MACHINE LEARNING
Science and Technology

PURPOSE-LED PUBLISHING™ Part of the MACHINE LEARNING Series

PAPER • OPEN ACCESS

An attention-based spatio-temporal neural operator for evolving physics

To cite this article: Vispi Karkaria *et al* 2025 *Mach. Learn.: Sci. Technol.* **6** 045036

IOP Publishing Mach. Learn.: Sci. Technol. 6 (2025) 045036 https://doi.org/10.1088/2632-2153/ae1277

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
13 June 2025

REVISED
24 September 2025

ACCEPTED FOR PUBLICATION
13 October 2025

PUBLISHED
6 November 2025

PAPER

An attention-based spatio-temporal neural operator for evolving physics

Vispi Karkaria[1], **Doksoo Lee**[1], **Yi-Ping Chen**[1], **Yue Yu**[2,*] and **Wei Chen**[1,*]

[1] Department of Mechanical Engineering, Northwestern University, Evanston, IL 60208, United States of America
[2] Department of Mathematics, Lehigh University, Bethlehem, PA 18015, United States of America
[*] Authors to whom any correspondence should be addressed.

E-mail: yuy214@lehigh.edu and weichen@northwestern.edu

Keywords: spatio-temporal neural operator, uncertainty quantification, scientific machine learning, out-of-distribution generalization, interpretability, attention mechanism, PDE modeling

Abstract

In scientific machine learning (SciML), a key challenge is learning unknown, evolving physical processes and making predictions across spatio-temporal scales. For example, in real-world manufacturing problems like additive manufacturing, users adjust known machine settings while unknown environmental parameters simultaneously fluctuate. To make reliable predictions, it is desired for a model to not only capture long-range spatio-temporal interactions from data but also adapt to new and unknown environments; traditional machine learning models excel at the first task but often lack physical interpretability and struggle to generalize under varying environmental conditions. To tackle these challenges, we propose the attention-based spatio-temporal neural operator (ASNO), a novel architecture that combines separable attention mechanisms for spatial and temporal interactions and adapts to unseen physical parameters. Inspired by the backward differentiation formula, ASNO learns a transformer for temporal prediction and extrapolation and an attention-based neural operator for handling varying external loads, enhancing interpretability by isolating historical state contributions and external forces, enabling the discovery of underlying physical laws and generalizability to unseen physical environments. Empirical results on SciML benchmarks demonstrate that ASNO outperforms existing models, establishing its potential for engineering applications, physics discovery, and interpretable machine learning.

Nomenclature

Symbol	Meaning
X, x	State vector (general)
X_m	State at timestep m
$\tilde{X}_{m+1}$	Extrapolated (homogeneous) state
F	External forcing/loading field
S	Hidden system (environment) state
t	Continuous time
m	Time-index
n	History length
Δt	Time step size
α_k, β	BDF coefficients (see (2))
$\mathcal{D}^{(\eta)}$	Dataset for system η
$\mathcal{F}$	ASNO operator mapping
$\hat{y}_{t+1}$	Predicted output at time $t+1$
y_t, $\hat{y}_t$	True and predicted values at time t
E_T	Cumulative error over T steps
e_t	Instantaneous error at timestep t
Q, K, V	Query, Key, and Value matrices used in attention mechanisms; used generically, with context-specific definitions below
W_E	Embedding matrix
W_{Tq}, W_{Tk}, W_{Tv}	Time-series transformer encoder attention weight matrices for Query (Q), Key (K), and Value (V) used in the explicit extrapolation step
P_k	Positional encoding at position k
d_{embed}, d_t, d	embedding dimension, key/query dimension, and feature dimension
H_t	Penultimate-step latent features used in computing the nonlocal attention operator (NAO) kernel weights for h and f
J_t	Intermediate NAO state at attention step t composed of latent features and forcing fields (H_t, F_t), iteratively updated across T layers
$W_{P,h}$, $W_{P,f}$, W_{Q_t}, W_{K_t}	Weight matrices in the NAO used for computing kernel projections; W_{Q_t} and W_{K_t} generate attention Query and Key vectors used in the implicit interaction modeling
$K_\cdot$	Learned nonlocal kernel operator acting over latent space in NAO
$\mathcal{H}$, $\mathcal{F}$	Input and output Banach function spaces
T	Number of attention steps in NAO

1. Introduction

The increasing complexity of spatio-temporal data in scientific fields such as fluid dynamics, material science, and manufacturing, particularly in the context of digital twin technologies, has made the interpretability and accuracy of machine learning models critically important. Digital twins-virtual replicas of physical systems-are becoming integral to modern manufacturing processes, enabling real-time monitoring, simulation, and optimization of physical assets (Kapteyn *et al* 2021, van Beek *et al* 2023, Karkaria *et al* 2024a). In such settings, the critical challenge lies in creating models that generalize to new physical parameters (e.g. PDE coefficients) while maintaining alignment with underlying physical principles to enable robust predictions and insights into system behavior. This dual capability enhances decision-making and facilitates system optimization by aligning predictions with real-world physics (Thelen *et al* 2022, Karkaria *et al* 2024b, Chen *et al* 2025).

Existing methods often struggle to generalize across unseen physical environments, particularly when tasked with interpreting spatial and temporal dependencies in a separable yet synergistic manner. Neglecting this separability can obscure the physical mechanisms driving system behavior, resulting in less interpretable and reliable predictions (Molnar 2020, Rudin *et al* 2022). For example, in material modeling, observable phenomena such as deformation fields may provide limited information, and inferring hidden parameters such as material properties remains an ill-posed problem. Similarly, in physics-based applications like manufacturing, accurately capturing the interplay between spatial interactions and evolving temporal dynamics is essential for refining product quality and operational efficiency (Ko *et al* 2022).

While existing data-driven surrogate architectures-such as transformer encoders, U-Net, DeepONet, and newer operator-based frameworks like the General Neural Operator Transformer (GNOT), Transolver, and Fourier neural operator (FNO) have been developed to capture spatio-temporal dependencies in physical systems (Zhang *et al* 2018, Lu *et al* 2021, Hao *et al* 2023, Wang *et al* 2024, Wu *et al* 2024), they often lack inherent mechanisms for reliable generalization to unseen conditions or for

disentangling spatial and temporal influences. Transformers, despite their strength in modeling long-range temporal correlations via self-attention, frequently underrepresent the spatial interactions essential for nonlinear phenomena such as turbulent flows, reaction-diffusion processes, and phase transitions. Moreover, many of these approaches struggle to separate the roles of temporal evolution and spatial coupling, which limits their applicability in PDE-governed settings. Finally, their adaptability to systems with varying PDE coefficients remains limited, reducing their effectiveness across diverse physical scenarios. While the specific implementations vary-from sequence-based encoders to spectral and attention-driven operators-all of these methods share the common challenge of balancing expressivity with robust, physics-informed generalization.

To address these limitations, we propose the Attention-based Spatio-Temporal Neural Operator (ASNO). In ASNO, a transformer encoder captures temporal dynamics in the solution field, while an attention-based nonlocal operator handles the long-range spatial dependencies and the interplay between solution and loading/environment fields. Inspired by the implicit-explicit (IMEX) scheme, ASNO leverages the backward differentiation formula (BDF) to separate temporal effects from spatial loading/solution interactions (Fredebeul 1998, Hu and Shu 2021). This separable architecture enhances generalizability to unseen physical parameters (e.g. initial conditions, loadings, environments) while improving interpretability.

The key contributions of this paper are as follows:

- We introduce ASNO, a novel architecture inspired by the IMEX scheme, which discovers separate mechanisms for capturing temporal effects and spatial loading/solution interactions.
- ASNO discovers a spatio-temporal relationship from data, enabling zero-shot generalizability to PDEs with unseen initial conditions, loadings, and environments.
- The attention mechanism provides insights into the separable contributions of temporal and spatial dynamics, overcoming the limitations of models focused on only one of the two.
- We conduct experiments on examples ranging from chaotic systems and PDE solving problems to real-world applications. Results demonstrate the advantages of ASNO over baseline transformer models and neural operator models in terms of generalizability, long-term stability, and interpretability.

The rest of this paper is organized as follows. In section 2, we review the theoretical background on implicit–explicit IMEX schemes and BDF, as well as prior work on transformer encoders and neural operators. Section 3 introduces the ASNO architecture, detailing (i) the transformer encoder used for temporal extrapolation and (ii) the NAO that performs spatial coupling and forward solves via a learned kernel. In section 4, we present empirical results on four benchmarks, namely Darcy flow, the Lorenz system, Navier–Stokes, and a directed energy deposition (DED) case study, demonstrating ASNO's accuracy, stability, and zero-shot generalizability. Finally, section 5 concludes with a summary of our contributions and outlines avenues for future work.

2. Background and related work

2.1. Transformers for time series data

Traditional time series models such as autoregressive integrated moving average and long short-term memory have effectively captured short-term dependencies but face limitations when dealing with long-range dependencies and highly non-linear, multivariate data (Nelson 1998, Graves and Graves 2012). Transformers, initially developed for natural language processing, revolutionized time series modeling by introducing self-attention mechanisms that capture dependencies across long sequences (Vaswani *et al* 2017, Tang and Matteson 2021, Zerveas *et al* 2021, Zhou *et al* 2021). This attention mechanism allows transformers to weigh different parts of the input sequence based on their relevance, making them highly suitable for long-sequence tasks (Zhao *et al* 2024).

Despite their strengths, transformers can be computationally expensive. Their complexity grows quadratically with sequence length, limiting scalability, especially when applied to large datasets (Fournier *et al* 2023). This has driven the development of models like Informer (Zhou *et al* 2021) and Reformer (Kitaev *et al* 2020), which employ sparse attention mechanisms to reduce complexity. Sparse attention mechanisms restrict the set of key-query interactions to locally or globally selected tokens, such as sliding windows, strided patterns, or learned global pivots, thereby lowering both memory usage and computational cost without significantly sacrificing modeling capacity. Models such as the temporal fusion transformer (TFT) have improved upon traditional transformers by using attention scores to enhance interpretability and effectively address practical challenges like variable input lengths and missing data (Lim *et al* 2021); in TFT, attention scores are computed per feature and timestep,

providing a transparent measure of each input's influence on the prediction and facilitating feature-level explanations. These features are byproducts of TFT's core innovations in handling complex temporal dependencies.

These developments underscore the adaptability of transformer architectures in handling diverse temporal tasks, though they remain limited in their ability to generalize to new physical parameters (e.g. PDE coefficients) and lack mechanisms that explicitly separate spatial and temporal interactions for interpretability (Cheng *et al* 2024).

2.2. Neural operators in scientific machine learning (SciML)

Neural operators have become essential in SciML, particularly for modeling mappings between function spaces (Li *et al* 2020a). These models are effective at solving forward problems in physics-based systems, such as those governed by PDEs, by serving as efficient surrogates for physical systems and offering black-box approximations without explicitly interpreting the underlying physical laws (Wen *et al* 2022, Kovachki *et al* 2023, Azizzadenesheli *et al* 2024, Cao *et al* 2024). For instance, DeepONet utilizes separate networks to encode input functions and evaluation points, thereby learning the nonlinear operator that maps function inputs to output values at arbitrary spatial locations (Lu *et al* 2019, 2021). However, as the problem's dimensionality increases, DeepONet faces scalability issues (Mandl *et al* 2024). Similarly, FNO, which operates in the frequency domain to better capture global spatial dependencies, becomes computationally expensive either at higher resolutions or when handling non-smooth data (Li *et al* 2020a). Recent advancements, such as the GNOT and Transolver, have improved the flexibility of neural operators for handling complex, spatially distributed data. GNOT incorporates heterogeneous normalized attention layers and geometric gating mechanisms, enabling it to address multi-scale problems and integrate diverse data sources effectively (Hao *et al* 2023). Similarly, Transolver leverages transformer-based architectures to solve PDEs on irregular geometries, enhancing its adaptability to various applications. Beyond these, several recent works have pushed the capabilities of neural operators further: the Galerkin transformer integrates spectral methods with attention to improve resolution and stability in PDE solvers (Cao 2021); the multipole graph neural operator scales efficiently to large domains by modeling long-range dependencies via multipole expansions (Li *et al* 2020b); the spectral neural operator (SNO) learns flexible, data-driven spectral filters to better handle non-smooth and multiscale signals (Rafiq *et al* 2022); and the Hierarchical Tensor Neural Operator uses tensorized representations to scale to ultra-high-dimensional problems (Liu *et al* 2022). These developments highlight the rapidly evolving landscape of neural operator research and provide valuable context for situating the present ASNO framework within ongoing efforts to improve scalability, adaptability, and physical fidelity. However, these models often struggle to generalize to unseen PDE parameters (e.g. new initial conditions or environments) and lack interpretability in disentangling spatial and temporal interactions (Wang *et al* 2024, Wu *et al* 2024).

Most neural operators primarily address forward PDE problems, predicting future states from current conditions by capturing spatio-temporal dependencies, but struggle with inverse problems, like reconstructing unknown PDE parameters, which often require prior knowledge of governing equations and are generally ill posed (Molinaro *et al* 2023). Integrating forward and inverse problem solving capabilities into a single spatio-temporal framework enables the development of more versatile models that can not only predict system behaviors but also infer hidden states and causal mechanisms. This dual functionality enhances model robustness and adaptability, qualities vital for scientific applications where elucidating system dynamics, which may inherently encode underlying causes, contributes to more informed simulation, and potentially supports control and optimization efforts, even if these are ultimately executed through computational rather than interpretative means (Kim and Lee 2024).

3. Model architecture

This section presents ASNO, a spatio-temporal neural operator integrating the transformer encoder for temporal dependencies and the nonlocal attention mechanism for spatial interactions. We outline the transformer encoder, followed by the NAO model, and explain their combined role in the form of a BDF for complex spatio-temporal prediction.

3.1. BDF

BDF is a family of popular numerical schemes for stiff differential equations, thanks to its high-order accuracy and large region of absolute stability (Fredebeul 1998). Here, $\dot{X}$ denotes the derivative of X with respect to the time variable t, stating that this is a temporal derivative. For a initial-value differential equation:

$$\dot{X} = F(t, X), \quad X(t_0) = X_0. \tag{1}$$

the general formula for a BDF can be written as

$$\sum_{k=1}^{n} \alpha_k X_{m-k+1} + X_{m+1} = \Delta t \beta F((m+1)\Delta t, X_{m+1}). \tag{2}$$

Here, Δt denotes the time step size, and X_m is the approximated solution at time instance $t_0 + m\Delta t$. Notice that (2) can be naturally decomposed as the two-phase implicit-explicit (IMEX) scheme (Ascher *et al* 1995):

$$-\sum_{k=1}^{n} \alpha_k X_{m-k+1} = \tilde{X}_{m+1}, \tag{3}$$

$$\tilde{X}_{m+1} - X_{m+1} + \Delta t \beta F((m+1)\Delta t, X_{m+1}) = 0. \tag{4}$$

In the explicit step (3), n historical steps are accumulated, hence it characterizes the temporal interactions. Notice that when the system is homogeneous, i.e. $F \equiv 0$, we have $\tilde{X}_{m+1} = X_{m+1}$. That means, the explicit steps generate an estimated temporal extrapolated solution for a system without external loading F. On the other hand, the implicit step (4) takes $\tilde{X}_{m+1}$ as the input and tries to solve a (possibly nonlinear) static equation of X_{m+1}. Hence, this step is characterized by capturing the spatial interaction between components of X, as well as their interplay with the external loading F.

Based on this idea, our key innovation in ASNO is to design a separable architecture that captures the temporal and spatial effects respectively. In particular, our architecture resembles the implicit-explicit decomposition of BDF. We notice that the external loading F is non-autonomous, meaning that it depends not only on X but also on another (hidden) time variant quantity, which can be seen as the hidden system state, S_{m+1}, to be discovered through learning.

We now formally state the learning settings in ASNO: consider multiple physical systems which can be described by a series of initial-valued problems:

$$\dot{X}^{(\eta)} = F\left(S^{(\eta)}, X^{(\eta)}\right), \; X^{(\eta)}(t_0) = X_0^{(\eta)}, \tag{5}$$

where $\eta = 1, \cdots, S$ is the sample/system index. Then, for each system, we assume a sequence of observations:

$$\mathcal{D}^{(\eta)} = \left\{X^{(\eta)}(m\Delta t), F^{(\eta)}(m\Delta t)\right\}_{m=0}^{T/\Delta t} \tag{6}$$

are provided. The goal of ASNO are three folds:

1. Identify a temporal extrapolation rule (3) from data, for a stable and accurate long-term prediction of X.
2. For each system with hidden state $S^{(\eta)}$, infer the underlying context or structure of $S^{(\eta)}$ directly from observational data, without relying on labeled supervision or prior knowledge of the state.
3. For each new and unseen system with the first n steps of (X, F) given, provide a zero-shot prediction rule without training.

3.2. Explicit step: temporal extrapolation

As the first component of ASNO, we employ a transformer encoder to resemble the explicit step (3). In particular, it processes time series data by capturing long-range temporal dependencies, and predicts future states of systems governed by high-dimensional, nonlinear dynamical equations.

In transformer encoder, the input to the encoder is a sequence of observations from previous time steps, denoted as $\{X_m, X_{m-1}, X_{m-2}, \ldots, X_{m-n+1}\}$, where n represents the number of past time steps considered. Each observation $X_t \in \mathbb{R}^d$ is projected into an embedding space of dimension d_{embed}, with positional encodings added to preserve temporal order:

$$E_t = X_t W_E + P_t, \tag{7}$$

where $W_E \in \mathbb{R}^{d \times d_{\text{embed}}}$ is the embedding matrix, and $P_t \in \mathbb{R}^{d_{\text{embed}}}$ is the positional encoding vector. The positional encoding vector injects time-specific bias into each embedding, enabling the model to distinguish tokens based on their relative and absolute positions in the sequence and thereby preserve temporal order. Positional encoding is implemented as a separate preprocessing layer applied to the embeddings before they enter the transformer encoder layers, and is not part of the self-attention computations themselves.

The encoded sequence $\{E_m, E_{m-1}, \ldots, E_{m-n+1}\}$ is then passed through multiple layers of the transformer encoder, each consisting of a multi-head self-attention mechanism and a feed-forward network, allowing the model to learn complex dependencies across time steps.

In the attention mechanism (Niu *et al* 2021), the query, key, and value matrices are derived from the encoded input sequence using weights $W_{Tq}, W_{Tk}, W_{Tv} \in \mathbb{R}^{d_{\text{embed}} \times d_t}$, which are learnable parameters for the query, key, and value transformations, respectively.

$$Q = E_t W_{Tq}, \quad K = E_t W_{Tk}, \quad V = E_t W_{Tv}, \tag{8}$$

For each system, the transformer encoder processes the previous n embeddings, $\{E_m, E_{m-1}, \ldots, E_{m-n+1}\}$, and produces a single latent vector H_{m+1}, summarizing all temporal information from those steps into one representation.

The attention score between the query at the tth time step, Q_t, and the keys K from previous steps is calculated as a weighted sum, capturing dependencies between the current time step and previous steps. Formally, for the m-th time step, the future latent space H_{m+1} is predicted using the attention-weighted sum of past values:

$$H_{m+1} = \text{TE}(X_m, X_{m-1}, \ldots, X_{m-n+1}) = \sum_{i=1}^{n} \alpha_{m,i} \cdot V_m, \tag{9}$$

where $\alpha_{m,i}$ represents the attention weights between Q_m and K_{m-i+1}, computed as

$$\alpha_{m,i} = \frac{\exp\left(\frac{Q_m \cdot K_{m-i+1}^{\top}}{\sqrt{d_t}}\right)}{\sum_{j=1}^{n} \exp\left(\frac{Q_m \cdot K_{m-j+1}^{\top}}{\sqrt{d_t}}\right)}, \tag{10}$$

where d_t is the dimensionality of the key vectors. This enables the transformer to focus on the most relevant past steps when extrapolating the latent state.

The predicted latent variable $H_{m+1} \in \mathbb{R}^{d_{\text{embed}}}$ corresponds to the homogeneous BDF extrapolation $\tilde{X}_{m+1}$ and is passed into the NAO for the implicit spatial correction step (4).

3.3. Implicit step: static PDE solver

For the implicit step (4), we take $\tilde{H}_{m+1}$ and F_{m+1} as inputs, and aim to approximate the solution X_{m+1}. Since (4) can be seen as a nonlinear static PDE solver, we propose to learn a neural operator as the surrogate solution operator. However, due to the possible change of underlying system state, S, generic neural operators (Lu *et al* 2019, Li *et al* 2020a, You *et al* 2022) would fail this task because they focus on the solution of one single PDE. Herein, we propose to employ the attention mechanism, in the form of the NAO (Yu *et al* 2024), which combines attention and neural operator learning, offering a generalizable solution operator across different PDEs. In particular, NAO expands on traditional attention by effectively modeling long-range spatial dependencies crucial for capturing continuous, nonlocal interactions within physical systems (Yu *et al* 2024). This approach allows NAO to aggregate global context, addressing limitations of standard attention in handling nonlocal, spatially distributed information, especially beneficial for high-dimensional data processing in fluid dynamics and thermodynamics.

Formally, NAO is designed to solve both forward and inverse problems, with the underlying system dynamics encapsulated by operators that map input functions $h \in \mathcal{H}$ to output functions $g \in \mathcal{F}$. This operator is defined as:

$$L_K[h] + \epsilon = g, \tag{11}$$

where $\mathcal{H}$ and $\mathcal{F}$ are Banach spaces, K is the nonlocal interaction kernel, and ϵ denotes additive noise accounting for model-system discrepancies. NAO introduces a critical component: a kernel map constructed via attention. This kernel map estimates the nonlocal kernel based on both input h and output g, enabling the model to capture global context across physical states.

Herein, we take h as the latent variable field H_{m+1} together with the loading fields F, and g as the prediction of the next time step X_{m+1}. The latent variable field H_{m+1} is constructed by passing the temporal features from the transformer encoder through a projection layer that reshapes them onto the spatial mesh, ensuring that each location y_k contains a d-dimensional embedding summarizing the system's past temporal dynamics. The loading field F is obtained directly from known external forcings or boundary conditions and is aligned to the same spatial discretization for compatibility in the NAO mapping.

The latent variable and loading fields are discretized over the spatial mesh $\{y_k\}_{k=1}^N$ as

$$\begin{aligned} H_{1:d} &= \left(H_j(y_k)\right)_{1\leqslant j\leqslant d,\ 1\leqslant k\leqslant N}, \\ F_{1:d} &= \left(F_j(y_k)\right)_{1\leqslant j\leqslant d,\ 1\leqslant k\leqslant N}, \end{aligned} \tag{12}$$

where d denotes the number of feature channels at each spatial location y_k, such as the dimensions of the latent embedding vector and the loading fields, ensuring that $H_{1:d}$ and $F_{1:d}$ fully represent all relevant channels of the system.

The overall process in NAO iteratively transforms an initial state

$$J_0 = (H_{1:d}, F_{1:d})$$

through T steps, each applying attention with residual connections:

$$J_t = \mathrm{Attn}\,(J_{t-1};\theta_t)\,J_{t-1} \;+\; J_{t-1}, \quad 1 \leqslant t \leqslant T, \tag{13}$$

where at each iteration we explicitly maintain the same block structure

$$J_t = \left(H_{1:d}^{(t)}, F_{1:d}^{(t)}\right) \tag{14}$$

with $H_{1:d}^{(t)}$ the updated latent variables after t steps and $F_{1:d}^{(t)}$ the corresponding loading fields.

Here

$$\mathrm{Attn}\,[J;\theta_t] = \sigma\left(\tfrac{1}{\sqrt{d_k}}\, J W_{Q_t}\,(W_{K_t})^\top\, J^\top\right). \tag{15}$$

After T steps, we form the learned kernel via a kernel map:

$$\begin{aligned} K[H_{1:d}, F_{1:d};\theta] = {} & W_{P,h}\,\sigma\left(\tfrac{1}{\sqrt{d_k}}\,(J_T)^\top\, W_{Q_{T+1}}\,\left(W_{K_{T+1}}\right)^\top J_T\right) \\ & + W_{P,f}\,\sigma\left(\tfrac{1}{\sqrt{d_k}}\,(F_T)^\top\, W_{Q_{T+1}}\,\left(W_{K_{T+1}}\right)^\top J_T\right). \end{aligned} \tag{16}$$

Here, $\theta = \{W_{P,h}, W_{P,f}, W_{Q_t}, W_{K_t}\}$ collects all the trainable weight matrices used in the kernel map. and compute the output

$$X_{1:d}^{\mathrm{out}}(y) = \int K[H_{1:d}, F_{1:d}]\,(y,z)\,F_{1:d}(z)\,\mathrm{d}z. \tag{17}$$

In this architecture, $\theta = \{W_{P,h}, W_{P,f}, W_{Q_t}, W_{K_t}\}$ are trainable, W_{Q_t}, W_{K_t} are query/key matrices, σ is a linear activation function following the suggestions in Cao (2021), Yu *et al* (2024), Lu and Yu (2025). $W_{P,h}$ and $W_{P,f}$ are projection matrices that map the attention outputs to the latent variable channel H and the loading field channel F, respectively, enabling separate handling of spatial interactions from the latent and forcing components. To optimize θ, we minimize the training loss across all S systems:

$$\mathcal{L} = \sum_{\eta=1}^{S} \left\| \int K[H_{1:d}, F_{1:d}]\,(y,z)\,F_{1:d}(z)\,\mathrm{d}z - X_{1:d}^{\mathrm{out,true}}(y) \right\|_{L^2}^2. \tag{18}$$

Here, $X_{1:d}^{\mathrm{out,true}}(y)$ denotes the ground-truth target measurements of the output field from the dataset for each system η, and for full notational clarity the norm may be written as

$$\left\| \int K[H_{1:d}, F_{1:d}]\,(y,z)\,F_{1:d}(z)\,\mathrm{d}z - X_{1:d}^{\mathrm{out,true},(\eta)}(y) \right\|_{L^2}^2 \tag{19}$$

making the dependence on η explicit. The NAO thus models nonlocal spatial interactions via a data-dependent kernel on the latent space H, capturing complex system states S across diverse PDEs and enhancing generalizability.

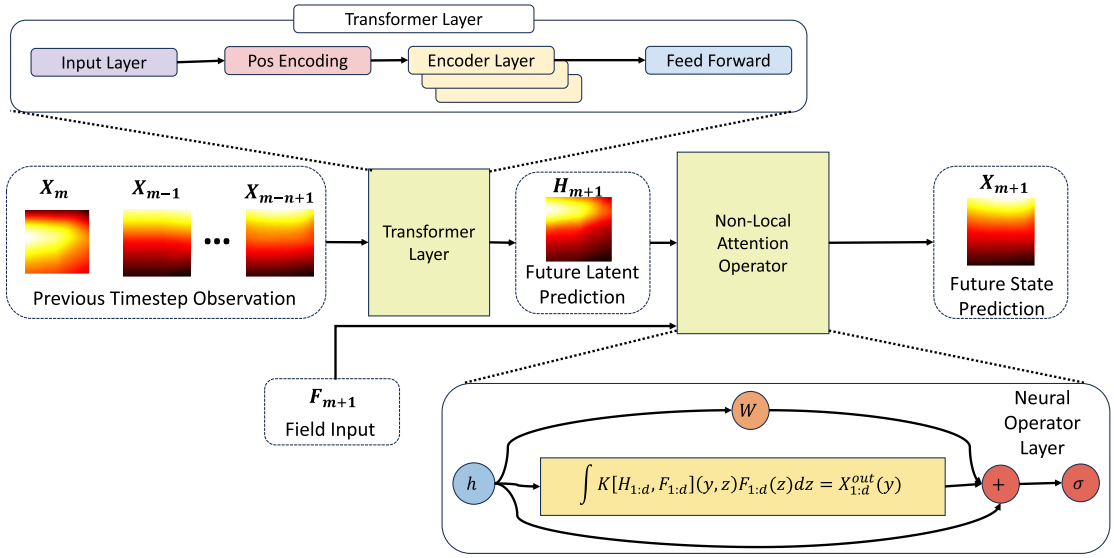

Figure 1. The ASNO framework with the transformer encoder for capturing temporal dependencies and the Non-Local Attention Operator (NAO) for modeling spatial interactions, following a backward differentiation formula (BDF) approach.

3.4. ASNO: summary and algorithm

The ASNO architecture combines the transformer encoder and the NAO to effectively capture both temporal and spatial dependencies in complex spatio-temporal data, as shown in figure 1.

As the transformer encoder processes the past observations and outputs the latent variable H_{m+1} using equation (9), H_{m+1} is fed into the NAO to provide the temporal information. This process can be represented as:

$$\begin{aligned} X^{\text{out}}_{m+1} &= \text{NAO}\left(H_{m+1}, F_{m+1}\right) \\ &= \text{NAO}\left(\text{TE}\left(X_m, X_{m-1}, \ldots, X_{m-n+1}\right), F_{m+1}\right) \\ &= \text{ASNO}\left(X_m, X_{m-1}, \ldots, X_{m-n+1}, F_{m+1}\right). \end{aligned} \tag{20}$$

Since the temporal information is compressed into a latent representation by the transformer encoder, and the NAO leverages it to learn spatial relationships, the combination of the two modules allows ASNO to learn spatio-temporal dependencies. The transformer encoder and NAO are jointly trained with a loss function that optimizes the model's ability to capture complex dependencies efficiently, making it adaptable to a wide range of scientific machine-learning tasks.

4. Experiments

In this section, we evaluate ASNO's performance on case studies involving chaotic systems and spatio-temporal PDEs, each chosen to illustrate its strengths in capturing complex dynamical behaviors crucial to SciML.

4.1. Learning a PDE solution operator for darcy flow

We evaluate ASNO on a dynamic Darcy flow equation, which models fluid flow through porous media and exhibits complex spatio-temporal dynamics. On a square domain $\Omega := [0,1]^2$, The governing equation writes:

$$\frac{\partial p\left(t,x\right)}{\partial t} - \nabla \cdot \left(b\left(x\right)\nabla p\left(t,x\right)\right) = g\left(t,x\right), \quad x \in \Omega, \qquad p\left(x\right) = 0, \quad \mathbf{x} \in \partial\Omega. \tag{21}$$

In this context, we aim to learn the solution operator of Darcy's equation and compute the pressure field $p(t,x)$, given the source field $g(t,x)$ and the underlying permeability field $b(x)$. In our dataset, both the pressure field $p(t,x)$ and the source field $g(t,x)$ are explicitly available, while the permeability field $b(x)$ is treated as a hidden parameter field that varies between samples and influences the dynamics implicitly. The training dataset consists of 100 time-series profiles on a 21×21 grid, each with 100 timesteps. Here, each profile is determined by a different hidden microstructure field $b(x)$. Input-output pairs are

generated via a sliding window of five consecutive timesteps (i.e. stride 1), yielding 96 samples per profile. We apply 20 random permutations per trajectory, resulting in 153 600 training samples and 38 400 test samples (80:20 split). All baseline models (FNO, U-Net, Transolver, GNOT, DeepONet, transformer encoder, transformer encoder + NAO) use the same inputs: the past five states $X_{m-4},\ldots,X_m$ and the current forcing field F_m, where the *state* X_m represents the pressure field at time step m and the *forcing field* F_m encodes the source or sink terms influencing fluid flow at each spatial location. For reproducibility, in our Darcy flow experiments we configure the transformer encoder with an embedding dimension $d_{\text{embed}} = 882$, $n_{\text{heads}} = 441$ attention heads, $n_{\text{layers}} = 3$ encoder layers, and dropout probability 0.1. The NAO is instantiated with $n_{\text{blocks}} = 2$ iterative attention layers, each using a single head ($r = 1$) with key/query dimension $d_k = 50$, and projection matrices $W_{P,h}, W_{P,f}$ mapping to the output grid size $S^2 = 441$. We train all models using the Adam optimizer with initial learning rate $\text{lr} = 3 \times 10^{-3}$, weight decay $\text{wd} = 10^{-2}$, and a StepLR scheduler that decays the learning rate by a factor of 0.7 every 100 epochs. Training runs for 25 epochs with batch size equal to the number of random permutations ($n_{\text{randperm}} = 20$), and we minimize a mean-squared-error loss implemented via our custom LpLoss. All experiments fix the random seed to 0 for reproducibility. Additionally, we conducted a sensitivity analysis to examine how the choice of key hyperparameters, specifically the temporal window size n and the number of attention steps T, influences both predictive performance and computational cost. Our findings indicate that increasing n generally improves short-term prediction accuracy by providing richer temporal context, but also increases memory usage and training time. Similarly, increasing T enhances the model's ability to capture long-range dependencies in spatial interactions, though with diminishing returns beyond a certain point and higher computational overhead. These trends suggest a trade-off between accuracy and efficiency, and the reported configuration reflects the best balance identified through preliminary experiments.

Table 1 compares the number of trainable parameters, GPU memory usage, test loss on the standard test set, and out-of-distribution (OOD) test losses for two variants of OOD data. Here, OOD refers to inputs whose underlying distributions differ from those seen during training. In our case, the OOD-f set uses time-varying source fields drawn from a different Gaussian random field (GRF) distribution than the training set, where a GRF is a spatial stochastic process in which the field values at any set of locations follow a joint Gaussian distribution, fully characterized by its mean function and covariance kernel. In this work, the GRF is generated using a stationary covariance function of the form $C(r) = \sigma^2 \exp\left(-\left(\frac{r}{\tau}\right)^{\alpha}\right)$, where $\alpha > 0$ controls the smoothness of the field, $\tau > 0$ is the correlation length scale controlling the rate of spatial correlation decay with distance r, and σ^2 is the variance. Larger values of α lead to smoother variations, while smaller τ values produce fields with more rapidly varying spatial patterns. Specifically, while training samples use source functions $g(t,x)$ drawn from a GRF with fixed hyperparameters $\alpha = 2$ and $\tau = 3$, the OOD-f samples are generated by modulating these GRFs with a sinusoidal time component and scaling their amplitudes by a factor of 200. In particular, the source term evolves over time as $g(t,x) = \tilde{g}(x) \cdot \sin(t)$, where $\tilde{g}(x)$ is the spatial GRF realization and $t = q\Delta t$ is the temporal index with $\Delta t = 5 \times 10^{-5}$. This construction introduces a dynamic temporal structure not present in the training distribution, thereby altering both the spatial and temporal characteristics of the source field. Meanwhile, the OOD-b set modifies the spatial distribution of the permeability field $b(x)$, which is a crucial hidden input in Darcy flow. While the training set uses binary bi-phase microstructures generated from a GRF with parameters $\alpha_\chi = 4$ and $\tau_\chi = 5$, the OOD-b set uses a rougher, higher-frequency microstructure generated from a GRF with $\alpha_\chi = 7$ and $\tau_\chi = 6$. This change leads to sharper, more fragmented phase boundaries and higher heterogeneity in the underlying permeability, directly impacting the operator's ability to generalize to previously unseen flow geometries. Generalizability is quantified by the extent to which the test loss increases on these OOD sets compared to the standard test set: smaller increases imply better robustness. ASNO achieves the lowest best and OOD test losses with a comparative number of trainable parameters among the neural-operator baselines, indicating the strongest OOD generalization with respect to both types of distributional shifts.

ASNO not only provides a high-performing predictor but also reveals underlying physical mechanisms through its learned representations. Its separable design helps to learn the implicit-explicit decomposition of backward differentiation formulas. Moreover, its modular structure enhances interpretability by allowing independent analysis of temporal and spatial contributions to the final prediction. To illustrate this, we compared the temporal predictions of the transformer encoder component with a classical BDF5 numerical approximation:

$$\tilde{X}_m = \frac{12}{137} X_{m-5} - \frac{75}{137} X_{m-4} + \frac{200}{137} X_{m-3} - \frac{300}{137} X_{m-2} + \frac{300}{137} X_{m-1}, \tag{22}$$

Table 1. Performance of different models on Darcy flow and Darcy OOD test sets.

Model	Trainable Params	GPU (MB)	Best test loss	Best OOD-f	Best OOD-b
ASNO	760 234	181	0.0368	**0.0673**	**0.0982**
FNO	900 224	214	0.0768	0.1129	0.1892
U-Net	820 994	**123**	0.1150	0.1523	0.2224
Transolver	810 573	422	0.0428	0.0721	0.1535
GNOT	760 349	208	0.0516	0.0811	0.1729
DeepONet	6230 000	2146	0.0537	0.0826	0.1826
Trans. Enc.	1620 394	173	0.0559	0.0927	0.1736
Linear Enc. + NAO	**720 398**	165	0.05 474	0.1245	0.1394

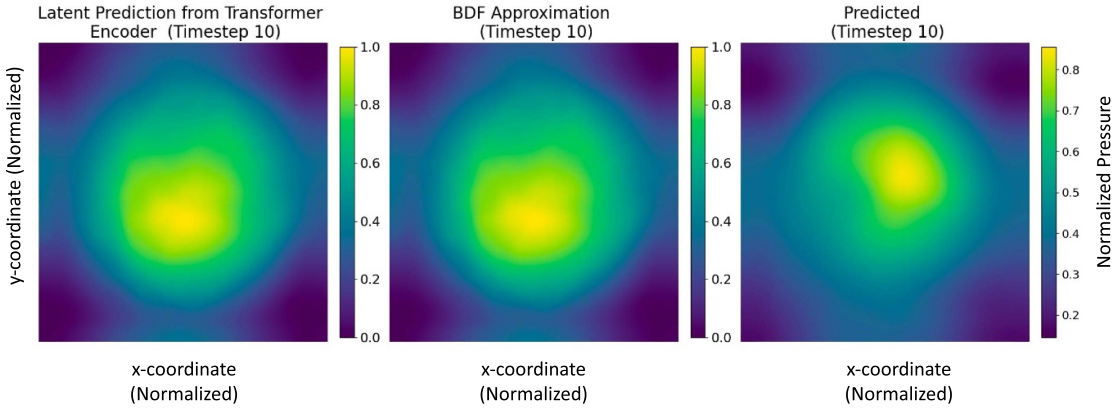

Figure 2. Comparison at Timestep 10 between the transformer encoder latent prediction (H_{m+1}), the BDF approximation ($\tilde{X}_{m+1}$), and the final ASNO output. The latent prediction aligns closely with the BDF estimate, while the NAO refines it using spatial field corrections.

showing that the learned latent representation H_m closely matches this high-order temporal extrapolation (see figure 2). For the spatial interactions captured by the NAO, we compared the learned kernel with a theoretical kernel derived from the Darcy discretization. Starting from

$$\tilde{X}_m = X_m + \frac{60}{137}\,\Delta t\,(A X_m + F_m)\,, \tag{23}$$

one obtains

$$X_m = \left(I + \tfrac{60}{137}\,\Delta t A\right)^{-1}\left(\tilde{X}_m - \tfrac{60}{137}\,\Delta t F_m\right)\,, \tag{24}$$

$$K_{\text{true}} = -\frac{60}{137}\,\Delta t\,\left(I + \tfrac{60}{137}\,\Delta t A\right)^{-1}\,. \tag{25}$$

Figure 3 compares the ground-truth kernel and the learned kernel from the NAO component, showing good agreement between the two. Additionally, the cumulative loss over time for various models illustrates that NAO outperforms existing methods in accuracy. When computing long-term time integration, each prediction X_t^{pred} is fed back as an input for the next step, so errors accumulate over successive rollouts. We observe that the error growth in the cumulative loss curves proceeds approximately linearly over time, which we attribute to the autoregressive nature of the rollout, where prediction errors in both the state variable X and the forcing term F (when applicable) propagate forward step-by-step. Since F is provided at each step from ground-truth measurements in our current setup, there is no additional compounding error from F itself; rather, the slope reflects the steady accumulation of state-prediction inaccuracies. The cumulative loss-also referred to as long-term time integration -is computed as

$$E_T \;=\; \sum_{t=1}^{T} \left\| X_t^{\text{true}} - X_t^{\text{pred}} \right\|_{L^2}\,, \tag{26}$$

where X_t^{true} and X_t^{pred} denote the ground-truth and predicted states at timestep t. This confirms that the spatial mechanism captured by NAO effectively reflects the physical structure of the underlying PDE and maintains strong predictive performance across timesteps. While our current experiments assume

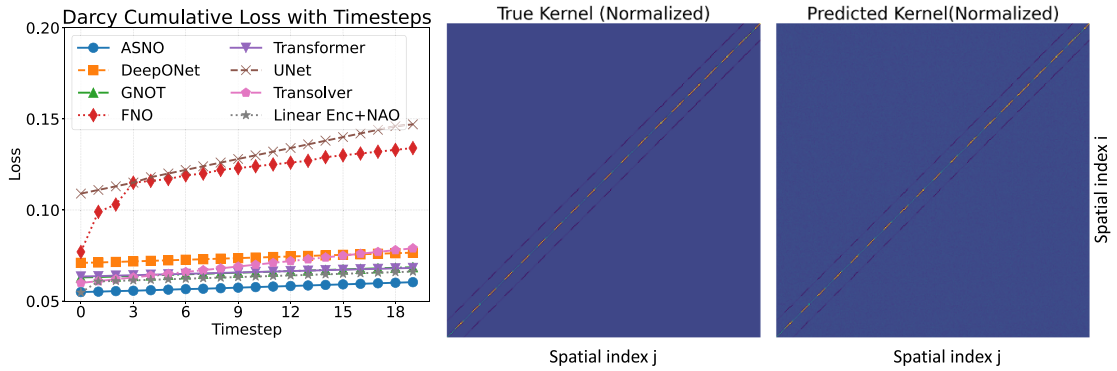

Figure 3. Left: Cumulative error over time for various models on the Darcy flow benchmark, illustrating ASNO's long-term stability. Right: Comparison between the ground-truth Darcy kernel and the spatial kernel learned by the nonlocal attention operator (NAO), demonstrating alignment with the underlying physics.

uniform timesteps for consistency across benchmarks, the ASNO framework is also capable of learning from non-uniform temporal grids by incorporating continuous-time positional encodings or explicit Δt embeddings into the transformer encoder, enabling it to model temporal dynamics as a function of variable time gaps.

These results show that ASNO balances predictive accuracy, interpretability, and computational cost, and maintains stable performance even under variations in forcing and permeability.

4.2. Chaotic ODE: Lorenz system

In this section, we evaluate the performance of ASNO on the Lorenz system, a well-known nonlinear dynamical model featuring high sensitivity to initial conditions:

$$\frac{\partial x}{\partial t} = \sigma\left(y(t) - x(t)\right) + g_1(t), \tag{27}$$

$$\frac{\partial y}{\partial t} = x(t)\left(\rho - z(t)\right) - y(t) + g_2(t), \tag{28}$$

$$\frac{\partial z}{\partial t} = x(t)y(t) - \beta z(t) + g_3(t). \tag{29}$$

Here, our goal is to learn the evolution operator of the Lorenz chaotic ODE system and accurately forecast the future state vector $X(t)$ given its past temporal history and external forcing parameters. In this setting, $X(t)$ corresponds to the full set of coupled state variables in the Lorenz model, while there is no $F(t)$ The dataset consists of 2000 time-series profiles with 100 different coefficients combinations (σ, ρ, β), 20 different loading functions (g_1, g_2, g_3), and the same initial conditions $(x(0), y(0), z(0)) = (0, 1, 0)$. Each profile contains 1000 timesteps. 80% profiles are used for training and 20% are for testing. To construct input-output pairs, we apply a sliding window of length 5 that moves at every timestep, yielding 996 windows per profile (from $t = 1$ through $t = 996$); we discard the final partial window to obtain 995 samples per profile. To improve model generalization and robustness, we augment the data by permuting the order of entire 5-step windows within each trajectory, rather than shuffling individual timesteps, which preserves local temporal structure while increasing data diversity, and by randomly permuting the profile indices to avoid overlap between train and test sets. This results in a total of $80 \times 995 \times 20 = 1592\,000$ training samples and $20 \times 995 \times 20 = 398\,000$ testing samples without leakage. For reproducibility, in our Lorenz system experiments the transformer encoder is configured with embedding dimension $d_{\text{embed}} = 20$, number of attention heads $n_{\text{heads}} = 2$, number of layers $n_{\text{layers}} = 3$, and dropout probability $p = 0.1$. The NAO is instantiated with $n_{\text{blocks}} = 3$, a single attention head $(r = 1)$, key/query dimension $d_k = 10$, and projection matrices mapping to the 3-dimensional state. We train on 90 trajectories (of length 1000) with 20 random permutations each (total batch size 100), using the Adam optimizer with initial learning rate $\text{lr} = 10^{-2}$, weight decay $\text{wd} = 10^{-5}$, and a StepLR scheduler (decay factor $\gamma = 0.7$ every 100 steps). Training runs for 10 000 epochs with fixed random seed 0 to ensure full reproducibility. ASNO is compared against five baselines (Transolver, DeepONet, transformer encoder, GNOT, and Linear Model + NAO), all trained under identical input formats: each model receives the five previous states $\{X_{m-4}, \ldots, X_m\}$ as well as the current forcing F_m. We did not include architectures such as U-Net, FNO, and GNOT in our Lorenz comparison due to architectural

Table 2. Comparison of ASNO and baselines models on the Lorenz system.

Model type	Trainable params	GPU (MB)	Best Test Loss
ASNO	258 808	76	**0.000 794**
Transolver	395 907	95	0.000 835
DeepONet	265 512	79	0.001 750
transformer encoder	**257 647**	**71**	0.001 821
GNOT	401 155	106	0.002 189
Linear Model + NAO	305 776	87	0.005 298

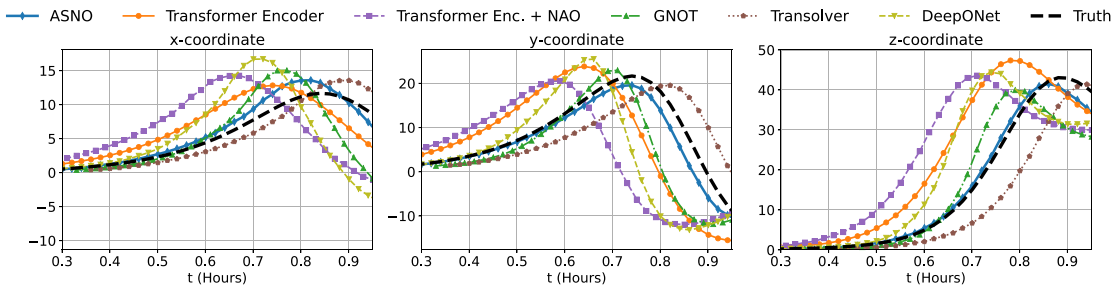

Figure 4. Long Time prediction comparison of Predicted vs. true time-series trajectories (x, y, z) for models listed in table 2 on the Lorenz system.

mismatch: U-Net is designed for high-dimensional spatial grids and relies on convolutional down-/up-sampling; applying it to a low-dimensional ODE time series forces unnatural reshaping and often leads to over-smoothing of the rapid state changes characteristic of chaos. FNO similarly assumes a spatial Fourier basis-its global spectral filters excel for PDEs on regular meshes but provide little benefit (and much overhead) when modeling three-dimensional trajectories in (x, y, z). GNOT builds in geometric gating layers that require constructing a graph or manifold structure even for these 3-variable ODEs, adding implementation complexity and extra hyperparameters with little payoff. The results in table 2 show that ASNO achieves the lowest test loss (0.000 794), demonstrating its predictive accuracy on this chaotic system.

By emulating an implicit-explicit BDF scheme, ASNO first employs a Transformer encoder to extrapolate the homogeneous (linear) dynamics from the past n states, thereby capturing dominant temporal modes and the system's exponential divergence rate without interference from nonlinear feedback. The Nonlocal Attention Operator then applies a compact learned kernel to introduce the corrective coupling terms (e.g. the products xy, xz), focusing exclusively on rapid cross-variable interactions. This two-stage decomposition lightens the representational load on each module: the transformer encoder solves a multistep linear recurrence, while the NAO specializes in a small nonlinear correction. As a result, ASNO achieves more stable long-term rollouts in the chaotic regime-small errors in one stage do not immediately amplify in the other-and empirical trajectories adhere much more closely to the true Lorenz attractor, especially in the sensitive y and z directions, than do those from monolithic models. For long-term rollouts, each new prediction is generated by feeding ASNO's previous output back into the model as the next input, mirroring an autoregressive integration scheme. Figure 4 shows ASNO's predictions remaining near the ground truth over many steps, whereas the other baselines exhibit deviation as of earlier timesteps. By cleanly separating temporal forecasting from coupling correction, ASNO is able to achieve improved long-term stability and robustness-properties that are particularly desirable for chaotic forecasting tasks in climate science, physics simulations, and real-time control.

4.3. Navier–Stokes benchmark for PDE testing

To assess ASNO's efficacy on complex PDEs, we evaluate it on the two-dimensional (2D) incompressible Navier–Stokes (NVS) equations in vorticity form.

In this context, our goal is to learn the solution operator of the 2D incompressible Navier–Stokes equations and predict the velocity field $u(t, x)$ given the external forcing field f. Here, X represents the velocity field at time t and $F(t)$ denotes the spatio-temporal forcing field driving the flow. The governing equation is given by

$$\frac{\partial \omega}{\partial t} + \mathbf{u} \cdot \nabla \omega = \nu \Delta \omega + f, \quad \text{with} \quad \Delta \psi = -\omega, \quad \mathbf{u} = \left(\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x} \right),$$

Table 3. Performance of models on Navier–Stokes equations: model complexity, GPU memory, and best test loss.

Model	Trainable params (M)	GPU (MB)	Best test loss
ASNO	4.66	880	**0.0213**
Transolver	4.14	911	0.0234
GNOT	5.25	1024	0.0322
DeepONet	5.10	3100	0.0921
Trans. Enc.	5.19	961	0.0967
FNO	4.10	846	0.1186
U-Net	5.02	991	0.1940
Linear Encoder + NAO	**4.05**	**791**	0.0328

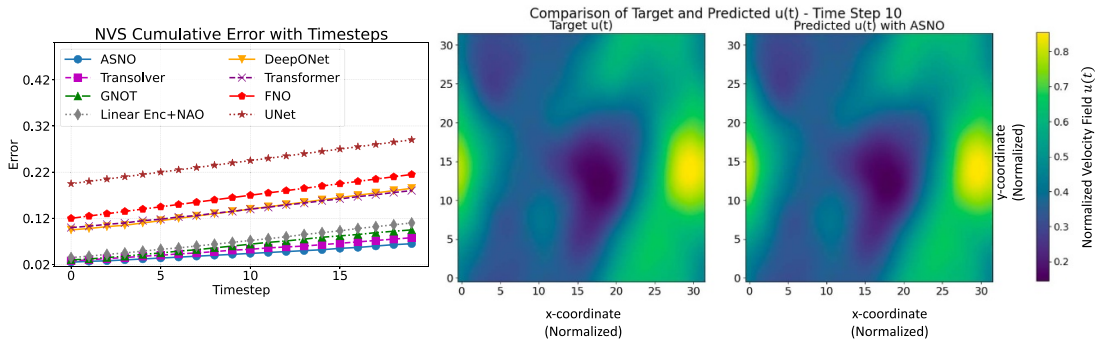

Figure 5. Left: Cumulative error comparison across models listed in table 3 on the Navier–Stokes benchmark. ASNO demonstrates minimal error growth over time, indicating long-term stability. Right: Visualization of predicted versus ground-truth velocity fields, showcasing ASNO's ability to recover fine-scale flow structures.

where ω is the scalar vorticity in 2D, $\mathbf{u} = (u, v)$ is the velocity field, ψ is the streamfunction, ν is the viscosity, and f is the external forcing function. This benchmark tests ASNO's ability to capture nonlinear interactions and multi-field dependencies, following the setup in Li *et al* (2020a). The dataset consists of 100 simulated time-series profiles, each with a spatial resolution of 30×30, governed by varying viscosity coefficients $\nu \in [10^{-4}, 1]$, and subject to a forcing term-driven velocity field. Each profile contains 100 temporal snapshots with a time step size of $\Delta t = 10^{-2}$. To construct training and testing data, a sliding window of length 5 and stride 1 is used, resulting in 96 samples per profile. Additionally, we apply 20 random permutations per trajectory to increase diversity and simulate realistic variations in flow dynamics. This yields $80 \times 96 \times 20 = 153\,600$ training samples and $20 \times 96 \times 20 = 38\,400$ testing samples. These settings create a comprehensive dataset for evaluating ASNO's generalization and stability in modeling high-dimensional, nonlinear fluid systems. For reproducibility, in our Navier–Stokes experiments the transformer encoder is configured with embedding dimension $d_{\text{embed}} = 100$, number of attention heads $n_{\text{heads}} = 50$, number of layers $n_{\text{layers}} = 3$, and dropout probability $p = 0.1$. The NAO is instantiated with $n_{\text{blocks}} = 3$ iterative attention layers, each using a single head ($r = 1$) with key/query dimension $d_k = 50$, and projection matrices $W_{P,h}, W_{P,f}$ mapping to the 30×30 output grid ($S^2 = 900$). We train on $n_{\text{train}} = 900$ trajectories (each of length 100 timesteps) on a 30×30 mesh, applying $n_{\text{perm}} = 20$ random permutations per trajectory (batch size $= 100$), with a 90/10 train/test split. Optimization uses the Adam optimizer with initial learning rate $\text{lr} = 3 \times 10^{-3}$, weight decay $\text{wd} = 3 \times 10^{-3}$, and a StepLR scheduler (decay factor $\gamma = 0.7$ every 100 steps). Training runs for 100 epochs with fixed random seed 0.

Table 3 compares ASNO against state-of-the-art models in test accuracy, efficiency, and parameter count. ASNO achieves the lower test loss while maintaining computational efficiency. Figure 5 illustrates cumulative error trends, demonstrating ASNO's stability over long time horizons. These results confirm ASNO's effectiveness in handling high-dimensional, nonlinear PDEs, generalizing across diverse physical systems with minimal numerical dissipation. Crucially, ASNO's separable architecture-where the transformer encoder approximates the homogeneous (viscous and pressure) evolution through a high-order multistep forecast and the Nonlocal Attention Operator captures the advective and coupling interactions spatially-reduces the representational burden on each component. By isolating the stiff linear dynamics

from the convective nonlinearities, ASNO attains both greater numerical stability and sharper resolution of fine-scale flow structures than monolithic models. Building on ASNO's success in modeling high-dimensional, nonlinear PDEs like the Navier–Stokes equations, we extend its application to real-world engineering challenges.

4.4. Melt pool temperature field prediction in additive manufacturing (AM)

In this section we present a case study on DED AM processes where the goal is to predict the evolution of the melt pool temperature field (Gunasegaram *et al* 2024). In this application-driven case study, our objective is to learn the thermal field evolution operator in a DED process, enabling accurate prediction of spatio-temporal temperature distributions from limited past thermal measurements and laser input parameters. Here, $X(t)$ is the spatial temperature field at time t and $F(t)$ encodes the laser power profile and scan path information over time. Accurate melt pool prediction is crucial in real-world manufacturing scenarios as small deviations in melt pool temperature can significantly affect part quality. First, to collect data to train ASNO we employ GAMMA, an in-house developed GPU-accelerated Finite Element Analysis (FEA) code, for part-scale simulations (Liao *et al* 2023). The GAMMA simulation follows the transient heat conduction equation, incorporating the essential partial differential equations (PDEs) that underpin the DED process:

$$\rho C_p(T)\frac{\partial T}{\partial t} + \nabla \cdot q = 0, \tag{30}$$

Where T is the temperature (K), ρ and C_p are the density ($\mathrm{g\,mm^{-3}}$) and the effective specific heat capacity ($\mathrm{J\,g^{-1}\,K^{-1}}$) of the material, respectively. The heat flux q is given by Fourier's law:

$$q = -k\nabla T, \tag{31}$$

where k is the thermal conductivity of the material. The boundary conditions in the DED process can be formulated as:

$$q \cdot n = \frac{-2\eta P}{\pi r_{\mathrm{beam}}^2}\exp\left(\frac{-2d^2}{r_{\mathrm{beam}}^2}\right) + \sigma\epsilon\left(T^4 - T_0^4\right) + h\left(T - T_0\right), \tag{32}$$

where η is the absorption coefficient (%), P is the laser power (W), r_{beam} is the beam radius (mm), and d is the distance (mm) from the material point to the center of the laser. h is the convection heat transfer coefficient ($\mathrm{W\,m^2\,K^{-1}}$), σ is the Stefan–Boltzmann constant ($5.67 \times 10^{-8}\,\mathrm{W\,m^2\,K^{-4}}$), ϵ is the material's emissivity, and T_0 is the ambient temperature. In the simulation, elements are activated and incorporated into the mesh when the distance between the element's center and the beam center is less than the beam's size.

To generate training data, we first generated 100 different laser power profiles with various combinations of laser power $P_{\mathrm{laser}}(m)$ and scanning rate $V_{\mathrm{scan}}(m)$ as process parameters, and then simulate the printing process of a thin wall using the GAMMA simulation. The temperature field $T_{\mathrm{pool}}(m)$ corresponding to the laser location $L_{\mathrm{laser}}(m)$ at each time step m was saved as the training output. Each temperature field is recorded on a 21×21 spatial grid, and each profile consists of 730 time steps. To construct the training data, a sliding window of length 5 and stride 1 was used, resulting in $730 - 5 + 1 = 726$ samples per profile. A random permutation of 20 per trajectory was applied to enhance data diversity while preserving temporal locality, yielding a total of $80 \times 726 \times 20 = 1161600$ training samples and $20 \times 726 \times 20 = 290400$ testing samples (based on an 80:20 split). With training data collected, the ASNO for predicting the melt pool temperature field can be trained by formulating the prediction based on the processing parameters and the history from the past five steps:

$$\begin{aligned}\hat{T}_{\mathrm{pool}}(m+1) = \mathrm{ASNO}\big(&P_{\mathrm{laser}}(m+1), V_{\mathrm{scan}}(m+1),\\ &L_{\mathrm{laser}}(m+1), T_{\mathrm{pool}}(m-4:m)\big).\end{aligned} \tag{33}$$

The predicted melt pool temperature field from a selected frame is shown in figure 6, compared with the ground truth result from the GAMMA simulation. For the AM DED case study, we configure the transformer encoder with embedding dimension $d_{\mathrm{embed}} = 100$, number of attention heads $n_{\mathrm{heads}} = 50$, number of layers $n_{\mathrm{layers}} = 3$, and dropout probability $p = 0.1$. The NAO uses $r = 1$ attention head per block, $n_{\mathrm{blocks}} = 2$, and key/query dimension $d_k = 20$, with projection matrices $W_{P,h}, W_{P,f} \in \mathbb{R}^{20\times 900}$ to map onto the 30×30 output grid. We train on $n_{\mathrm{train}} = 41$ trajectories (each with $n_{\mathrm{timesteps}} = 730$) sampled on a 30×30 mesh, applying $n_{\mathrm{perm}} = 20$ random permutations per trajectory (batch size $= 100$), using the Adam optimizer (initial learning rate $\mathrm{lr} = 3 \times 10^{-3}$, weight decay $\mathrm{wd} = 1 \times 10^{-2}$), a StepLR scheduler

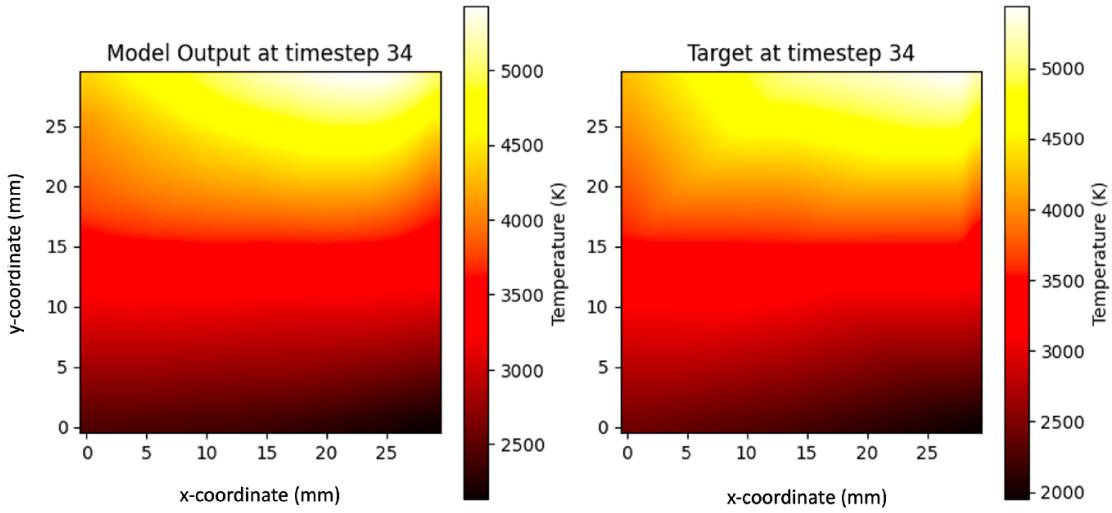

Figure 6. Prediction results on the DED AM case study with the ASNO prediction on the left, and target results on the right.

(decay factor $\gamma = 0.5$ every 100 steps), and minimize a custom L_p loss for 1000 epochs with random seed 0. As a result, the ASNO achieves a mean absolute percentage error of 2.50%. Additional prediction results of different process input scenarios for AM are provided in appendix.

5. Conclusion

In this paper, we introduce ASNO, a spatio-temporal framework designed to enhance generalizability and interpretability in modeling time-dependent differential equations. ASNO leverages a separable kernel approach IMEX BDF, enabling to disentangle temporal and spatial interactions to provide insights on system dynamics. The transformer encoder estimates homogeneous system solutions from historical states, analogous to the explicit step in BDF, while a neural operator models spatial interactions and external load impacts. To ensure robust generalization across diverse physical systems, ASNO integrates the NAO, which facilitates adaptation to varying PDE parameters such as initial conditions, loadings, and environments. Evaluations on benchmarks including the Lorenz system, Darcy flow equation, Navier Stokes, and the DED application demonstrate that ASNO outpertforms over the existing models in terms of accuracy, long-term stability, and zero-shot generalizability to unseen physical parameters. In addition, ASNO offers interpretability by isolating the influence of temporal and spatial dynamics, revealing how historical states and spatial loadings contribute to governing behaviors, and aligning predictions with underlying physical laws. This dual emphasis on generalizability and interpretability makes ASNO suitable for real-time decision-making in high-stakes environments, while its attention mechanisms uncover meaningful patterns for physics-informed discovery and decision-making. Nevertheless, like any data-driven framework, ASNO has limitations, particularly in low-data regimes where insufficient temporal or spatial coverage could impact generalization quality. Another promising but unexplored direction is the adaptation of ASNO to inverse problem settings, where system parameters are inferred alongside state evolution, which may require additional architectural and optimization considerations. Future research can extend ASNO's applications to foundational modeling for transfer learning across different physical systems, further advancing its role in uncovering and predicting complex physical phenomena in scientific and industrial domains.

Data availability statement

The data that support the findings of this study will be openly available following an embargo at the following URL/DOI: https://github.com/Vispikarkaria/ASNO.

Acknowledgments

We appreciate the grant support from the NSF HAMMER-ERC (Engineering Research Center for Hybrid Autonomous Manufacturing, Moving from Evolution to Revolution) under Award Number EEC-2133630, and the NSF MADE-PUBLIC Future Manufacturing Research Grant Program under Award

Number CMMI-2037026. Wei Chen and Doksoo Lee acknowledge support from the NSF Boosting Research Ideas for Transformative and Equitable Advances in Engineering (BRITE) Fellow Program (CMMI 2227641). Yi-Ping Chen appreciates the Taiwan-Northwestern Doctoral Scholarship funded by the Ministry of Education in Taiwan, and the fellowship support from the Predictive Science and Engineering Design (PSED) cluster at Northwestern University. Yue Yu was funded by the National Institute of Health under Award 1R01GM157589-01 and the AFOSR under Grant FA9550-22-1-0197. Portions of this research were conducted on Lehigh University's Research Computing infrastructure, partially supported by NSF Award 2019035.

Appendix. Melt pool temperature field prediction in AM

In this section, we provide a comprehensive analysis of the melt pool characteristics predicted by ASNO and other comparative models. The melt pool in DED is a key factor in determining material properties. Accurate modeling of the melt pool helps assess temperature distribution, phase transformation, and thermal cycles, which directly affect the microstructure and mechanical performance of the final product. Additionally, understanding melt pool dynamics is essential for process control, defect mitigation, and ensuring part consistency in DED applications. By comparing ASNO's predictive performance with other models, we aim to demonstrate how well it captures these critical characteristics in a manufacturing setting.

The simulations were conducted over a 21×21 spatial grid, with temperature distributions captured at multiple timesteps throughout the deposition process. In each timestep, the model outputs a 2D temperature field that reflects the temperature variations across the substrate due to the moving heat source.

Figure 7 illustrates these results, showing a side-by-side comparison between ASNO's model predictions and the target values across three distinct timesteps (10, 25 and 36). The color gradient, representing temperature from 2000 K to 5500 K, demonstrates the intensity of heat within the melt pool and the cooling pattern radiating outward from the laser's path. In each row, the predicted temperature distribution is shown on the left, directly alongside the corresponding target temperature distribution on the right, offering a clear visual comparison of ASNO's accuracy at various time steps. The ASNO results show good agreement with the target profiles.

These visualizations underscore ASNO's ability to model high-temperature zones and predict the progression of thermal profiles over time accurately. Notably, the model predictions retain stability over longer timesteps, capturing the complexity of the transient heat flow in DED processes. The agreement between the predicted and target distributions suggests that ASNO is well-suited for high-fidelity simulations in AM applications, where precise thermal predictions are crucial for quality control and optimization.

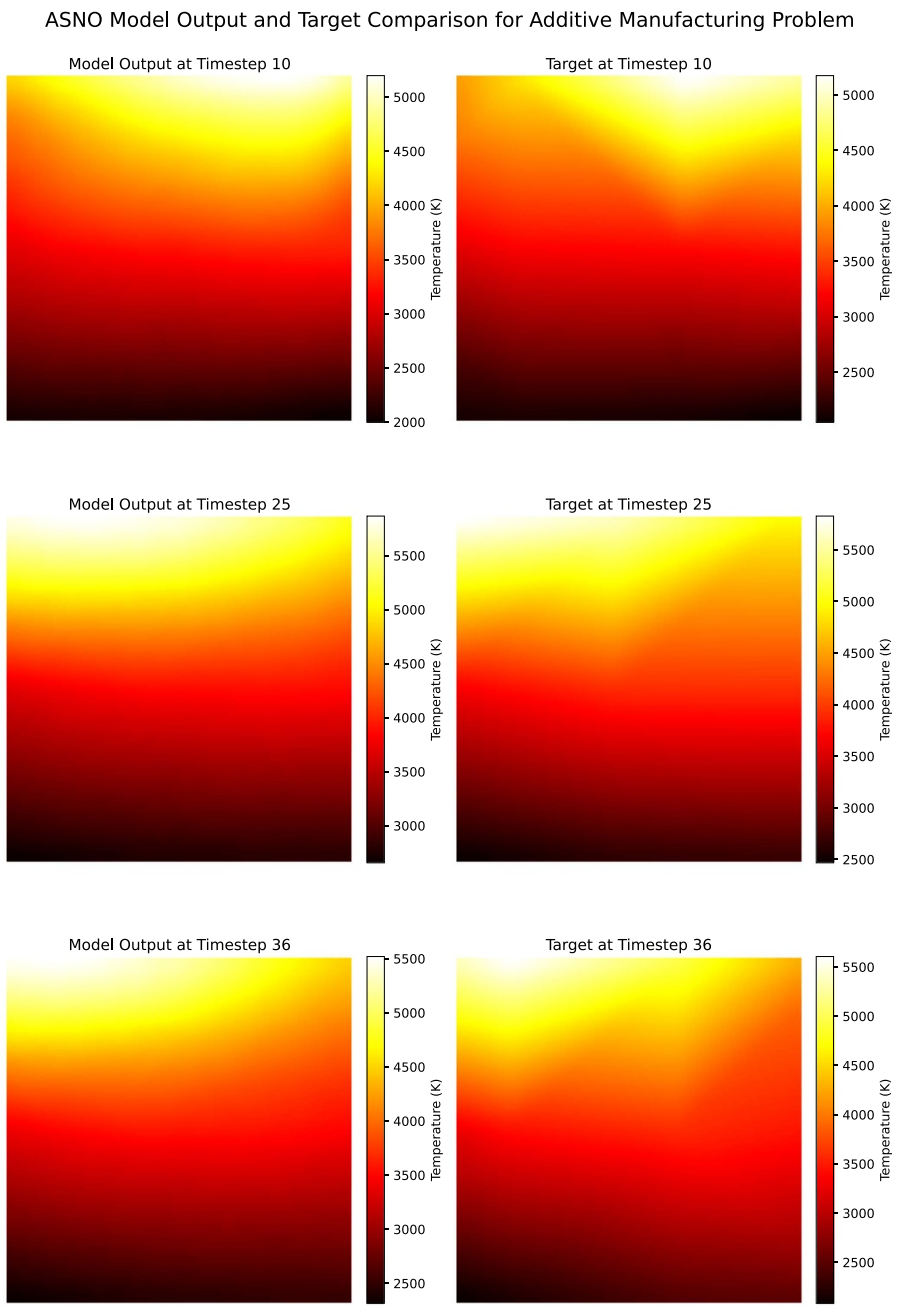

Figure 7. Comparison of ASNO model output and target temperature distributions for the melt pool in directed energy deposition at timesteps 10, 25, and 36. Each row represents a different timestep, with the model output shown on the left and the target on the right. The color scale indicates temperature in Kelvin, highlighting ASNO's accuracy in capturing the thermal profile of the melt pool.

ORCID iDs

Vispi Karkaria 0000-0003-3733-2415
Doksoo Lee 0000-0002-4597-6643
Yue Yu 0000-0002-9150-3986

References

Ascher U M, Ruuth S J and Wetton B T 1995 Implicit-explicit methods for time-dependent partial differential equations *SIAM J. Numer. Anal.* **32** 797–823

Azizzadenesheli K, Kovachki N, Li Z, Liu-Schiaffini M, Kossaifi J and Anandkumar A 2024 Neural operators for accelerating scientific simulations and design *Nat. Rev. Phys.* **6** 1–9

Cao Q, Goswami S and Karniadakis G E 2024 Laplace neural operator for solving differential equations *Nat. Mach. Intell.* **6** 631–40

Cao S 2021 Choose a transformer: Fourier or galerkin *Advances in Neural Information Processing Systems* vol 34 pp 24924–40
Chen Y P, Karkaria V, Tsai Y K, Rolark F, Quispe D, Gao R X, Cao J and Chen W 2025 Real-time decision-making for digital twin in additive manufacturing with model predictive control using time-series deep neural networks (arXiv:2501.07601)
Cheng C W, Huang J, Zhang Y, Yang G, Schönlieb C B and Aviles-Rivero A I 2024 Mamba neural operator: who wins? transformers vs. state-space models for PDES (arXiv:2410.02113)
Fournier Q, Caron G M and Aloise D 2023 A practical survey on faster and lighter transformers *ACM Comput. Surv.* **55** 1–40
Fredebeul C 1998 A-bdf: a generalization of the backward differentiation formulae *SIAM J. Numer. Anal.* **35** 1917–38
Graves A and Graves A 2012 Long short-term memory *Supervised Sequence Labelling With Recurrent Neural Networks* pp 37–45
Gunasegaram D, Barnard A, Matthews M, Jared B, Andreaco A, Bartsch K and Murphy A 2024 Machine learning-assisted in-situ adaptive strategies for the control of defects and anomalies in metal additive manufacturing *Add. Manuf.* **81** 104013
Hao Z, Wang Z, Su H, Ying C, Dong Y, Liu S, Cheng Z, Song J and Zhu J 2023 GNOT: a general neural operator transformer for operator learning *Int. Conf. on Machine Learning* (PMLR) pp 12556–69
Hu J and Shu R 2021 On the uniform accuracy of implicit-explicit backward differentiation formulas (imex-bdf) for stiff hyperbolic relaxation systems and kinetic equations *Math. Comput.* **90** 641–70
Kapteyn M G, Pretorius J V and Willcox K E 2021 A probabilistic graphical model foundation for enabling predictive digital twins at scale *Nat. Comput. Sci.* **1** 337–47
Karkaria V, Goeckner A, Zha R, Chen J, Zhang J, Zhu Q, Cao J, Gao R X and Chen W 2024a Towards a digital twin framework in additive manufacturing: machine learning and bayesian optimization for time series process optimization *J. Manuf. Syst.* **75** 322–32
Karkaria V, Tsai Y-K, Chen Y-P and Chen W 2024b An optimization-centric review on integrating artificial intelligence and digital twin technologies in manufacturing *Eng. Optim.* **57** 1–47
Kim D and Lee J 2024 A review of physics informed neural networks for multiscale analysis and inverse problems *Multiscale Sci. Eng.* **6** 1–11
Kitaev N, Kaiser Ł and Levskaya A 2020 Reformer: the efficient transformer (arXiv:2001.04451)
Ko H, Kim J, Lu Y, Shin D, Yang Z and Oh Y 2022 Spatial-temporal modeling using deep learning for real-time monitoring of additive manufacturing *Int. Design Engineering Technical Conf. and Computers and Information in Engineering Conf.* vol 86212 (American Society of Mechanical Engineers) p V002T02A019
Kovachki N, Li Z, Liu B, Azizzadenesheli K, Bhattacharya K, Stuart A and Anandkumar A 2023 Neural operator: learning maps between function spaces with applications to PDES *J. Mach.: Learn. Res.* **24** 1–97
Li Z, Kovachki N, Azizzadenesheli K, Liu B, Bhattacharya K, Stuart A and Anandkumar A 2020a Fourier neural operator for parametric partial differential equations (arXiv:2010.08895)
Li Z, Kovachki N, Azizzadenesheli K, Liu B, Stuart A, Bhattacharya K and Anandkumar A 2020b Multipole graph neural operator for parametric partial differential equations *Advances in Neural Information Processing Systems* vol 33 pp 6755–66
Liao S, Golgoon A, Mozaffar M and Cao J 2023 Efficient gpu-accelerated thermomechanical solver for residual stress prediction in additive manufacturing *Comput. Mech.* **71** 879–93
Lim B, Arık S O, Loeff N and Pfister T 2021 Temporal fusion transformers for interpretable multi-horizon time series forecasting *Int. J. Forecast.* **37** 1748–64
Liu X, Xu B and Zhang L 2022 HT-NET: hierarchical transformer based operator learning model for multiscale PDES
Lu F and Yu Y 2025 Transformer learns the cross-task prior and regularization for in-context learning (arXiv:2505.12138)
Lu L, Jin P and Karniadakis G E 2019 Deeponet: learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators (arXiv:1910.03193)
Lu L, Jin P, Pang G, Zhang Z and Karniadakis G E 2021 Learning nonlinear operators via deeponet based on the universal approximation theorem of operators *Nat. Mach. Intell.* **3** 218–29
Mandl L, Goswami S, Lambers L and Ricken T 2024 Separable deeponet: breaking the curse of dimensionality in physics-informed machine learning (arXiv:2407.15887)
Molinaro R, Yang Y, Engquist B and Mishra S 2023 Neural inverse operators for solving PDE inverse problems (arXiv:2301.11167)
Molnar C 2020 *Interpretable Machine Learning* (Lulu.com)
Nelson B K 1998 Time series analysis using autoregressive integrated moving average (arima) models *Acad. Emer. Med.* **5** 739–44
Niu Z, Zhong G and Yu H 2021 A review on the attention mechanism of deep learning *Neurocomputing* **452** 48–62
Rafiq M, Rafiq G, Jung H-Y and Choi G S 2022 SSNO: spatio-spectral neural operator for functional space learning of partial differential equations *IEEE Access* **10** 15084–95
Rudin C, Chen C, Chen Z, Huang H, Semenova L and Zhong C 2022 Interpretable machine learning: fundamental principles and 10 grand challenges *Stat. Surv.* **16** 1–85
Tang B and Matteson D S 2021 Probabilistic transformer for time series analysis *Advances in Neural Information Processing Systems* vol 34 pp 23592–608
Thelen A, Zhang X, Fink O, Lu Y, Ghosh S, Youn B D, Todd M D, Mahadevan S, Hu C and Hu Z 2022 A comprehensive review of digital twin-part 1: modeling and twinning enabling technologies *Struct. Multidiscip. Optim.* **65** 354
van Beek A, Nevile Karkaria V and Chen W 2023 Digital twins for the designs of systems: a perspective *Struct. Multidiscip. Optim.* **66** 49
Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L and Polosukhin I 2017 Attention is all you need *Advances in Neural Information Processing Systems* p 30
Wang Q *et al* 2024 P^2c^2net: PDE-preserved coarse correction network for efficient prediction of spatiotemporal dynamics (arXiv:2411.00040)
Wen G, Li Z, Azizzadenesheli K, Anandkumar A and Benson S M 2022 U-fno-an enhanced fourier neural operator-based deep-learning model for multiphase flow *Adv. Water Resour.* **163** 104180
Wu H, Luo H, Wang H, Wang J and Long M 2024 Transolver: a fast transformer solver for PDES on general geometries (arXiv:2402.02366)
You H, Zhang Q, Ross C J, Lee C-H and Yu Y 2022 Learning deep implicit fourier neural operators (IFNOs) with applications to heterogeneous material modeling *Comput. Methods Appl. Mech. Eng.* **398** 115296
Yu Y, Liu N, Lu F, Gao T, Jafarzadeh S and Silling S 2024 Nonlocal attention operator: materializing hidden knowledge towards interpretable physics discovery (arXiv:2408.07307)
Zerveas G, Jayaraman S, Patel D, Bhamidipaty A and Eickhoff C 2021 A transformer-based framework for multivariate time series representation learning *Proc. 27th ACM SIGKDD Conf. on Knowledge Discovery & Data Mining* pp 2114–24

Zhang Y, Zhang Y, Zhang Z, Bao J and Song Y 2018 Human activity recognition based on time series analysis using u-net (arXiv:1809.08113)

Zhao T, Fang L, Ma X, Li X and Zhang C 2024 Tfformer: a time-frequency domain bidirectional sequence-level attention based transformer for interpretable long-term sequence forecasting *Pattern Recognit.* **158** 110994

Zhou H, Zhang S, Peng J, Zhang S, Li J, Xiong H and Zhang W 2021 Informer: beyond efficient transformer for long sequence time-series forecasting *Proc. AAAI Conf. on Artificial Intelligence* vol 35 pp 11106–15