a limiting error approximately one order of magnitude higher. For most mesh-free operators (including SPH and LABFM), achieving the expected convergence behaviour requires a sufficiently smooth underlying function, moderate particle disorder, and an adequate

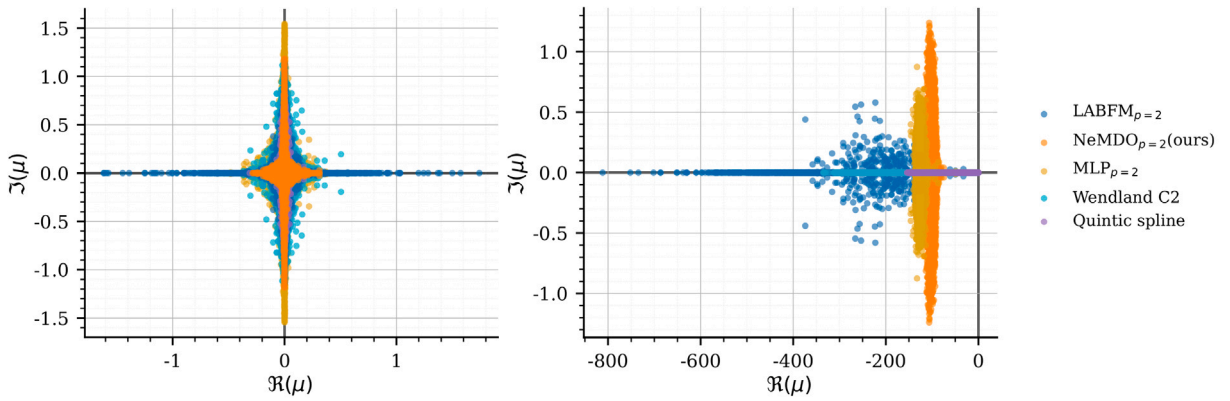


Fig. 4. Normalised eigenvalue spectrum of the global x-derivative operator with node disturbance $\epsilon = 1.0$ and 2500 nodes. *Left*: x-derivative operators; *right*: Laplacian operators.

support size. Under these conditions, the limiting accuracy in consistent methods is typically bounded by the accuracy of the solution of the linear system [56]. NeMDO is not a formally consistent method (polynomial consistency is learnt, not enforced); thus, as resolution increases, the dominant limiting factor becomes the first-order moment residuals, since higher-order moment terms are scaled by higher powers of the spacing and decay faster — leaving the first-order residuals as the error floor, reported in Table 1. The comparatively poor SPH performance is attributed to the high particle disorder and the complexity of the test function, both of which are known to degrade uncorrected kernel approximations [13].

For the Laplacian, LABFM exhibits the expected first-order convergence, achieving the lowest errors over the range of resolutions considered. The learned Laplacian displays a decay faster than first-order at coarse resolutions; however, this behaviour is only observed over a small range of resolutions, and while first-order convergence is expected, we observe faster convergence because higher-order truncation error terms dominate the error when s is large (this effect is strongly dependent on the choice of ϕ). As resolution increases, the error saturates at a limiting value of order 10^{-3}, after which the error grows with order s^{-1}, similarly to traditional SPH formulations [72]. This behaviour is consistent with the presence of a residual consistency error in the Laplacian operator, with similar divergence observed in formally consistent mesh-free methods and spectral SPH formulations, albeit at finer resolutions [62,71]. Despite this limitation, the learned Laplacian remains significantly more accurate than uncorrected SPH kernels across the entire range of s. Remarkably, for the Laplacian, the MLP achieves similar limiting accuracy to the graph-based architecture.

Overall, these results suggest that learning to approximate polynomial consistency provides meaningful improvements over traditional SPH and exhibits similar behaviour to the formally consistent method within a finite resolution range.

5.3. Stability

To analyse the stability of the discrete operators, we construct a global discrete derivative matrix $\mathbf{G}^D$ as a re-arrangement of the local operators L_i^D. The global discrete operator matrix acts as a mapping that describes how a global field ϕ evolves across the *entire domain* simultaneously, and the eigenvalues of $\mathbf{G}^D$ provide insight into the stability properties of the discretisation. Eigenvalues on the imaginary axis correspond to advective (translational) modes, while eigenvalues with nonzero real parts indicate growth or decay. These are of particular interest in dynamical systems, where the spectral distribution of $\mathbf{G}^D$ dictates the long-term behaviour of the numerical solution.

For convective derivatives (i.e. first-order spatial derivatives), the continuous operator is purely dispersive, with Fourier modes corresponding to translation and no amplification or diffusion. Accordingly, the eigenvalues μ of a stable discrete approximation should ideally lie on the imaginary axis, i.e. $\Re(\mu) = 0 \; \forall \; \mu$. In contrast, the Laplacian operator is purely dissipative, i.e. $\Re(\mu) \leq 0 \;\; \forall \; \mu$, reflecting the decay of all modes due to diffusion [14].

Fig. 4 (*left*) shows the normalised eigenvalues of the discrete x-derivative operator on a noisy particle distribution. NeMDO's (graph-based architecture) spectrum is tightly clustered near the imaginary axis, with a comparable spread to the quintic spline, and lies closer to the imaginary axis than that of LABFM constructed with the same neighbourhood size $|\mathcal{N}_i|$ and the Wendland C2 kernel. The MLP baseline exhibits a spectral spread along the real axis similar to that of the Wendland C2 kernel, while its imaginary components are larger in magnitude than those of any other operator. Given that high-order discretisations are known to generate small-scale oscillations [73], it is notable that NeMDO and the MLP baseline — both trained to approximate second-order polynomial consistency — exhibits stability properties comparable to the quintic spline and Wendland C2, respectively. This suggests that the learned operator achieves competitive stability characteristics per neighbour than both corrected and uncorrected kernel-based baselines.

For the Laplacian, no eigenmodes exhibit growth in time ($\Re(\mu) \leq 0 \;\; \forall \, \mu$) for any of the operators considered. The Morris Laplacian [60] exhibits only real components, corresponding to purely diffusive behaviour, which likely contributes to its widespread

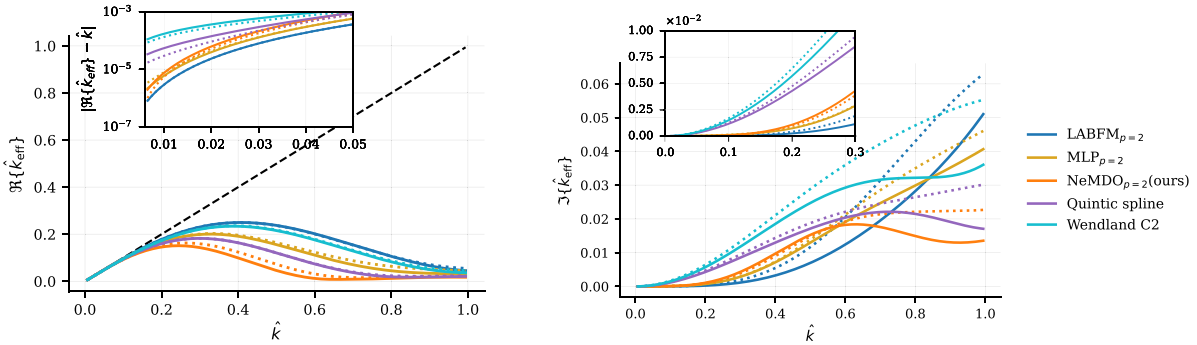


Fig. 5. Resolving power of the $x-$derivative for LABFM, NeMDO, MLP, and SPH (with the Quintic Spline and Wendland C2). The *left* plot shows the real part of the modal response, and the *right* plot shows the imaginary part on disordered particle distributions with noise level $\epsilon = 1.0$. The horizontal axis shows the true wavenumber k normalised by the Nyquist wavenumber k_{Ny}, while the vertical axis shows the corresponding effective wavenumber $k_{\mathrm{eff}}/k_{\mathrm{Ny}}$. The dashed black line indicates the ideal response, corresponding to a spectral method. The inset on the *left* plot shows the absolute difference between the different operators and the ideal spectral response. The inset on the *right* plot shows zoomed in graph of $\Im\{\hat{k}_{\mathrm{eff}}\}$ for range $0 \leq \hat{k} \leq 0.3$. The full and dotted lines indicate $k_y/k_x = 0$ and $k_y/k_x = 1$, respectively.

use and robustness in SPH simulations. NeMDO's Laplacian displays a larger spread along the imaginary axis than the SPH and LABFM baselines; however, its eigenvalues are more tightly clustered overall, indicating more uniform damping of modes. Similarly, the MLP baseline shows real eigenvalues clustered, with a large spread in the imaginary axis. Overall, the learned operators demonstrate a more uniform spectral response across modes in disordered particle configurations for both first- and second-order derivatives, although a large dispersiveness is observed in the Laplacian. It should be noted that the results presented in this section are with particle disorder of $\epsilon = 1.0$, which represents the worst-case scenario regarding node distribution compared to standard simulations that traditionally use particle shifting [64–66].

5.4. Modal response

Insight into the behaviour of the derivative operators can be gained from an analysis of the modal response [74]. In principle, spectral discretisations can exactly represent all modes below the Nyquist limit [75]—where the Nyquist wavenumber is given by $k_{Ny} = \pi/s_i$—however, truncated discretisations only approximate the underlying function up to a finite truncation error, leading to a modified (effective) wavenumber response.

The modal response can be analysed by setting $\phi = \phi(\mathbf{x}_{ji})$ to be a local (to each stencil) multi-dimensional Fourier series; in our case, we restrict the analysis to a two-dimensional domain

$$\phi(x, y) = \sum_{k_x \in \mathbb{Z}} \sum_{k_y \in \mathbb{Z}} c_{k_x,k_y} e^{\mathrm{i}(k_x x + k_y y)}, \tag{28}$$

where the c_{k_x,k_y} are complex constants and k_x, k_y are (appropriately scaled) wavenumbers. The effective wavenumber is then $k_{\mathrm{eff}} = L_i^D(\phi)$ representing the wavenumber captured by the discrete operator. For the first derivative with respect to x, the effective wavenumber in the ith stencil is given by

$$k_{\mathrm{eff}} = \sum_{j \in \mathcal{N}_i} \sin(k_x x_{ji} + k_y y_{ji}) w_{ji}^x + \mathrm{i} \sum_{j \in \mathcal{N}_i} (1 - \cos(k_x x_{ji} + k_y y_{ji})) w_{ji}^x. \tag{29}$$

The real part of Eq. (29) characterises the numerical scheme's dispersion error; specifically, an error exists whenever $\Re\{k_{\mathrm{eff}}\} \neq k_x$. Similarly, the imaginary part represents dissipation error, occurring if $\Im\{k_{\mathrm{eff}}\} \neq 0$. For the Laplacian, the effective wavenumber is given by

$$q_{\mathrm{eff}}^2 = \sum_{j \in \mathcal{N}_i} (1 - \cos(k_x x_{ji} + k_y y_{ji})) w_{ji}^{\Delta} - \mathrm{i} \sum_{j \in \mathcal{N}_i} \sin(k_x x_{ji} + k_y y_{ji}) w_{ji}^{\Delta} \tag{30}$$

where $q_{\mathrm{eff}}^2 = k_x^2 + k_y^2$ for spectral methods. In the equation above, the real part of q_{eff}^2 relates to the dissipation, and the imaginary part to the dispersion error.

Fig. 5 shows the modal response of the different discretisations for the x-derivative operator, and Fig. 6 for the Laplacian. For both differential operators, all methods recover the true modal response at low wavenumbers and progressively deviate as the true wavenumber increases, indicating larger errors for high-wavenumber modes. In both cases, LABFM$_{p=2}$ remains closest to the spectral reference, followed by the Wendland C2 kernel. For the x-derivative, the MLP baseline ranks ahead of the quintic spline, whereas for the Laplacian the ordering reverses. The GNN and MLP operators exhibit near-identical resolving power for the Laplacian, consistent with the comparable convergence rates observed in Fig. 3 (right). The insets show that at low wavenumbers NeMDO outperforms SPH by approximately two orders of magnitude for the real part of the x-derivative (Fig. 5, *left*) and one order for the real part

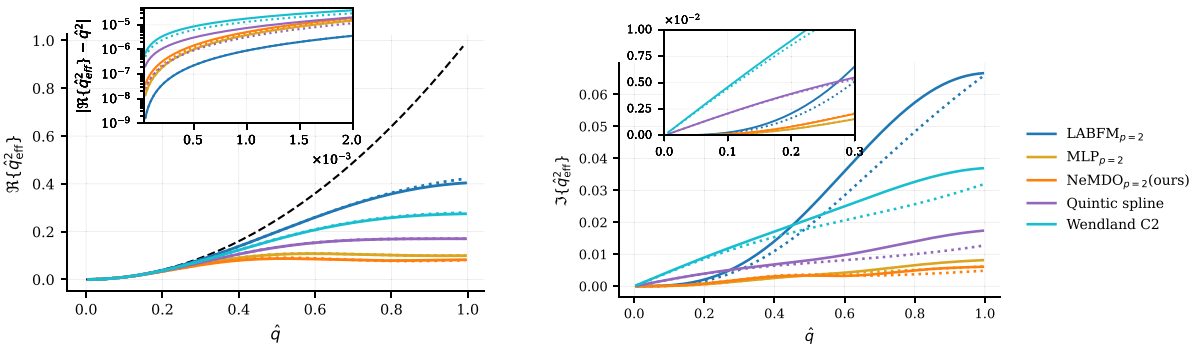


Fig. 6. Resolving power of the Laplacian for LABFM, NeMDO, MLP, and SPH (with the Quintic Spline and Wendland C2). The *left* plot shows the real part of the modal response and the *right* plot shows the imaginary part on disordered particle distributions with noise level $\epsilon = 1.0$. The horizontal axis shows the true wavenumber q normalised by the Nyquist wavenumber k_{Ny}, while the vertical axis shows the corresponding effective wavenumber. The dashed black line indicates the ideal response, corresponding to a spectral method. The inset on the *left* plot shows the absolute difference between the different operators and the ideal spectral response. The inset on the *right* plot shows zoomed in graph of $\Im\{\hat{q}^2_{\text{eff}}\}$ for range $0 \leq \hat{q} \leq 0.3$. The full and dotted lines indicate $k_y/k_x = 0$ and $k_y/k_x = 1$, respectively.

of the Laplacian (Fig. 6, *left*). Although the SPH kernels exhibit a stronger modal response than NeMDO for higher wavenumber, this spectral advantage is confined to sufficiently low resolutions and did not translate into smaller error during the convergence study in practice due to their pronounced sensitivity to node disorder shown in Fig. 3 and the relatively fine resolutions plotted. The imaginary parts of the differential operators indicate that NeMDO has smaller dissipation and dispersion errors in the $x-$derivative and the Laplacian, respectively, compared to the other operators.

The global spectral analysis (Section 5.3) and the local modal response (Section 5.4) provide complementary perspectives on the numerical performance of these operators regarding dispersion and dissipation. While both analyses reflect the influence of particle irregularity, they do so at different scales of aggregation. The modal response characterises the local fidelity of the operator, demonstrating that NeMDO weights are effectively optimised to suppress dispersion error on a per-stencil basis, even under high node disorder. Conversely, the global spectrum captures the matrix non-normality of the assembled system. The large imaginary values in the NeMDO and LABFM Laplacian (Fig. 4) are not a result of local dispersive failure — which the modal response proves is minimal — but rather a manifestation of the global matrix non-normality. This arises from the lack of reciprocity between neighbours in a disordered mesh-free assembly ($w_{ij} \neq w_{ji}$). The bidirectional edges of the star graph in Section 4 operate within a single stencil; they do not connect one stencil to another. The network is evaluated independently per stencil, so node i never observes the neighbourhood of j, and the edge (i, j) is predicted twice from two different input geometries with nothing tying the two weights together. The resulting asymmetry is generic to consistent mesh-free operators constructed independently at each node: in LABFM and corrected SPH [76] the same edge is built twice, from two separate local systems, and thus the imaginary spread is observed. The Morris operator used for the SPH baseline is the exception: its weights depend only on the pair separation and the neighbour volume, and are therefore reciprocal for the uniform-volume distributions considered here, consistent with the purely real Laplacian spectrum in Fig. 4. Reciprocity in that case is obtained by construction, at the cost of polynomial consistency.

Reciprocity could in principle be encouraged through a symmetry penalty, or enforced by predicting a single weight per shared edge. Symmetric weights render the global Laplacian symmetric, so either measure would be expected to reduce the observed spread. However, in this case, a shared edge weight must satisfy the moment conditions at both endpoints, whose stencils generally differ, so reciprocity can be obtained only at the cost of consistency or by coupling stencils and abandoning the strictly local, independently evaluated construction. Quantifying this trade-off is left for future work.

5.5. Computational cost and accuracy

We now evaluate the cost–accuracy trade-off of NeMDO relative to LABFM and SPH. For each configuration, we measure the wall-clock time required to compute stencil weights for the x-derivative on a fixed particle cloud and evaluate the resulting L_2 error of the test function in Eq. (27), further details regarding the computational cost can be found in Appendix C. All timings reported here are performed in single precision for every method, so that the comparison is not confounded by differing floating-point formats; at the resolutions considered, single precision is not the limiting factor on the reported errors. We report results for NeMDO$^1_{p=2}$, NeMDO$^2_{p=2}$, NeMDO$^3_{p=2}$, where the superscripts identifies increasing numbers of trainable parameters, to examine how computational cost and limiting accuracy scale with model capacity.

Fig. 7 shows that NeMDO$^1_{p=2}$ achieves substantially lower error than the SPH kernels at comparable cost, and yields a speedup of roughly 3 times over LABFM$_{p=2}$ up to its limiting error of 4×10^{-3}. NeMDO$^2_{p=2}$ reaches an error floor of 5×10^{-4}, roughly 3 orders of magnitude below the SPH baselines, at a cost 2× slower than LABFM$_{p=2}$. Increasing the number of network parameters further in NeMDO$^3_{p=2}$ reduces the error floor again, at the expense of increased forward time. It is important to note that all NeMDO

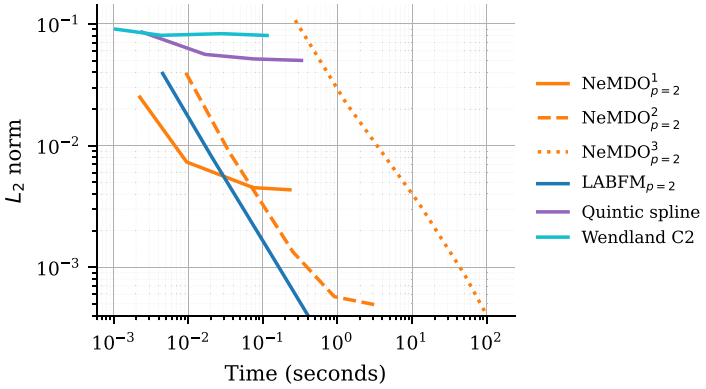


Fig. 7. Evaluation of the trade-off between parameter count, wall-clock forward time, and accuracy for NeMDO and baseline kernel methods when computing stencil weights. All methods are timed in single precision. The vertical axis shows the L_2 error of the x-derivative of the test function in Eq. (27) at a different resolutions; horizontal lines mark the resolution at which each operator converges (i.e., further refinement does not reduce this error). All NeMDO models are trained with geometric noise $\epsilon = 1.0$, while the convergence tests are performed at $\epsilon = 0.5$.

models in Fig. 7 were trained with geometric noise of magnitude $\epsilon = 1.0$, so they retain their accuracy for particle disorder up to this level. The figure reports results for $\epsilon = 0.5$, which is approximately the highest noise level at which the SPH kernels still exhibit convergence, while LABFM$_{p=2}$ (and other consistent methods) maintain the same convergence for particle disorder from $\epsilon = 0 - 1.0$ for second-order approximations [17]. Additional experiments (Section 6.3) indicate that training with milder noise makes the learning problem simpler, yielding lower errors for the same architecture. In this sense, the cost–accuracy curves in Fig. 7 provide a conservative, worst-case characterisation of NeMDO's performance. Taken together, these results show that NeMDO limiting error varies with model, with the smallest model showing significant speed-up compared to LABFM at substantially higher accuracy than SPH uncorrected kernels. All speedup factors quoted here are relative to LABFM$_{p=2}$, i.e. at matched order of approximation; the Taylor–Green vortex simulations in Section 5.6.2 use LABFM$_{p=4}$, which carries a larger stencil and linear system, so these factors should not be transferred to that case.

5.6. Application to partial differential equations

5.6.1. Poisson's equation

To assess the effectiveness of the learned operators in practical settings, we use them to solve the Poisson equation in a mesh-free setting. The Poisson equation is given by

$$\nabla^2 \phi = f(x, y) \tag{31}$$

in the domain Ω, with Dirichlet boundary conditions

$$\phi = g_1(x, y) \quad \text{on } \Gamma_D. \tag{32}$$

We assemble a global discrete Laplacian matrix $\mathbf{G}^{\boldsymbol{\Delta}} \in \mathbb{R}^{N \times N}$, where N is the total number of nodes, by distributing each local operator L_i^{Δ} into the corresponding row. Specifically, the ith row of $\mathbf{G}^{\boldsymbol{\Delta}}$ is constructed by mapping the entries of L_i^{Δ} to the columns associated with the neighbours in $\mathcal{N}_i$:

$$G^{\Delta}_{i,j} = w^{\Delta}_{ji} \quad \forall\, j \neq i \tag{33a}$$

$$G^{\Delta}_{i,i} = -\sum_j w^{\Delta}_{ji}. \tag{33b}$$

The discretised form of Eq. (31) is

$$\mathbf{G}^{\boldsymbol{\Delta}} \boldsymbol{\Phi} = \boldsymbol{F} \tag{34}$$

with the solution vector

$$\boldsymbol{\Phi} = \left[\phi_1,\ \phi_2,\ \ldots,\ \phi_N\right]^T \tag{35}$$

and source vector

$$\boldsymbol{F} = \left[f_1,\ f_2,\ \ldots,\ f_N\right]^T. \tag{36}$$

The global linear system, Eq. (34), is solved iteratively via the stabilised bi-conjugate gradient algorithm with Jacobi preconditioning. Since the solution is obtained implicitly, boundary conditions must be enforced directly within the system. Dirichlet conditions are

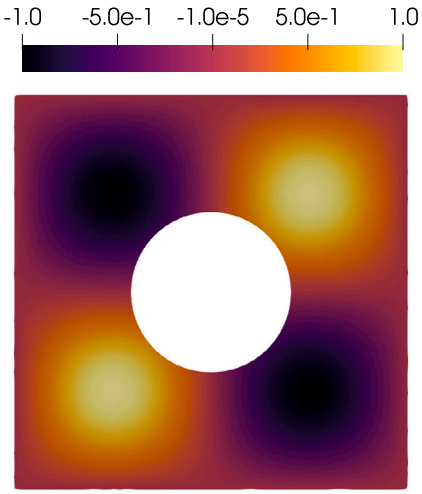


Fig. 8. Analytical solution to the Poisson problem.

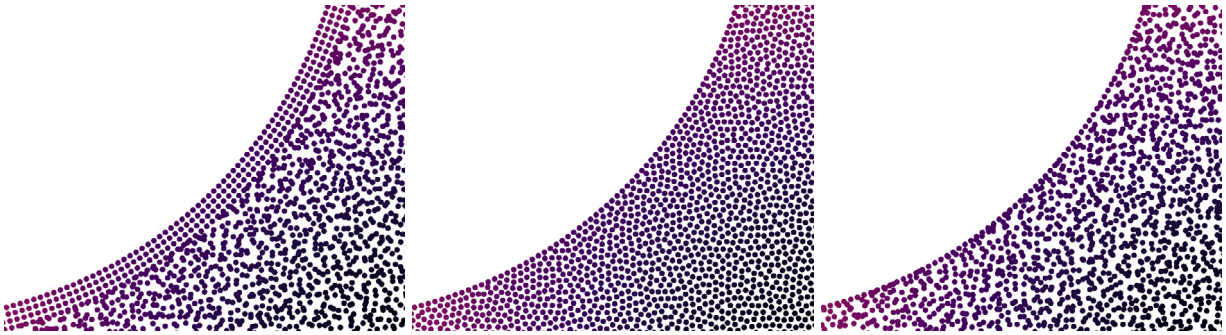


Fig. 9. Close-up of the (left) uniformly distributed, (centre) particle-shifted, (right) and noisy boundary nodes along the cylinder surface.

imposed by modifying the relevant rows and columns of $\mathbf{G}^{\Delta}$ and $\boldsymbol{F}$. Specifically, for a boundary node b,

$$G^{\Delta}_{b,j} = 0 \quad \forall\, j \neq b \tag{37a}$$

$$G^{\Delta}_{b,b} = 1 \tag{37b}$$

$$F_b = g_{1,b}, \tag{37c}$$

which ensures that the solution to the global linear system satisfies $\phi_b = g_{1,b}$. Neumann boundary conditions are not considered in the present work, but can be accommodated using ghost nodes to complete near-boundary stencils, following the approach described in King et al. [17].

We consider a square domain $(x, y) \in [0, 1]^2$ with a circular hole of radius $r = 0.2$ centred at $(0.5, 0.5)$. The source term is set to $f := \sin(2\pi x)\sin(2\pi y)$, with Dirichlet boundary conditions prescribed from the corresponding analytical solution on the cylinder boundaries. We then assess NeMDO's robustness by varying the boundary node arrangement. Three node distributions are considered: (i) interior nodes on a perturbed Cartesian lattice with $\epsilon = 1.0$ and uniform spacing along the cylinder boundary, (ii) the same interior distribution with noisy boundary placement ($\epsilon = 1.0$), and (iii) all nodes repositioned via the iterative particle-shifting algorithm of Flyer et al. [67]. The same trained model is deployed across all three distributions without retraining, allowing us to evaluate robustness across different point clouds. For each configuration, we compare NeMDO trained with complete stencils only against the augmented variant trained with the one-sided stencil procedure described in Section 4. Fig. 8 shows the analytical solution, and Fig. 9 provides a close-up of the cylinder boundary and the different particle arrangements tested.

Fig. 10 presents the pointwise error for each discrete operator at two resolutions for the uniform node arrangement at the cylinder boundary. At the lower resolution (top row), LABFM, the Wendland C2 kernel, and the augmented NeMDO model produce qualitatively similar error distributions, each showing a four-quadrant structure. The Wendland C2 error field is noticeably rougher, which we hypothesise is due to the lack of polynomial consistency in the uncorrected kernel weights: without consistency, local truncation errors vary incoherently across the domain in unstructured node distributions, rather than following a smooth spatial

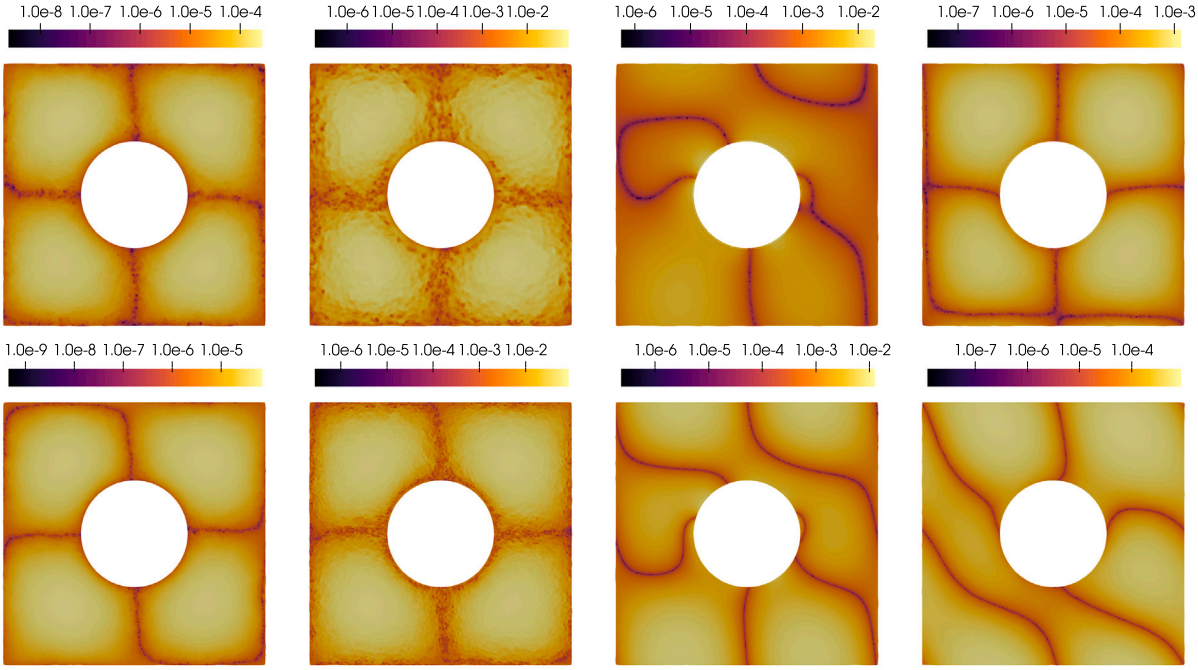


Fig. 10. Pointwise error for the Poisson problem with uniform boundary nodes. Top row: $s = 6 \times 10^{-3}$; bottom row: $s = 3 \times 10^{-3}$. From left to right: LABFM, Wendland C2, NeMDO, and augmented NeMDO.

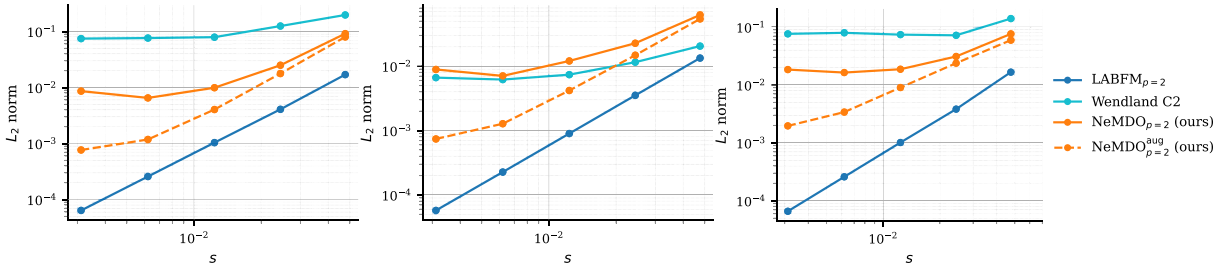


Fig. 11. L_2 convergence for the Poisson problem with uniform (left), particle-shifted (centre), and noisy (right) boundary node distributions.

pattern. The augmented NeMDO and LABFM exhibit a minor break of symmetry at the lower resolution. Peak errors are of order 10^{-4} for LABFM, 10^{-3} for augmented NeMDO, and 10^{-2} for both the Wendland C2 kernel and standard NeMDO. At the higher resolution (bottom row), LABFM and the Wendland C2 kernel both display smooth, nearly symmetric error distributions. The peak error decreases between the two resolutions for LABFM and augmented NeMDO, while the Wendland C2 kernel and standard NeMDO show no improvement.

We note that the augmented NeMDO error is nearly symmetric at the lower resolution but loses this symmetry at the higher resolution. At coarse resolution the total error is dominated by the discretisation error, which inherits the spatial structure of the solution and is therefore symmetric. As the resolution increases, the discretisation error decreases and the residual becomes dominated by the weight-approximation error of the network, which can exhibit stencil-to-stencil variability and does not respect the symmetry of the problem. LABFM and the Wendland C2 kernel both retain symmetric error distributions at both resolutions. Although the Wendland C2 kernel has large absolute error, it has low stencil-to-stencil variability compared to NeMDO at the limiting resolution, so nodes at symmetric locations receive similar errors and the large-scale symmetric pattern is preserved.

Fig. 11 shows the L_2 norm of the predicted solutions relative to the analytical solution for the three boundary configurations. With uniformly distributed boundary nodes (left), the Wendland C2 kernel performs the poorest, plateauing near the upper end of 10^{-2}. Standard NeMDO also stagnates, near the upper end of 10^{-3}, while the augmented variant improves by roughly an order of magnitude, settling near the upper end of 10^{-4}. LABFM exhibits convergent behaviour across all tested resolutions, and produces practically identical results across all three boundary configurations. The noisy boundary case (right) shows little change relative to the uniform results, with only a slight increase in limiting error for both NeMDO variants. In the particle-shifted case (centre), the Wendland C2 kernel improves by roughly an order of magnitude, benefiting from the more regular node spacing produced

by the shifting algorithm, while both NeMDO variants show convergence behaviour practically identical to the uniform case. Overall, these results suggest that NeMDO is robust to the boundary node arrangement. A broader investigation of how the training particle distribution affects learning is presented in Section 6.3. The boundary treatment demonstrated here is scoped to a single smooth curved Dirichlet boundary; corners, mixed boundary conditions and material interfaces are not considered. Extension of the one-sided augmentation to more general boundary configurations is left for future work.

5.6.2. Navier–Stokes equations

Our next PDE is the weakly compressible Navier–Stokes equations in an Eulerian unstructured node distribution. While SPH is generally implemented in Lagrangian schemes, particle clustering and instabilities can significantly affect the quality of the simulation [59,77,78]. Our objective in this work is to assess the quality of the differential operator; thus, we restrict our investigation to Eulerian simulations, and consider the two-dimensional Taylor–Green vortex.

Weakly compressible SPH is known to generate pressure oscillations; thus, to improve numerical stability we dealiase the solution with a high-order filter at each time-step [79]. Furthermore, high-order collocated methods are known to frequently lead to spurious oscillations at small scales [73,80]; thus, we also use a high-order filter in LABFM, and for our framework we train a hyperviscous operator $\text{NeMDO}^{\text{hyp}}_{p=4}$. It is worth noting that the LABFM-computed filters used by both SPH and LABFM satisfy the moment conditions exactly via a linear solve, whereas the NeMDO hyperviscous operator only approximately enforces the desired filtering. Consequently, NeMDO operates with a weaker stabilisation than the classical baselines, making the comparison conservative with respect to NeMDO's performance. NeMDO is also fully compatible with externally computed filters from any mesh-free framework, which can be applied as a drop-in replacement for the learned hyperviscous operator. The high-order filter is a fourth-order biharmonic operator $-\alpha\nabla^4$. The coefficient α_i is calibrated a priori at each stencil so that a tensor-product mode $\cos(\kappa_i x)\cos(\kappa_i y)$ at two-thirds of the Nyquist wavenumber is damped to one-third of its amplitude per application, giving $\alpha_i = (2/3)\left[\sum_j \left(1-\cos(\kappa_i x_{ji})\cos(\kappa_i y_{ji})\right) w^{\text{hyp}}_{ji}\right]^{-1}$, where $\kappa_i = (2/3)\pi/\ell_i$ is the target wavenumber and $\ell_i = 2h_i\sqrt{\pi/|\mathcal{N}_i|}$ is the local length scale. The filtered field is ultimately given by $F(\phi) = \left(1-\beta\alpha_i\nabla^4\right)\phi$, where β is 1, 0.1 and 0.02 for uncorrected SPH, LABFM, and NeMDO, respectively. These scalings were selected individually for each discretisation to maximise accuracy while ensuring stability.

Governing Equations: We solve the two-dimensional compressible Navier–Stokes equations in conservative form

$$\frac{\partial \rho}{\partial t} + \frac{\partial \rho u_k}{\partial x_k} = 0 \tag{38a}$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_k}{\partial x_k} = -\frac{1}{\text{Ma}^2}\frac{\partial \rho}{\partial x_i} + \frac{1}{\text{Re}}\frac{\partial^2 u_i}{\partial x_k \partial x_k} \tag{38b}$$

where ρ is the density, u_i is the ith component velocity, and Re is the Reynolds number, and Ma is the Mach number. The system is closed with a barotropic equation of state absorbed into the momentum equation. The analytical solution of the TGV for incompressible flow is given by:

$$u(x,y,t) = -\exp\left(-\frac{8\pi^2}{\text{Re}}t^*\right)\cos(2\pi x)\sin(2\pi y) \tag{39a}$$

$$v(x,y,t) = \exp\left(-\frac{8\pi^2}{\text{Re}}t^*\right)\sin(2\pi x)\cos(2\pi y), \tag{39b}$$

on the spatial domain $(x, y) \in [0, 1]^2$ equipped with periodic boundary conditions. In the equation above, u and v denote the velocity components in the x- and y-directions, respectively, which we use as a reference. In the experiments carried in this section, we used a Reynolds number of 100 and Mach number of 0.1. In the weakly compressible regime, the numerical discretisation error dominates the deviation introduced by compressibility effects; therefore, the incompressible solution can be used as a Ref. [62]. The collocation points were distributed using the iterative particle shifting algorithm of [67], which iteratively relocates nodes from regions of high density to regions of low density, yielding a more uniform spatial distribution. Notably, this distribution differs from the randomly perturbed grids used during training (see Table 6), and the results therefore demonstrate that the learned operators generalise to unseen point-cloud statistics.

Fig. 12 shows snapshots of the velocity magnitude of the TGV at different time units for the SPH operator with Wendland C2 kernel, $\text{NeMDO}_{p=2}$, and $\text{LABFM}_{p=4}$. We simulate the TGV with the SPH operator and NeMDO for coarse and fine resolutions to test the convergence, while $\text{LABFM}_{p=4}$ as a representative high-order mesh-free discretisation.

Qualitatively, NeMDO outperforms SPH for both coarse and fine resolutions. While SPH at $s = 1/70$ shows asymmetry in the flow at 3 time units, our operator at the same resolution only shows asymmetry after 10 time units. At higher resolutions, our operator shows a significant improvement in capturing the large flow structure compared to the lower resolution, indicating convergence. SPH does not show significant improvement when increasing resolution. LABFM outperforms both methods, which is expected due to the fourth-order accuracy enforced. These results agree with the limiting errors observed earlier in the convergence study, showing SPH low limiting error, while NeMDO shows a clear convergent behaviour and significant improvement over SPH.

To perform a quantitative assessment of the results, we compute the error between the analytical solution and the discrete operators with a relative root mean squared error (relative L_2 error) based on the velocity magnitude, shown in Fig. 13. These results confirm the observations conducted in the qualitative analysis, where both SPH and NeMDO exhibit convergent behaviour with increasing resolution; however, the SPH error saturates at order 10^{-3}, whereas NeMDO achieves errors approximately one order

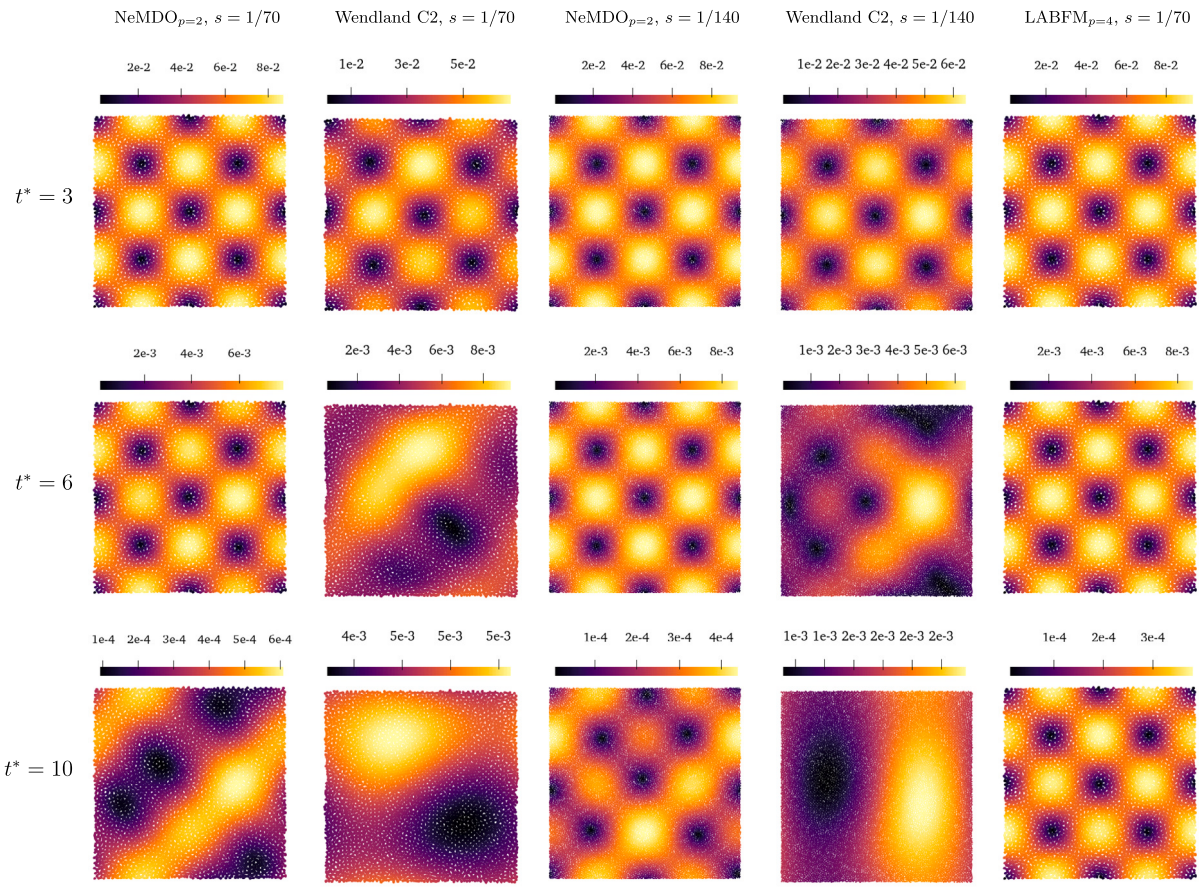


Fig. 12. Velocity magnitude for the Taylor–Green vortex with different operators at different resolutions. Colour axes are rescaled at different time scales for better visualisation of the flow structures.

of magnitude smaller across the tested resolutions. Indeed, the observed performance of NeMDO is consistent with the operator-level accuracy and stability diagnostics reported earlier, indicating that the learnt operators can be directly embedded within a compressible flow solver and used as a standalone spatial discretisation without problem-specific tuning. In contrast, Eulerian SPH requires either high-order filtering (provided by a high-order consistent method) or kernel correction to obtain stable solutions. Because NeMDO is trained without access to PDE solution data, the learnt operators can be deployed in a zero-shot manner at the differential operator level in mesh-free simulations, enabling direct reuse within PDE solvers without retraining or further changes in the code.

6. Parametric study

To assess the robustness and consistency of the proposed operator construction, we conduct a parametric study examining the influence of: (i) stencil size, (ii) Taylor truncation, (iii) particle disorder, and (iv) rotational robustness.

6.1. Stencil size

We first study how the accuracy of the learned operators depends on the stencil size $|\mathcal{N}_i|$. Table 2 reports the average moment error and standard deviation for x-derivative operators as $|\mathcal{N}_i|$ is varied for disordered node distribution, while keeping the network architecture and all other training parameters fixed.

Table 2 shows that increasing $|\mathcal{N}_i|$ from 10 up to 50 neighbours reduces the average moment error, this reflects the improved conditioning of the local Taylor system when more neighbours are included (indicated by the smaller weights in Fig. 14). However, further increasing the stencil size to $|\mathcal{N}_i| = 100$ does not lead to additional gains. This suggests the presence of an optimal stencil size, beyond which particle noise, network capacity, and training dataset size, rather than the number of neighbours, become the

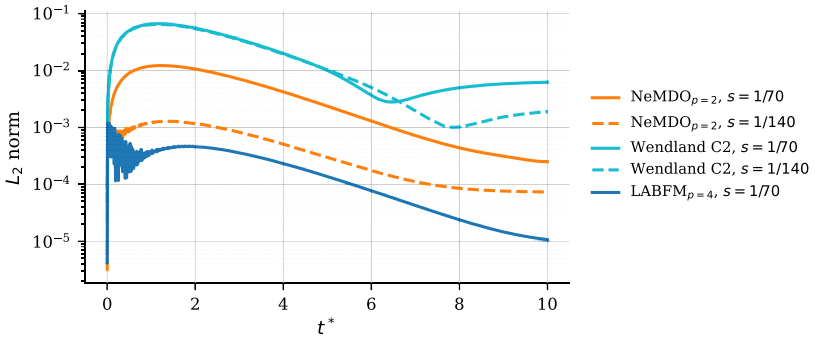
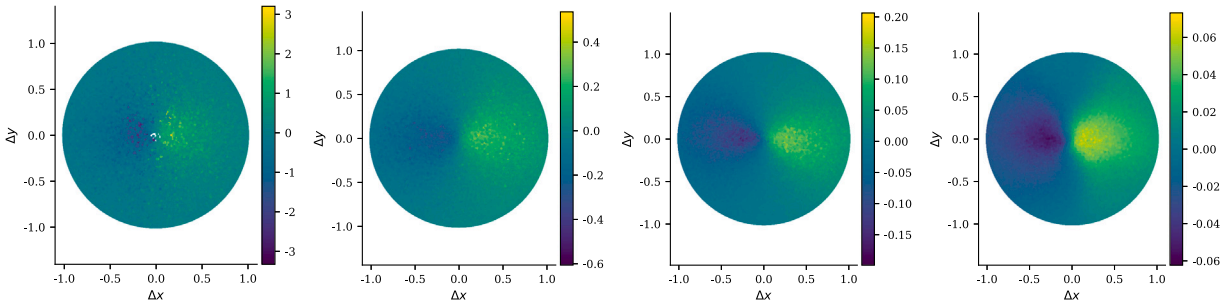


Fig. 13. Relative root mean squared error of different operators with respect to the analytical solution (Eq. (39)) for velocity. s indicates the average particle spacing, and t^* indicates characteristic time units of the flow.

Table 2
NeMDO moment residuals for x-derivative operators with varying stencil size $|\mathcal{N}_i|$. Mean absolute error (MAE) and standard deviation are reported for each monomial moment, averaged over independently perturbed neighbourhood realisations. Residuals quantify the extent to which the learned discrete operators satisfy the imposed Taylor consistency constraints (smaller values are better).

$\lvert\mathcal{N}_i\rvert$	Operator	Metric	x	y	$x^2/2$	xy	$y^2/2$
10	NeMDO$_{p=2}$	MAE	6.44×10^{-3}	6.58×10^{-3}	9.56×10^{-3}	9.97×10^{-3}	9.95×10^{-3}
		St.d.	9.89×10^{-3}	1.05×10^{-2}	1.76×10^{-2}	1.70×10^{-2}	1.63×10^{-2}
25	NeMDO$_{p=2}$	MAE	7.84×10^{-4}	8.12×10^{-4}	5.23×10^{-4}	6.47×10^{-4}	6.29×10^{-4}
		St.d.	1.06×10^{-3}	1.10×10^{-3}	7.21×10^{-4}	9.13×10^{-4}	8.91×10^{-4}
50	NeMDO$_{p=2}$	MAE	4.43×10^{-4}	3.95×10^{-4}	4.00×10^{-4}	4.15×10^{-4}	4.14×10^{-4}
		St.d.	5.68×10^{-4}	5.18×10^{-4}	5.25×10^{-4}	5.76×10^{-4}	5.60×10^{-4}
100	NeMDO$_{p=2}$	MAE	1.25×10^{-3}	1.24×10^{-3}	1.63×10^{-3}	1.79×10^{-3}	1.59×10^{-3}
		St.d.	1.61×10^{-3}	1.56×10^{-3}	2.15×10^{-3}	2.29×10^{-3}	2.03×10^{-3}



(a) $|\mathcal{N}_i| = 10$ (b) $|\mathcal{N}_i| = 25$ (c) $|\mathcal{N}_i| = 50$ (d) $|\mathcal{N}_i| = 100$

Fig. 14. Parametric results for varying stencil size n for node disorder of $\epsilon = 1.0$. Colours indicate value of weights. To generate the plots, multiple stencils with normalised relative positions have their weights predicted and plotted on top of each other.

dominant limiting factors. Consistent with the convergence results in Section 5.2, larger residuals (reported in Table 1) translate into lower accuracy. Consequently, models exhibiting larger moment residuals demonstrate a higher error floor.

Fig. 14 shows the plot of multiple stencil configurations with the particles coloured by weight on top of each other. Stencils with a smaller neighbour count exhibit high-frequency spatial fluctuations. The anti-symmetric character is present at small $|\mathcal{N}_i|$, but is overlaid by substantial scatter in the individual weights. As the neighbour count increases, this scatter reduces and the anti-symmetric structure becomes more visible. This noisy distribution implies that the local consistency conditions are satisfied through highly non-symmetric weight assignments. As neighbour count increases, the stencils undergo a clear structural refinement, transitioning from a noisy distribution of weights towards a clearer anti-symmetric stencil structure. Furthermore, the magnitude of the weights decreases with increasing neighbour count, reflecting a transition from an ill-conditioned local approximation to a more regularised and distributed representation.

Fig. 15 illustrates the convergence behaviour of the NeMDO's x-derivative operator with multiple neighbourhood size $|\mathcal{N}_i|$. At lower resolutions ($s = 10^{-1}$), stencils with a smaller number of neighbours yield superior L_2 error norms; however, as the resolution

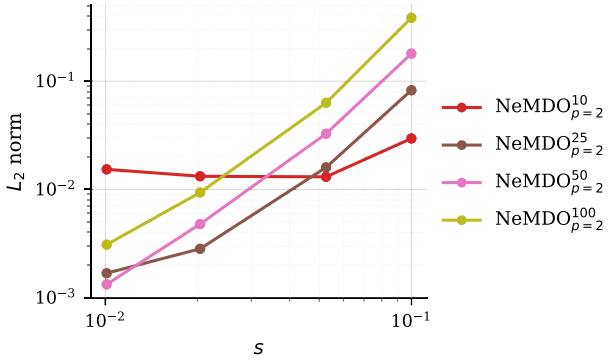


Fig. 15. Convergence of the discrete x-derivative operator on a smooth test function with $\epsilon = 1.0$. Relative L_2 norm versus particle spacing s for the learned NeMDO operator with varying stencil size. The superscript in the operator indicates the number of particles within the computational stencil.

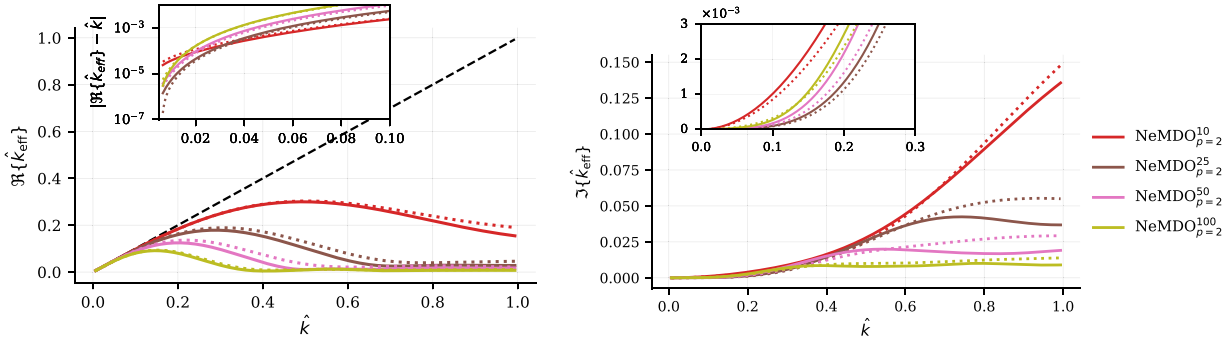


Fig. 16. Resolving power of the gradient operator for varying number of particles in computational stencil. The *left* plot shows the real part of the modal response, and the *right* plot shows the imaginary part on disordered particle distributions with noise level $\epsilon = 1.0$. The horizontal axis shows the true wavenumber k normalised by the Nyquist wavenumber k_{Ny}, while the vertical axis shows the corresponding effective wavenumber $k_{\text{eff}}/k_{\text{Ny}}$. The dashed black line indicates the ideal response, corresponding to a spectral method. The inset on the *left* plot shows the absolute difference between the different operators and the ideal spectral response. The inset on the *right* plot shows zoomed in graph of $\Im\{\hat{k}_{\text{eff}}\}$ for range $0 \leq \hat{k} \leq 0.3$. The full and dotted lines indicate $k_y/k_x = 0$ and $k_y/k_x = 1$, respectively.

increases, larger stencils demonstrate a more favourable asymptotic error floor. This observed stagnation in accuracy is consistent with the moment analysis presented in Table 2. For example, the first-order moment sum for NeMDO$^{10}_{p=2}$ is approximately 1.3×10^{-2}, which directly corresponds to the resolution at which the operator reaches its formal truncation error limit.

Fig. 16 illustrates the resolving power of NeMDO across varying neighbourhood sizes. At higher wavenumbers, reducing the stencil size significantly improves the real component of the modal response. This spectral behaviour directly correlates with the lower L_2 error norms observed at coarser resolutions in Fig. 15. However, at low wavenumbers (Fig. 16, *left* inset), the NeMDO$^{10}_{p=2}$ operator exhibits a larger absolute error compared to configurations with larger supports. This deterioration in the low-frequency modal response is a consequence of larger moment residuals, which impose a stricter truncation error limit. Conversely, increasing the stencil size leads to a reduction in dissipation error. These results suggest a fundamental trade-off between improved modal response for high-wavenumber under-constrained stencils and the improved consistency afforded by larger neighbourhood sizes.

6.2. Taylor truncation

We now vary the Taylor truncation by increasing the number of moments included in $\mathbf{M}^D$. In traditional high-order mesh-free numerical methods, this enforces polynomial consistency to a higher degree, which leads to more accurate solutions at equivalent resolution (i.e. improved resolving power). However, the local systems that must be solved become more poorly conditioned for higher orders of approximation; thus, these usually require larger computational stencils, and are more prone to instabilities [17,56].

Fig. 17 illustrates the stencil structure for the NeMDO x-derivative operator trained with third-order Taylor monomials and $|\mathcal{N}_i| = 50$. Consistent with the second-order approximation (Fig. 14c, also $|\mathcal{N}_i| = 50$), the third-order operator maintains a defined anti-symmetric structure about the origin. However, the higher-order consistency requirement leads to increased weight magnitudes compared to the second-order counterpart. Furthermore, these larger weights exhibit a higher spatial concentration closer to the origin. This suggests that as the polynomial order increases, the operator prioritises the nearest neighbours to satisfy the more stringent higher-order moments.

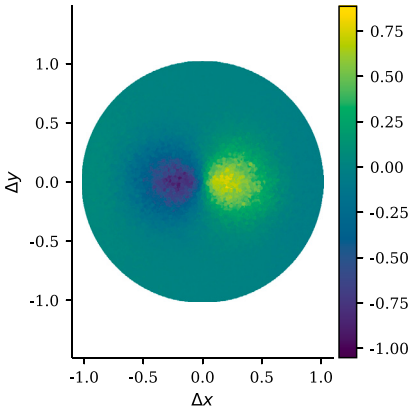


Fig. 17. Normalised third-order NeMDO for the $x-$derivative, colours indicate weight value. Model was trained with $|\mathcal{N}_i| = 50$ and particle disturbance of $\epsilon = 1.0$ (and inferred with the same noise).

Table 3
NeMDO moment residuals for x-derivative operators with Taylor monomials up to third order, $\text{NeMDO}^{x}_{p=3}$. Mean absolute error (MAE) and standard deviation are reported for each monomial moment, averaged over independently perturbed neighbourhood realisations. Residuals quantify the extent to which the learned discrete operators satisfy the imposed Taylor consistency constraints (smaller values are better).

Metric	x	y	$x^2/2$	xy	$y^2/2$	$x^3/6$	$x^2y/2$	$xy^2/2$	$y^3/6$
MAE	2.08×10^{-3}	2.07×10^{-3}	1.71×10^{-3}	1.92×10^{-3}	1.85×10^{-3}	2.42×10^{-3}	2.43×10^{-3}	2.39×10^{-3}	2.86×10^{-3}
St.d.	2.70×10^{-3}	2.73×10^{-3}	2.24×10^{-3}	2.52×10^{-3}	2.40×10^{-3}	3.29×10^{-3}	3.21×10^{-3}	3.21×10^{-3}	3.61×10^{-3}

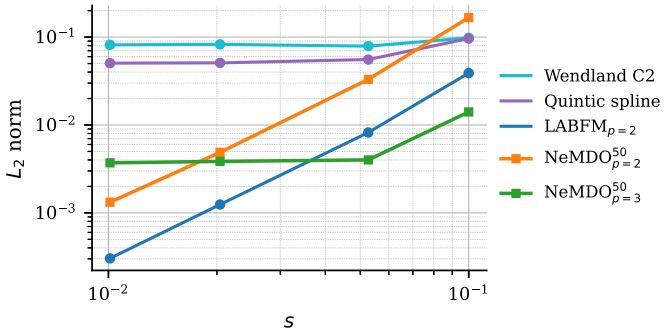


Fig. 18. Convergence of the discrete x-derivative operator on a smooth test function with $\epsilon = 0.5$. Relative L_2 norm versus particle spacing s for the learned NeMDO operator with varying Taylor truncation. The superscript in the NeMDO operators indicate the stencil size $|\mathcal{N}_i|$.

Table 3 shows NeMDO average moments up to third-order Taylor monomials averaged over multiple stencil configurations unseen during training. All moments are satisfied with similar accuracy approximately at order 10^{-3}. Compared to the second-order approximation with the same stencil size (in Table 2), the moments are about one order of magnitude larger. By increasing the terms present in the Taylor truncation, the loss function becomes more complex, requiring the optimiser to balance a larger set of competing consistency conditions within the same local support. This increased requirement for higher-order polynomial accuracy causes degradation in moment residuals.

Fig. 18 illustrates the convergence behaviour of the third-order NeMDO x-derivative operator. At lower resolutions, $\text{NeMDO}^{50}_{p=3}$ outperforms all other operators, demonstrating the benefits of increasing the Taylor truncation. However, as the resolution increases, the model trained with more stencil moments quickly reaches its limiting error, indicated by the sum of the first-order moments in Table 3.

Fig. 19 illustrates the resolving power as a function of the Taylor truncation order. Consistent with classical high-order numerical schemes, extending the truncated expansion improves the spectral fidelity of the real component of the modal response. This enhancement is further evidenced by the convergence results in Fig. 18, where the $p = 3$ model significantly outperforms the lower-order operators at coarse resolutions. However, for wavenumbers $\hat{k} \geq 0.5$, the real component of the $\text{NeMDO}_{p=3}$ response begins to attenuate, falling below the performance of both $\text{LABFM}_{p=2}$ and the Wendland C2 kernel. At low wavenumbers ($\hat{k} \leq 0.04$), the inset of Fig. 19 (*left*) reveals a distinct divergence in error scaling. For $\text{NeMDO}_{p=2}$ and $\text{LABFM}_{p=2}$, the error magnitude exhibits a steep positive gradient for $\hat{k} \in [0, 0.04]$, which subsequently transitions to a lower growth rate as $\hat{k}$ increases. This initial steepness

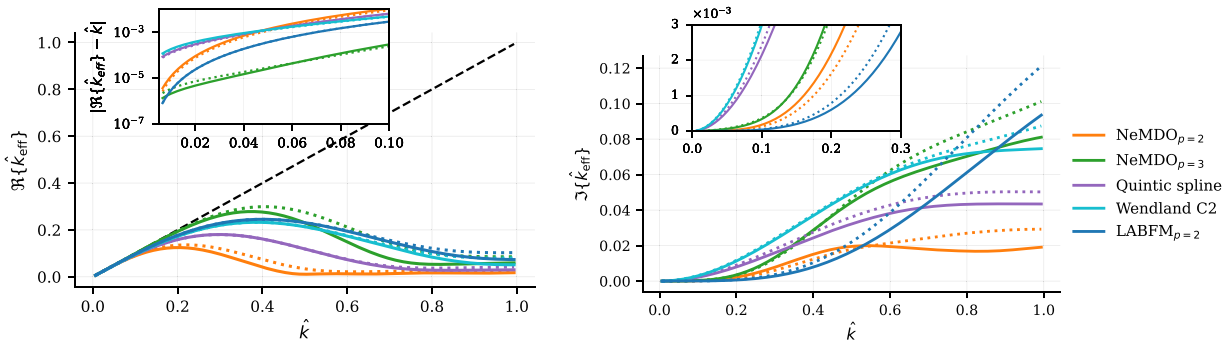


Fig. 19. Resolving power of the gradient operator for increasing Taylor truncation included during training. The *left* plot shows the real part of the modal response, and the *right* plot shows the imaginary part on disordered particle distributions with noise level $\epsilon = 1.0$. The horizontal axis shows the true wavenumber k normalised by the Nyquist wavenumber k_{Ny}, while the vertical axis shows the corresponding effective wavenumber $k_{\text{eff}}/k_{\text{Ny}}$. The dashed black line indicates the ideal response, corresponding to a spectral method. The inset on the *left* plot shows the absolute difference between the different operators and the ideal spectral response. The inset on the *right* plot shows zoomed in graph of $\Im\{\hat{k}_{\text{eff}}\}$ for range $0 \leq \hat{k} \leq 0.3$. The full and dotted lines indicate $k_y/k_x = 0$ and $k_y/k_x = 1$, respectively.

reflects the operators' ability to resolve increasingly smooth modes with high accuracy. In contrast, the error profile for NeMDO$_{p=3}$ remains relatively invariant across this low-frequency regime. This plateau indicates that the $p = 3$ operator has reached a truncation error floor, where the error is no longer dominated by the wavenumber but by the constant moment residuals reported in Table 3. A similar spectral stagnation is observed for the SPH kernels (albeit at a much larger absolute error), which exhibit a horizontal error profile as $\hat{k} \to 0$, indicating a lack of formal consistency at lower wavenumbers. The imaginary part of the modal response is shown in Fig. 19 (*right*). Including additional terms of the Taylor truncation degrades stability. Interestingly, NeMDO$_{p=3}$ exhibits a profile similar to Wendland C2 across the entire wavenumber spectrum, with marginally better stability up to $\hat{k} \approx 0.85$.

6.3. Particle disorder

This section investigates the sensitivity of the NeMDO operators to varying degrees of geometric disorder, ϵ, during both the training and inference phases. To evaluate the generalisation capabilities of the learned weights, we trained separate NeMDO models across a spectrum of particle perturbations, specifically $\epsilon \in \{0.1, 0.5, 1.0\}$. We then performed a cross-evaluation to determine the impact of training-set stochasticity on the resulting operator consistency. Table 4 summarises the moment residuals for each configuration, providing a quantitative measure of how well each model maintains polynomial consistency when subjected to different noise regimes.

The cross-consistency analysis in Table 4 demonstrates a clear trade-off between peak specialisation and operational robustness. The operator trained on minimal disorder ($\epsilon = 0.1$) achieves the lowest total MAE for the low-order moments (2.04×10^{-4}) when evaluated in its native noise. However, this model is highly specialised; when subjected to higher disorders ($\epsilon \in \{0.5, 1.0\}$), this model encounters out-of-distribution geometric configurations that were not represented during training, leading up to a two-order-of-magnitude increase in residuals.

In contrast, training on the highest noise level ($\epsilon = 1.0$) provides a broader coverage of stencil arrangements, including the near-regular states found at lower noise levels, the resulting model functions as a robust generalist. While this breadth prevents the operator from reaching the same high accuracy on a more structured point cloud, it ensures that the moments remain well-satisfied for a wide range of particle distribution. For Lagrangian simulations, where particle distributions inevitably transition from ordered to highly disordered states, the distributional robustness afforded by high-noise training is a critical prerequisite for maintaining numerical consistency throughout the simulation.

Fig. 20 illustrates the L_2 error norm for the x-derivative operator across varying spatial resolutions and degrees of particle disorder. Across all noise regimes, both SPH kernels exhibit a significantly higher truncation error floor compared to the NeMDO and LABFM frameworks. For the majority of operators, the asymptotic error limit is strongly correlated with the magnitude of particle disorder, with reduced stochasticity leading to superior accuracy.

An exception is observed in the LABFM results, which maintain a relatively invariant profile regardless of the disorder level. Given that LABFM explicitly enforces consistency by solving local linear systems at each evaluation point, it remains numerically robust even in highly disordered configurations where a valid solution to the moment equations exists. This behaviour aligns with the extensive analysis of particle disorder and approximation order conducted by King et al. [17].

The NeMDO results further substantiate the distributional coverage observations from Table 4. Operators trained under high perturbations ($\epsilon = 1.0$) demonstrate a convergence plateau that remains largely independent of the inference noise level, provided the latter does not exceed the training threshold. Conversely, models trained on low-disorder distributions exhibit a significant increase in the limiting error when applied to particle configurations that lie outside their training manifold. While LABFM achieves robustness by solving a linear system for each local neighbourhood, NeMDO attains a similar robustness by training on a wide ensemble of disordered configurations ($\epsilon = 1.0$).

Table 4
NeMDO moment residuals for x-derivative operators with varying particle disorder ϵ. Mean absolute error (MAE) and standard deviation are reported for each monomial moment, averaged over independently perturbed neighbourhood realisations. Residuals quantify the extent to which the learned discrete operators satisfy the imposed Taylor consistency constraints (smaller values are better). Trained ϵ denotes to the particle disorder during training, and inferred ϵ denotes the particle disorder during inference.

Trained ϵ	Operator	Inferred ϵ	Metric	x	y	$x^2/2$	xy	$y^2/2$
1.0	NeMDO$_{p=2}$	1.0	MAE	7.88×10^{-4}	7.73×10^{-4}	8.51×10^{-4}	8.86×10^{-4}	8.91×10^{-4}
			St.d.	1.07×10^{-3}	9.89×10^{-4}	1.21×10^{-3}	1.24×10^{-3}	1.22×10^{-3}
		0.5	MAE	4.38×10^{-4}	4.88×10^{-4}	4.23×10^{-4}	3.93×10^{-4}	4.81×10^{-4}
			St.d.	4.74×10^{-4}	4.78×10^{-4}	5.26×10^{-4}	5.04×10^{-4}	5.79×10^{-4}
		0.1	MAE	3.60×10^{-4}	5.83×10^{-4}	2.57×10^{-4}	3.56×10^{-4}	3.48×10^{-4}
			St.d.	1.03×10^{-4}	9.63×10^{-5}	1.76×10^{-4}	1.43×10^{-4}	1.66×10^{-4}
0.5	NeMDO$_{p=2}$	1.0	MAE	2.35×10^{-3}	1.73×10^{-3}	1.64×10^{-3}	1.75×10^{-3}	1.52×10^{-3}
			St.d.	3.52×10^{-3}	2.47×10^{-3}	2.50×10^{-3}	2.67×10^{-3}	2.26×10^{-3}
		0.5	MAE	4.74×10^{-4}	4.41×10^{-4}	4.63×10^{-4}	5.13×10^{-4}	4.83×10^{-4}
			St.d.	6.21×10^{-4}	5.88×10^{-4}	6.15×10^{-4}	6.85×10^{-4}	6.39×10^{-4}
		0.1	MAE	2.13×10^{-4}	5.85×10^{-5}	9.95×10^{-5}	1.08×10^{-4}	3.42×10^{-4}
			St.d.	1.18×10^{-4}	6.95×10^{-5}	8.67×10^{-5}	1.22×10^{-4}	9.58×10^{-5}
0.1	NeMDO$_{p=2}$	1.0	MAE	1.60×10^{-2}	1.34×10^{-2}	1.32×10^{-2}	1.39×10^{-2}	1.21×10^{-2}
			St.d.	2.09×10^{-2}	1.92×10^{-2}	1.89×10^{-2}	1.73×10^{-2}	1.72×10^{-2}
		0.5	MAE	6.11×10^{-3}	5.34×10^{-3}	6.58×10^{-3}	6.70×10^{-3}	5.74×10^{-3}
			St.d.	8.13×10^{-3}	7.42×10^{-3}	9.46×10^{-3}	8.45×10^{-3}	8.23×10^{-3}
		0.1	MAE	1.09×10^{-4}	9.50×10^{-5}	1.33×10^{-4}	1.38×10^{-4}	1.43×10^{-4}
			St.d.	1.69×10^{-4}	1.37×10^{-4}	1.97×10^{-4}	2.14×10^{-4}	1.99×10^{-4}
–	Wendland C2	1.0	MAE	1.01×10^{-1}	7.08×10^{-2}	4.80×10^{-2}	4.93×10^{-2}	2.55×10^{-2}
			St.d.	1.21×10^{-1}	8.84×10^{-2}	5.90×10^{-2}	6.16×10^{-2}	3.18×10^{-2}
		0.5	MAE	4.94×10^{-2}	4.02×10^{-2}	2.44×10^{-2}	2.67×10^{-2}	1.28×10^{-2}
			St.d.	6.01×10^{-2}	5.03×10^{-2}	3.09×10^{-2}	3.37×10^{-2}	1.57×10^{-2}
		0.1	MAE	1.03×10^{-2}	8.34×10^{-3}	5.73×10^{-3}	5.41×10^{-3}	2.78×10^{-3}
			St.d.	1.23×10^{-2}	1.05×10^{-2}	7.13×10^{-3}	6.77×10^{-3}	3.48×10^{-3}
–	Quintic Spline	1.0	MAE	6.59×10^{-2}	4.57×10^{-2}	3.77×10^{-2}	3.98×10^{-2}	2.12×10^{-2}
			St.d.	8.04×10^{-2}	5.75×10^{-2}	4.69×10^{-2}	4.95×10^{-2}	2.64×10^{-2}
		0.5	MAE	3.44×10^{-2}	2.42×10^{-2}	1.98×10^{-2}	2.09×10^{-2}	1.10×10^{-2}
			St.d.	4.24×10^{-2}	3.03×10^{-2}	2.46×10^{-2}	2.59×10^{-2}	1.38×10^{-2}
		0.1	MAE	7.01×10^{-3}	4.98×10^{-3}	4.05×10^{-3}	4.26×10^{-3}	2.23×10^{-3}
			St.d.	8.63×10^{-3}	6.19×10^{-3}	5.04×10^{-3}	5.27×10^{-3}	2.80×10^{-3}

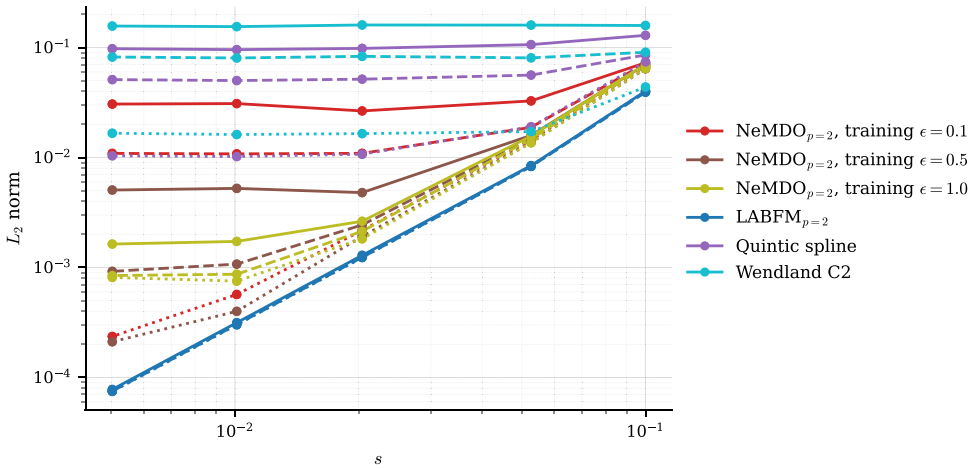


Fig. 20. Convergence of the discrete x-derivative operator on a smooth test function (Eq. (27)) for LABFM$_{p=2}$, SPH (with the quintic spline and Wendland C2 kernels) and NeMDO, with varying particle disorder ϵ. Full lines indicate inference with ϵ = 1.0, dashed lines indicate inference with ϵ = 0.5, and dotted lines indicate inference with ϵ = 0.1.

Table 5
Moment residuals for the learned x-derivative and Laplacian operators applied to a single fixed stencil rotated through eight orientations. Residuals are reported for each monomial moment, with the standard deviation across orientations given in the final row of each block. The rightmost column reports the maximum residual across all monomials at each orientation. Small variation across orientations indicates that the accuracy of the learned operators is largely insensitive to the orientation of the stencil.

Operator	Rotation	x	y	$x^2/2$	xy	$y^2/2$	$\|e\|_\infty$
$\text{NeMDO}^{x}_{p=2}$	0°	7.49×10^{-5}	9.33×10^{-5}	1.92×10^{-4}	8.86×10^{-5}	4.83×10^{-5}	1.92×10^{-4}
	45°	2.10×10^{-5}	1.38×10^{-4}	8.45×10^{-5}	4.74×10^{-5}	7.51×10^{-5}	1.38×10^{-4}
	90°	1.38×10^{-5}	7.94×10^{-5}	1.41×10^{-4}	8.01×10^{-5}	8.13×10^{-6}	1.41×10^{-4}
	135°	3.36×10^{-5}	7.17×10^{-5}	1.70×10^{-4}	7.92×10^{-6}	3.20×10^{-5}	1.70×10^{-4}
	180°	4.11×10^{-5}	1.49×10^{-4}	1.07×10^{-5}	7.35×10^{-5}	9.53×10^{-6}	1.49×10^{-4}
	225°	5.52×10^{-5}	3.18×10^{-5}	9.62×10^{-5}	2.14×10^{-5}	4.00×10^{-5}	9.62×10^{-5}
	270°	7.16×10^{-5}	4.23×10^{-5}	8.12×10^{-5}	8.27×10^{-5}	3.12×10^{-5}	8.27×10^{-5}
	315°	4.65×10^{-6}	7.39×10^{-5}	6.75×10^{-5}	9.36×10^{-5}	8.99×10^{-5}	9.36×10^{-5}
	St.d.	4.58×10^{-5}	3.86×10^{-5}	5.53×10^{-5}	5.87×10^{-5}	4.36×10^{-5}	–
$\text{NeMDO}^{\Delta}_{p=2}$	0°	2.53×10^{-4}	6.32×10^{-4}	9.89×10^{-4}	6.95×10^{-4}	5.23×10^{-4}	9.89×10^{-4}
	45°	5.49×10^{-4}	1.77×10^{-3}	8.01×10^{-4}	7.04×10^{-4}	7.35×10^{-4}	1.77×10^{-3}
	90°	5.32×10^{-4}	8.43×10^{-4}	6.15×10^{-4}	4.05×10^{-4}	7.52×10^{-4}	8.43×10^{-4}
	135°	5.02×10^{-4}	8.60×10^{-4}	8.57×10^{-4}	5.49×10^{-4}	4.43×10^{-4}	8.60×10^{-4}
	180°	3.88×10^{-5}	9.82×10^{-4}	3.39×10^{-4}	4.48×10^{-4}	7.71×10^{-4}	9.82×10^{-4}
	225°	7.74×10^{-5}	8.69×10^{-4}	8.37×10^{-4}	1.12×10^{-3}	1.00×10^{-3}	1.12×10^{-3}
	270°	4.71×10^{-4}	1.21×10^{-3}	8.55×10^{-4}	9.75×10^{-4}	4.12×10^{-5}	1.21×10^{-3}
	315°	7.27×10^{-5}	1.14×10^{-3}	8.79×10^{-4}	9.36×10^{-4}	5.20×10^{-4}	1.14×10^{-3}
	St.d.	3.43×10^{-4}	3.25×10^{-4}	1.90×10^{-4}	2.44×10^{-4}	2.70×10^{-4}	–

6.4. Rotational robustness

This section examines the robustness of the learned operators to rigid rotation of the stencil. A single stencil generated with $\epsilon = 1.0$ is rotated through eight orientations at 45° intervals, and at each orientation the moment residuals are evaluated against the correspondingly transformed target vector. No rotational symmetry is imposed by the architecture, so consistency across orientations is a property the network has learned rather than one guaranteed by construction.

Table 5 reports the results. For the x-derivative, the maximum residual across monomials varies between 8.27×10^{-5} and 1.92×10^{-4}, a factor of 2.3 across the full range of orientations, with a standard deviation per monomial of the same order as the residuals themselves. For the Laplacian, the maximum residual varies between 8.43×10^{-4} and 1.77×10^{-3}, a factor of 2.1. No orientation is degenerate, and the residuals at 45° offset from the axes are comparable to those aligned with them, indicating that the framework is robust to rotational variance.

7. Limitations

In its present form, each model is trained for a fixed stencil size $|\mathcal{N}_i|$ and does not natively support variable-size neighbourhoods at inference time. Currently, this lack of topological flexibility necessitates the training of discrete operators for every desired neighbour count. Allowing variable-size neighbourhoods would provide greater flexibility and simplify integration with varying particle distributions.

Another limitation of the current framework is that the achievable accuracy degrades as particle disorder increases. This behaviour reflects the growing difficulty of satisfying polynomial moment constraints under highly irregular stencils and is not unique to NeMDO: classical SPH discretisations and consistency-corrected mesh-free methods similarly suffer from reduced accuracy and conditioning issues under strong node disorder at higher polynomial consistency enforcement. These effects become especially relevant in Lagrangian settings, where particle densities and neighbourhood sizes can vary significantly over time. In such cases, the framework would likely need to be coupled with particle-shifting or regularisation strategies [64–66], as is standard practice in SPH-based methods, to maintain suitable stencil quality. Extending the architecture to more robustly handle adaptive stencils and highly disordered configurations is an important direction for future work.

8. Conclusion & future work

This work presented NeMDO, a self-supervised learning framework for the construction of mesh-free discrete differential operators on unstructured particle sets. By training graph neural networks to satisfy polynomial moment constraints derived from truncated Taylor expansions, we have developed a methodology for generating operators that are physics-agnostic, strictly local, and robust to significant geometric disorder. Through multiple benchmarks — including moment and stability analysis, convergence studies, modal response, Poisson's equation and Taylor–Green vortex simulations — we demonstrated that NeMDO operators achieve superior accuracy compared to standard SPH discretisations. The framework has a capacity-dependent error floor: accuracy cannot be refined indefinitely at fixed model capacity, and lower limiting error requires larger models at higher inference cost. Within that

Table 6
Nearest-neighbour distance statistics (normalised by smoothing length h) for each point cloud distribution. μ: mean, σ: standard deviation, CV: coefficient of variation (σ/μ) (with $h = 1.5$ s).

Distribution	μ/h	σ/h	CV
Cartesian ($\epsilon = 0$)	0.667	∼0	∼0
Perturbed ($\epsilon = 0.1$)	0.638	0.018	0.028
Perturbed ($\epsilon = 0.5$)	0.537	0.088	0.160
Perturbed ($\epsilon = 1.0$)	0.428	0.151	0.352
Iterative particle shifting	0.649	0.029	0.045

floor, a trained model applies across resolutions without retraining, and model capacity provides an additional design axis absent from traditional discretisations, whose cost to obtain weights is fixed by the algorithm.

While the current study focuses primarily on the fundamental properties of the discrete operators and their weight-generation consistency, rather than the implementation of complex boundary conditions or adaptive refinement strategies, it provides the necessary mathematical foundation for more sophisticated mesh-free applications. More broadly, this research suggests that learning spatial discrete operators provides a principled pathway for integrating machine learning into hybrid mesh-free PDE solvers.

The present implementation utilises a message-passing GNN architecture and an MLP baseline, however, NeMDO is compatible with multiple permutation-invariant architectures, including Transformers [81], DeepSets [82], and Equivariant Neural Networks [83]. In particular, rotational invariance is not explicitly enforced in the current architecture but is achieved empirically through exposure to stencils with diverse orientations during training. Equivariant graph neural networks, such as E(n) Equivariant GNN [84], could enforce this property by construction and represent a promising direction for future development. Consequently, future research will benchmark alternative architectures to optimise the balance between expressive power and inference speed. Similarly, the stencils used throughout this work are constructed using nearest-neighbour selection, the simplest available approach, adopted to ensure a fair comparison across all methods. NeMDO is agnostic to stencil construction, where the learned mapping infers operator weights from whatever local geometry it is given, and more sophisticated strategies, including support selection for complex domains [85], geometry-guided collocation from CAD representations [69], are fully compatible and will be explored in future work. Other directions for future work include: (i) applying NeMDO to three dimensions, complex geometries and Lagrangian flows, where the robustness of the learned operators to extreme topological deformation can be fully leveraged; (ii) investigate training objectives that incorporate spectral or modal targets directly into the loss function to further refine high-wavenumber resolving power; (iii) integrate symbolic regression for discovering compact kernel functions that retain the robustness of learned operators while offering the interpretability of classical mesh-free methods; (iv) incorporate uncertainty quantification into the learned operators, enabling confidence estimates [86,87] that allow a hybrid solver to fall back adaptively to classical collocation reconstruction when the local geometry lies outside the training manifold.

CRediT authorship contribution statement

Lucas Gerken Starepravo: Writing – review & editing, Writing – original draft, Visualization, Validation, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Georgios Fourtakas:** Supervision, Conceptualization. **Steven Lind:** Writing – review & editing, Supervision, Conceptualization. **Ajay B. Harish:** Supervision, Conceptualization. **Tianning Tang:** Investigation, Conceptualization. **Jack R.C. King:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

Jack R. C. King is funded by the Royal Society via a University Research Fellowship (URF\R1\221290). We would like to acknowledge Dr. Henry Broadley for his insightful discussions, and for assistance given by Research IT and the use of the Computational Shared Facility at the University of Manchester.

Appendix A. Point cloud statistics

Table 6 reports nearest-neighbour distance statistics for the point cloud distributions used during training and inference throughout the manuscript. Specifically, the mean, standard deviation, and coefficient of variation of the nearest-neighbour distance, normalised by the smoothing length h to ensure resolution invariance, are shown for different particle distributions.

To further characterise the structural differences between distributions, we compute the pair correlation function $g(r)$, which measures the probability of finding a particle at distance r relative to a spatially uniform (Poisson) distribution. Pairwise distances

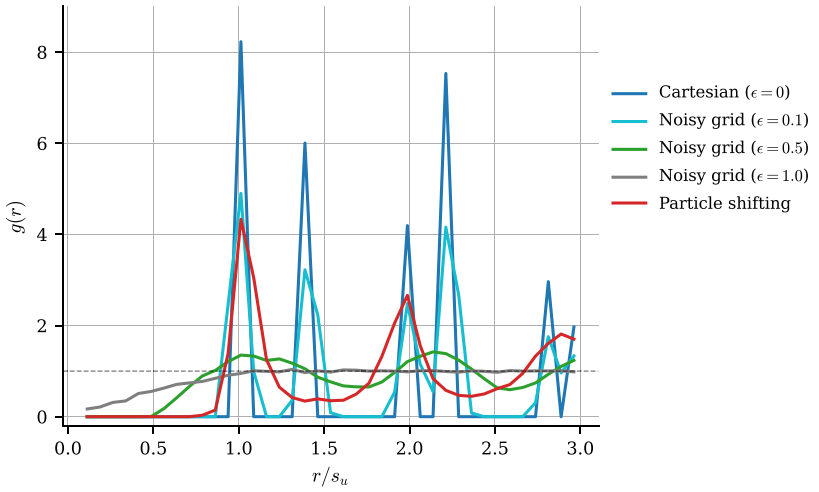


Fig. 21. Pair correlation function $g(r)$ for each point cloud distribution with radial bins $\Delta r = 0.05$. The dashed line indicates $g(r) = 1$ (uniform random reference).

from each interior particle to all neighbours within a cutoff $r_{\max} = 2h$, where $h = 1.5$ s, are histogrammed into radial bins of width Δr and normalised by the number of query particles, the annular area $2\pi r \Delta r$, and the density $\rho = N/|\Omega|$. A value of $g(r) = 1$ corresponds to a completely random arrangement; peaks above unity indicate preferred inter-particle spacings, and below the opposite.

$$g(r_k) = \frac{1}{N} \sum_{i=1}^{N} \frac{n_i(r_k)}{\rho \, 2\pi r_k \, \Delta r}, \tag{A.1}$$

where N is the number of particles, r_k denotes the centre of the kth radial bin, and $n_i(r_k)$ is the number of neighbours of particle i in the annular bin centred at r_k with width Δr.

Fig. 21 shows $g(r)$ for all distributions listed in Table 6. The Cartesian grid ($\epsilon = 0$) exhibits sharp peaks at the lattice distances $r/s_u = 1, \sqrt{2}, 2$. As the perturbation magnitude increases, these peaks broaden and diminish: at $\epsilon = 0.5$ the lattice signature persists as oscillations about $g(r) = 1$, whereas at $\epsilon = 1.0$ the curve is essentially flat, indicating that all long-range lattice memory has been destroyed. The particle-shifted distribution displays a well-defined exclusion zone at small r, a sharp first-neighbour peak, and rapid decay towards $g(r) = 0.5$ with second-shell structure. This highlights the challenging scenario which we validated NeMDO, and why we expect it to do well in more realistic point-cloud arrangements [69,85].

Appendix B. Model hyper-parameters and training dataset

Below we summarise the NeMDO architectures used in the different experimental sections.

B.0.0.1. Main experiments. For the polynomial consistency and convergence (Section 5.2), stability (Section 5.3), modal response (Section 5.4), Poisson's equation and Taylor–Green vortex simulations (Section 5.6), we use a single architecture for both the x-derivative and Laplacian operators, denoted $\text{NeMDO}^{x}_{p=2}$ and $\text{NeMDO}^{4}_{p=2}$. Their hyper-parameters are listed in Table 7. In the computational cost-accuracy plot (Fig. 7), this configuration is referred to as $\text{NeMDO}^{3}_{p=2}$. The models presented in Table 7 were trained with approximately 7 million neighbourhood graphs, with about 2 million samples used for validation and 1 million reserved for testing, yielding a total of roughly 10 million neighbourhoods. The models were trained with a starting learning rate of 1×10^{-5} and a plateau scheduler for a total of 2000 epochs. Training each of these models required approximately 60 h on a single NVIDIA A100 GPU. The dataset size was chosen conservatively to remove data volume as a limiting factor; no attempt was made to identify the minimum sufficient dataset size, and comparable performance is likely attainable at reduced cost.

B.0.0.2. Parametric study. During the parametric study on Taylor truncation and stencil size (Section 6), we fix the network architecture and vary only the polynomial consistency $p \in \{2, 3\}$ and the number of neighbours $|\mathcal{N}_i| \in \{10, 25, 50, 100\}$. All models in this group share the same parameter count and embedding dimension. The models have 183.7k trainable parameters, all MLPs (encoder, decoder, message and update) only have 1 hidden layer, the GNN has 2 graph layers, and all models with an embedding of size 128, with particle disorder of $\epsilon = 1.0$, trained for 1000 epochs, with a starting learning rate of 1×10^{-4} with a plateau scheduler. The models presented in the parametric investigation were trained with approximately 115k neighbourhood graphs, with about 33k samples used for validation and 16k reserved for testing, yielding a total of roughly 164k neighbourhoods.

Table 7
NeMDO architectures used in the main experiments.

Operator	$\lvert\mathcal{N}_i\rvert$	# Parameters	F_h	MLP hidden layers (Emb/Out/Msg/Upd)	Graph layers	ϵ
$\text{NeMDO}^{x}_{p=2}$	35	2.2M	256	3	3	1.0
$\text{NeMDO}^{4}_{p=2}$	35	2.2M	256	3	3	1.0
$\text{NeMDO}^{\text{hyp}}_{p=4}$	150	2.2M	256	3	3	1.0

Table 8
NeMDO architectures used in the cost–accuracy experiments.

Operator	$\lvert\mathcal{N}_i\rvert$	# Parameters	F_h	MLP hidden layers (Emb/Out/Msg/Upd)	Graph layers	ϵ
$\text{NeMDO}^{1}_{p=2}$	10	11.1k	32	1	2	1.0
$\text{NeMDO}^{2}_{p=2}$	15	46.3k	64	1	2	1.0

B.0.0.3. Cost–accuracy experiments. For the computational cost and accuracy study in Fig. 7, we use smaller-capacity models to probe the trade-off between parameter count, runtime, and limiting error. The architectures are summarised in Table 8, all models were trained for 2000 epochs. $\text{NeMDO}^{3}_{p=2}$ (Fig. 7) corresponds to $\text{NeMDO}^{x}_{p=2}$ listed in Table 7. The models presented in the computational cost-accuracy experiments (Table 8) were trained with approximately 115k neighbourhood graphs, with about 33k samples used for validation and 16k reserved for testing, yielding a total of roughly 164k neighbourhoods.

Appendix C. Computational cost details

Providing a fully balanced performance comparison between fundamentally different approaches is inherently difficult. The estimates in Fig. 7 should therefore be interpreted as indicative rather than absolute. All timings were obtained on a single AMD Ryzen 7 7800X3D core (single thread). For the SPH baselines, we measure the time required to compute weights with the kernels. For LABFM, we include the time to assemble and solve the resulting linear systems. For NeMDO, we measure the wall-clock time of inference—we do not include the training time since the learned operator can be trained offline once, and implemented on different PDEs. The time measurements provided in Fig. 7 represents the total time for each operator to predict the x-derivative weights for all nodes in a given resolution.

We do not report GPU or multi-threaded timings in this work; a systematic study of heterogeneous CPU–GPU execution and distributed inference is left for future work. LABFM dense, low-rank solves and ML libraries are well suited to GPU execution, and a batched implementation is the relevant comparison for production codes. We therefore note that the ordering reported here may change in such comparison. The inference cost of NeMDO is likewise a property of the chosen architecture rather than of the framework: architectures better matched to this problem may attain the same accuracy at lower cost, and the timings reported here should not be read as a bound on what the approach can achieve. The key takeaway from the present measurements is the scaling behaviour of the methods: NeMDO eliminates the per-stencil linear solves required by corrected-kernel schemes and exposes explicit architectural and stencil-size knobs to trade compute for accuracy within a single framework.

Data availability

The complete implementation for neural network training, along with the datasets used for training and the scripts for convergence, stability, and resolving power analyses, is available as open-source at https://github.com/uom-complexfluids/nemdo. The repository is organised into modular directories for training and testing, including all benchmarked methods used in this study.

References

[1] J.W. Thomas, Numerical Partial Differential Equations: Finite Difference Methods, Springer Science & Business Media, 2013.
[2] S.K. Lele, Compact finite difference schemes with spectral-like resolution, J. Comput. Phys. 103 (1) (1992) 16–42, http://dx.doi.org/10.1016/0021-9991(92)90324-R.
[3] R.W. Clough, Original formulation of the finite element method, Finite Elem. Anal. Des. 7 (2) (1990) 89–101, http://dx.doi.org/10.1016/0168-874X(90)90001-U.
[4] P.W. McDonald, The Computation of Transonic Flow Through Two-Dimensional Gas Turbine Cascades, vol. 79825, American Society of Mechanical Engineers, 1971.
[5] R. MacCormack, A. Paullay, Computational efficiency achieved by time splitting of finite difference operators, in: 10th Aerospace Sciences Meeting, 1972, p. 154.
[6] D.I. Gottlieb, S.A. Orszag, Numerical Analysis of Spectral Methods: Theory and Applications, SIAM, Philadelphia, PA, 1977.
[7] R. Vacondio, C. Altomare, M.D. Leffe, X. Hu, D.L. Touzé, S. Lind, J.-C. Marongiu, S. Marrone, B.D. Rogers, A. Souto-Iglesias, Grand challenges for smoothed particle hydrodynamics numerical schemes, Comput. Part. Mech. 8 (3) (2021) 575–588, http://dx.doi.org/10.1007/s40571-020-00354-1.
[8] S.J. Lind, B.D. Rogers, P.K. Stansby, Review of smoothed particle hydrodynamics: Towards converged Lagrangian flow modelling, Proc. R. Soc. A: Math. Phys. Eng. Sci. 476 (2241) (2020) 20190801, http://dx.doi.org/10.1098/rspa.2019.0801.
[9] G.R. Liu, Y.T. Gu, An Introduction to Meshfree Methods and Their Programming, first ed., Springer, Dordrecht, 2005, http://dx.doi.org/10.1007/1-4020-3468-7.

[10] B.D. Wood, X. He, S.V. Apte, Modeling turbulent flows in porous media, Annu. Rev. Fluid Mech. 52 (2020) http://dx.doi.org/10.1146/annurev-fluid-010719-060317.
[11] L.B. Lucy, A numerical approach to the testing of the fission hypothesis, Astron. J. 82 (1977) 1013–1024, http://dx.doi.org/10.1086/112164.
[12] J.J. Monaghan, Simulating free surface flows with SPH, J. Comput. Phys. 110 (2) (1994) 399–406, http://dx.doi.org/10.1006/jcph.1994.1034.
[13] N.J. Quinlan, M. Basa, M. Lastiwka, Truncation error in mesh-free particle methods, Internat. J. Numer. Methods Engrg. 66 (13) (2006) 2064–2085, http://dx.doi.org/10.1002/nme.1617.
[14] B. Fornberg, E. Lehto, Stabilization of RBF-generated finite difference methods for convective PDEs, J. Comput. Phys. 230 (6) (2011) 2270–2285, http://dx.doi.org/10.1016/j.jcp.2010.12.014.
[15] B. Schrader, S. Reboux, I.F. Sbalzarini, Discretization correction of general integral PSE operators for particle methods, J. Comput. Phys. 229 (11) (2010) 4159–4182, http://dx.doi.org/10.1016/j.jcp.2010.02.004.
[16] B.J. Gross, N. Trask, P. Kuberry, P.J. Atzberger, Meshfree methods on manifolds for hydrodynamic flows on curved surfaces: A generalized moving least-squares (GMLS) approach, J. Comput. Phys. 409 (2020) 109340, http://dx.doi.org/10.1016/j.jcp.2020.109340.
[17] J.R.C. King, S.J. Lind, A.M.A. Nasar, High order difference schemes using the local anisotropic basis function method, J. Comput. Phys. 415 (2020) 109549, http://dx.doi.org/10.1016/j.jcp.2020.109549.
[18] N. Flyer, B. Fornberg, V. Bayona, G.A. Barnett, On the role of polynomials in RBF-FD approximations: I. interpolation and accuracy, J. Comput. Phys. 321 (2016) 21–38, http://dx.doi.org/10.1016/j.jcp.2016.05.026.
[19] J.R.C. King, A mesh-free framework for high-order direct numerical simulations of combustion in complex geometries, Comput. Methods Appl. Mech. Engrg. 421 (2024) 116762, http://dx.doi.org/10.1016/j.cma.2024.116762.
[20] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, J. Comput. Phys. 378 (2019) 686–707, http://dx.doi.org/10.1016/j.jcp.2018.10.045.
[21] N. Kovachki, Z. Li, B. Liu, K. Azizzadenesheli, K. Bhattacharya, A. Stuart, A. Anandkumar, Neural operator: Learning maps between function spaces with applications to PDEs, J. Mach. Learn. Res. 24 (89) (2023) 1–97, URL: http://jmlr.org/papers/v24/21-1524.html.
[22] Y. Bar-Sinai, S. Hoyer, J. Hickey, M.P. Brenner, Learning data-driven discretizations for partial differential equations, Proc. Natl. Acad. Sci. 116 (31) (2019) 15344–15349, http://dx.doi.org/10.1073/pnas.1814058116.
[23] L.G. Starepravo, S.J. Lind, G. Fourtakas, A.B. Harish, T. Tang, J.R.C. King, Learning mesh-free discrete differential operators with self-supervised graph neural networks, in: AI&PDE: ICLR 2026 Workshop on AI and Partial Differential Equations, 2026, URL: https://openreview.net/forum?id=vPQ9dpQ2rE.
[24] W.K. Liu, S. Jun, Y.F. Zhang, Reproducing kernel particle methods, Internat. J. Numer. Methods Fluids 20 (8–9) (1995) 1081–1106, http://dx.doi.org/10.1002/fld.1650200824.
[25] N. Trask, M. Perego, P. Bochev, A high-order staggered meshless method for elliptic problems, SIAM J. Sci. Comput. 39 (2) (2017) A479–A502, http://dx.doi.org/10.1137/16M1055992.
[26] N. Trask, M. Maxey, X. Hu, A compatible high-order meshless method for the Stokes equations with applications to suspension flows, J. Comput. Phys. 355 (2018) 310–326, http://dx.doi.org/10.1016/j.jcp.2017.10.039.
[27] A.I. Tolstykh, D.A. Shirobokov, On using radial basis functions in a "finite difference mode" with applications to elasticity problems, Comput. Mech. 33 (1) (2003) 68–79, http://dx.doi.org/10.1007/s00466-003-0501-9.
[28] V. Bayona, M. Kindelan, Propagation of premixed laminar flames in 3D narrow open ducts using RBF-generated finite differences, Combust. Theory Model. 17 (5) (2013) 789–803, http://dx.doi.org/10.1080/13647830.2013.801519.
[29] V. Bayona, N. Flyer, G.M. Lucas, A.J.G. Baumgaertner, A 3-D RBF-FD solver for modeling the atmospheric global electric circuit with topography (GEC-RBFFD v1.0), Geosci. Model. Dev. 8 (10) (2015) 3007–3020, http://dx.doi.org/10.5194/gmd-8-3007-2015.
[30] D. Koschier, J. Bender, B. Solenthaler, M. Teschner, Smoothed particle hydrodynamics techniques for the physics-based simulation of fluids and solids, in: Proceedings of the Eurographics Annual Conference, 2020, http://dx.doi.org/10.2312/EGT.20191035.
[31] I.J. Schoenberg, Contributions to the problem of approximation of equidistant data by analytic functions. Part b. On the problem of osculatory interpolation. A second class of analytic approximation formulae, Quart. Appl. Math. 4 (1946) 112–141, URL: https://api.semanticscholar.org/CorpusID:125667988.
[32] H. Wendland, Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree, Adv. Comput. Math. 4 (1995) 389–396, http://dx.doi.org/10.1007/BF02123482.
[33] J.J. Monaghan, Smoothed particle hydrodynamics, Rep. Progr. Phys. 68 (8) (2005) 1703, http://dx.doi.org/10.1088/0034-4885/68/8/R01.
[34] R. Vacondio, B.D. Rogers, P.K. Stansby, P. Mignosa, J. Feldman, Variable resolution for SPH: A dynamic particle coalescing and splitting scheme, Comput. Methods Appl. Mech. Engrg. 256 (2013) 132–148, http://dx.doi.org/10.1016/j.cma.2012.12.014.
[35] M. Ferrand, D.R. Laurence, B.D. Rogers, D. Violeau, C. Kassiotis, Unified semi-analytical wall boundary conditions for inviscid, laminar or turbulent flows in the meshless SPH method, Internat. J. Numer. Methods Fluids 71 (4) (2013) 446–472, http://dx.doi.org/10.1002/fld.3666.
[36] G. Fourtakas, J.M. Dominguez, R. Vacondio, B.D. Rogers, Local uniform stencil (LUST) boundary condition for arbitrary 3-D boundaries in parallel Smoothed Particle Hydrodynamics (SPH) models, Comput. & Fluids 190 (2019) 346–361, http://dx.doi.org/10.1016/j.compfluid.2019.06.009.
[37] W. Dehnen, H. Aly, Improving convergence in Smoothed Particle Hydrodynamics simulations without pairing instability, Mon. Not. R. Astron. Soc. 425 (2) (2012) 1068–1082, http://dx.doi.org/10.1111/j.1365-2966.2012.21439.x.
[38] Z. Hao, S. Liu, Y. Zhang, C. Ying, Y. Feng, H. Su, J. Zhu, Physics-informed machine learning: A survey on problems, methods and applications, 2023, arXiv:2211.08064, URL: https://arxiv.org/abs/2211.08064.
[39] L. Lu, P. Jin, G. Pang, Z. Zhang, G.E. Karniadakis, Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators, Nat. Mach. Intell. 3 (3) (2021) 218–229, http://dx.doi.org/10.1038/s42256-021-00302-5.
[40] Z. Li, N.B. Kovachki, K. Azizzadenesheli, B. Liu, K. Bhattacharya, A. Stuart, A. Anandkumar, Fourier neural operator for parametric partial differential equations, in: Proceedings of the International Conference on Learning Representations, 2021, URL: https://openreview.net/forum?id=c8P9NQVtmnO.
[41] A. Anandkumar, K. Azizzadenesheli, K. Bhattacharya, N. Kovachki, Z. Li, B. Liu, A. Stuart, Neural operator: Graph kernel network for partial differential equations, in: ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations, 2019, URL: https://openreview.net/forum?id=fg2ZFmXFO3.
[42] S. Deshpande, S.P.A. Bordas, J. Lengiewicz, MAgNET: A graph U-Net architecture for mesh-based simulations, Eng. Appl. Artif. Intell. 133 (2024) 108055, http://dx.doi.org/10.1016/j.engappai.2024.108055.
[43] S. Deshpande, R.I. Sosa, S.P.A. Bordas, J. Lengiewicz, Convolution, aggregation and attention based deep neural networks for accelerating simulations in mechanics, Front. Mater. 10 (2023) 1128954, http://dx.doi.org/10.3389/fmats.2023.1128954.
[44] S. Subramanian, P. Harrington, K. Keutzer, W. Bhimji, D. Morozov, M. Mahoney, A. Gholami, Towards foundation models for scientific machine learning: Characterizing scaling and transfer behavior, 2023, arXiv:2306.00258, URL: https://arxiv.org/abs/2306.00258.
[45] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, P. Battaglia, Learning to simulate complex physics with graph networks, in: H.D. III, A. Singh (Eds.), Proceedings of the 37th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 119, PMLR, 2020, pp. 8459–8468.
[46] Z. Li, A.B. Farimani, Graph neural network-accelerated Lagrangian fluid simulation, Comput. Graph. 103 (2022) 201–211, http://dx.doi.org/10.1016/j.cag.2022.02.004.

[47] A.P. Toshev, G. Galletti, J. Brandstetter, S. Adami, N.A. Adams, Learning Lagrangian fluid mechanics with E(3)-equivariant graph neural networks, 2023, arXiv:2305.15603.
[48] A.P. Toshev, J.A. Erbesdobler, N.A. Adams, J. Brandstetter, Neural SPH: Improved neural modeling of Lagrangian fluid dynamics, 2024, arXiv:2402.06275.
[49] A.P. Toshev, T. Kalinov, N. Gao, S. Günnemann, N.A. Adams, On learning quasi-Lagrangian turbulence, in: ICLR 2025 Workshop on Machine Learning Multiscale Processes, 2025, URL: https://openreview.net/forum?id=R1Lrk1EffC.
[50] J. Zhuang, D. Kochkov, Y. Bar-Sinai, M.P. Brenner, S. Hoyer, Learned discretizations for passive scalar advection in a two-dimensional turbulent flow, Phys. Rev. Fluids 6 (2021) 064605, http://dx.doi.org/10.1103/PhysRevFluids.6.064605.
[51] D. Kochkov, J.A. Smith, A. Alieva, Q. Wang, M.P. Brenner, S. Hoyer, Machine learning–accelerated computational fluid dynamics, Proc. Natl. Acad. Sci. 118 (21) (2021) e2101784118, http://dx.doi.org/10.1073/pnas.2101784118.
[52] G. de Romémont, F. Renac, J. Nunez, F. Chinesta, A data-driven learned discretization approach in finite volume schemes for hyperbolic conservation laws and varying boundary conditions, Comput. & Fluids 307 (2026) 106978, http://dx.doi.org/10.1016/j.compfluid.2026.106978.
[53] M. Liu-Schiaffini, J. Berner, B. Bonev, T. Kurth, K. Azizzadenesheli, A. Anandkumar, Neural operators with localized integral and differential kernels, 2024, arXiv:2402.16845.
[54] Y. Choi, S.W. Cheung, Y. Kim, P.-H. Tsai, A.N. Diaz, I. Zanardi, S.W. Chung, D.M. Copeland, C. Kendrick, W. Anderson, T. Iliescu, M. Heinkenschloss, Defining foundation models for computational science: A call for clarity and rigor, 2025, arXiv:2505.22904.
[55] Z. Zheng, X. Li, Theoretical analysis of the generalized finite difference method, Comput. Math. Appl. 120 (2022) 1–14, http://dx.doi.org/10.1016/j.camwa.2022.06.017.
[56] V. Shankar, G.B. Wright, R.M. Kirby, A.L. Fogelson, A radial basis function (RBF)-finite difference (FD) method for diffusion and reaction-diffusion equations on surfaces, 2014, arXiv:1404.0812.
[57] B. Fornberg, N. Flyer, Solving PDEs with radial basis functions, Acta Numer. 24 (2015) 215–258, http://dx.doi.org/10.1017/S0962492914000130.
[58] S. Adami, X.Y. Hu, N.A. Adams, A transport-velocity formulation for smoothed particle hydrodynamics, J. Comput. Phys. 241 (2013) 292–307, http://dx.doi.org/10.1016/j.jcp.2013.01.043.
[59] P.N. Sun, A. Colagrossi, S. Marrone, M. Antuono, A.M. Zhang, Multi-resolution delta-plus-SPH with tensile instability control: Towards high Reynolds number flows, Comput. Phys. Comm. 224 (2018) 63–80, http://dx.doi.org/10.1016/j.cpc.2017.11.016.
[60] J.P. Morris, P.J. Fox, Y. Zhu, Modeling low Reynolds number incompressible flows using SPH, J. Comput. Phys. 136 (1) (1997) 214–226, http://dx.doi.org/10.1006/jcph.1997.5776.
[61] M.B. Liu, G.R. Liu, Smoothed Particle Hydrodynamics (SPH): An overview and recent developments, Arch. Comput. Methods Eng. 17 (1) (2010) 25–76, http://dx.doi.org/10.1007/s11831-010-9040-7.
[62] J.R.C. King, S.J. Lind, High-order simulations of isothermal flows using the local anisotropic basis function method (LABFM), J. Comput. Phys. 449 (2022) 110760, http://dx.doi.org/10.1016/j.jcp.2021.110760.
[63] Y. Li, C. Gu, T. Dullien, O. Vinyals, P. Kohli, Graph matching networks for learning the similarity of graph-structured objects, 2019, arXiv:1904.12787.
[64] R. Xu, P. Stansby, D. Laurence, Accuracy and stability in incompressible SPH (ISPH) based on the projection method and a new approach, J. Comput. Phys. 228 (18) (2009) 6703–6725, http://dx.doi.org/10.1016/j.jcp.2009.05.032.
[65] S.J. Lind, R. Xu, P.K. Stansby, B.D. Rogers, Incompressible smoothed particle hydrodynamics for free-surface flows: A generalised diffusion-based algorithm for stability and validations for impulsive flows and propagating waves, J. Comput. Phys. 231 (4) (2012) 1499–1523, http://dx.doi.org/10.1016/j.jcp.2011.10.027.
[66] G. Oger, S. Marrone, D.L. Touzé, M. de Leffe, SPH accuracy improvement through the combination of a quasi-Lagrangian shifting transport velocity and consistent ALE formalisms, J. Comput. Phys. 313 (2016) 76–98, http://dx.doi.org/10.1016/j.jcp.2016.02.039.
[67] N. Flyer, G.A. Barnett, L.J. Wicker, Enhancing finite differences with radial basis functions: Experiments on the Navier–Stokes equations, J. Comput. Phys. 316 (2016) 39–62, http://dx.doi.org/10.1016/j.jcp.2016.02.078.
[68] H. Kraus, J. Kuhnert, A. Meister, P. Suchde, A meshfree point collocation method for elliptic interface problems, Appl. Math. Model. 113 (2023) 241–261, http://dx.doi.org/10.1016/j.apm.2022.08.002.
[69] T. Jacquemin, P. Suchde, S.P.A. Bordas, Smart cloud collocation: Geometry-aware adaptivity directly from CAD, Comput.-Aided Des. 154 (2023) 103409, http://dx.doi.org/10.1016/j.cad.2022.103409.
[70] P. Suchde, T. Jacquemin, O. Davydov, Point cloud generation for meshfree methods: An overview, Arch. Comput. Methods Eng. 30 (2) (2023) 889–915, http://dx.doi.org/10.1007/s11831-022-09820-w.
[71] M. Lin, G. Fourtakas, B.D. Rogers, A novel high-order spectral incompressible smoothed particle hydrodynamics (ISPH) scheme with an immersed boundary method (IBM), J. Comput. Phys. 540 (2025) 114264, http://dx.doi.org/10.1016/j.jcp.2025.114264.
[72] Z.-F. Meng, P.-N. Sun, P.-P. Wang, B.C. Khoo, A.-M. Zhang, High-order SPH: A review of the method and applications, Arch. Comput. Methods Eng. (2025) http://dx.doi.org/10.1007/s11831-025-10346-0.
[73] E. Lamballais, V. Fortuné, S. Laizet, Straightforward high-order numerical dissipation via the viscous term for direct and large eddy simulation, J. Comput. Phys. 230 (9) (2011) 3270–3275, http://dx.doi.org/10.1016/j.jcp.2011.01.040.
[74] H. Broadley, J. King, S. Lind, Improving the accuracy of meshless methods via resolving power optimisation using multiple kernels, J. Comput. Phys. 566 (2026) 115238, http://dx.doi.org/10.1016/j.jcp.2026.115238.
[75] A. Brandenburg, Computational aspects of astrophysical MHD and turbulence, in: Advances in Nonlinear Dynamos, CRC Press, 2003, pp. 269–344, http://dx.doi.org/10.1201/9780203493137.ch9.
[76] J. Bonet, T.-S. Lok, Variational and momentum preservation aspects of smooth particle hydrodynamic formulations, Comput. Methods Appl. Mech. Engrg. 180 (1) (1999) 97–115, http://dx.doi.org/10.1016/S0045-7825(99)00051-1.
[77] D.J. Price, Smoothed particle hydrodynamics and magnetohydrodynamics, J. Comput. Phys. 231 (3) (2012) 759–794, http://dx.doi.org/10.1016/j.jcp.2010.12.011, Special Issue: Computational Plasma Physics.
[78] H.-G. Lyu, P. Sun, A. Colagrossi, A.-M. Zhang, Towards SPH simulations of cavitating flows with an EoSB cavitation model, Acta Mech. Sin. 39 (2022) http://dx.doi.org/10.1007/s10409-022-22158-x.
[79] A. Jameson, W. Schmidt, E. Turkel, Numerical solution of the Euler equations by finite volume methods using Runge–Kutta time stepping schemes, in: Proceedings of the 14th Fluid and Plasma Dynamics Conference, AIAA, 1981, http://dx.doi.org/10.2514/6.1981-1259.
[80] N. Flyer, G.B. Wright, B. Fornberg, Radial basis function-generated finite differences: A mesh-free method for computational geosciences, in: Handbook of Geomathematics, Springer Berlin Heidelberg, 2020, pp. 1–30, http://dx.doi.org/10.1007/978-3-642-27793-1_61-1.
[81] T. Lin, Y. Wang, X. Liu, X. Qiu, A survey of transformers, 2021, arXiv:2106.04554.
[82] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. Salakhutdinov, A. Smola, Deep sets, 2018, arXiv:1703.06114.
[83] S. Batzner, A. Musaelian, L. Sun, M. Geiger, J.P. Mailoa, M. Kornbluth, N. Molinari, T.E. Smidt, B. Kozinsky, E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials, Nat. Commun. 13 (1) (2022) 2453, http://dx.doi.org/10.1038/s41467-022-29939-5.
[84] V.G. Satorras, E. Hoogeboom, M. Welling, E(n) equivariant graph neural networks, 2022, http://dx.doi.org/10.48550/arXiv.2102.09844, arXiv:2102.09844.
[85] T. Jacquemin, S.P.A. Bordas, A unified algorithm for the selection of collocation stencils for convex, concave, and singular problems, Internat. J. Numer. Methods Engrg. 122 (16) (2021) 4292–4312, http://dx.doi.org/10.1002/nme.6703.
[86] S. Deshpande, J. Lengiewicz, S.P.A. Bordas, Probabilistic deep learning for real-time large deformation simulations, Comput. Methods Appl. Mech. Engrg. 398 (2022) 115307, http://dx.doi.org/10.1016/j.cma.2022.115307.
[87] S. Deshpande, H. Rappel, M. Hobbs, S.P.A. Bordas, J. Lengiewicz, Gaussian process regression + deep neural network autoencoder for probabilistic surrogate modeling in nonlinear mechanics of solids, Comput. Methods Appl. Mech. Engrg. 437 (2025) 117790, http://dx.doi.org/10.1016/j.cma.2025.117790.