CTF4NUCLEAR: A COMMON TASK FRAMEWORK FOR SCIENTIFIC MACHINE LEARNING IN NUCLEAR SYSTEMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Designing, monitoring, and operating nuclear systems is exceptionally challenging because of the complex physical phenomena and their interactions that govern the system dynamics. While high-fidelity simulations can resolve these non-linear, multi-physics interactions (e.g., between neutronics and thermal-hydraulics), they are computationally expensive and rarely suitable for multi-query and real-time applications. Furthermore, model-based approaches inherently rely on simplifying assumptions, leading to inevitable discrepancies with real-world measurements. Machine Learning (ML) surrogates offer a promising alternative; however, systematic comparisons of their performance across relevant operating conditions and data constraints remain limited. In safety-critical settings such as nuclear engineering, a fair and rigorous comparison of different ML methods, and a clear understanding of their advantages and limitations, is of paramount importance. To address this gap, we introduce a Common Task Framework (CTF) for the application of ML in nuclear engineering, building upon previous efforts in dynamical systems and seismology. This CTF considers a curated set of datasets spanning different nuclear and nuclear-adjacent systems, including molten salt, micro-reactors, electrically conducting fusion coolants, and experimental thermal-hydraulics facilities. To demonstrate the rigour and fairness of the CTF, we apply it to the one of the most strongly coupled system, the molten salt reactor, evaluating the performance of all methods on twelve established metrics (forecasting, noise robustness, limited data, and parametric generalisation), and identifying substantial differences in their robustness and generalisation. Additionally, we introduce a sparse-observation monitoring paradigm to evaluate whether models can reconstruct and forecast system dynamics from limited sensor measurements, providing a common setting for comparing forecasting and state-estimation methods. Overall, the proposed CTF provides a reproducible basis for systematic comparison of data-driven dynamical models in nuclear engineering and establishes a framework for assessing ML capability under realistic data and observation constraints in complex, safety-critical physical systems.

1 INTRODUCTION

Nuclear fission and fusion reactors are among the most safety-critical dynamical systems ever engineered, requiring continuous real-time knowledge of the full physical state of the reactor core (neutron flux, temperature, coolant velocity, and delayed-neutron precursor concentrations) (Duderstadt & Hamilton, 1976). These requirements are becoming more stringent as nuclear plants are expected to operate in load-following mode alongside variable renewable sources (IAEA, 2023; Ruth et al., 2014), and next-generation digital twins (Grieves, 2019; Mohanty & Vilim, 2021; Brown, 2022) must now provide accurate real-time state estimates for control, monitoring, and accident prognosis. These requirements imply adequate instrumentation to provide the needed data: however, nuclear cores are extreme environments, with in-core instrumentation becoming almost physically impossible in Generation-IV or fusion concepts, forcing sensors to the core boundary or external shielding structures (Cammi et al., 2024; Riva et al., 2024). Recovering the full multi-physics state from these sparse, out-of-core observations is an inherently ill-conditioned inverse problem (Argaud et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

2018) that involves fields that the sensors never directly measure. Conversely, the governing Partial Differential Equations (PDE) of a nuclear fission reactor couple, at minimum, the neutron Boltzmann transport equation with the Navier-Stokes system (Duderstadt & Hamilton, 1976; Demazière, 2020) and define Full Order Models (FOMs) whose runtime spans hours to days, preventing many-query and real-time use (Kapteyn et al., 2020).

As a result of the complexity of nuclear FOMs, the nuclear engineering community is developing a growing number of surrogate and data-driven modelling approaches. Reduced Order Modelling (ROM) based on the Proper Orthogonal Decomposition (POD) (Lassila et al., 2014; Rozza et al., 2020) provides a core compression framework, with projection-based approaches such as GeN-ROM (German et al., 2022) extending it to fully intrusive multi-physics surrogates. Non-intrusive methods that integrate measurement data include the Generalised Empirical Interpolation Method (Maday et al., 2015) and the Parametrized-Background Data-Weak formulation (Maday et al., 2014), which were applied to nuclear state estimation, sensor placement, and multi-physics bias correction (Gong et al., 2022a; Cammi et al., 2024; Gong et al., 2023). More recently, deep learning methods have entered the nuclear domain. Physics-Informed Neural Networks (Raissi et al., 2019) have been applied to point-kinetics transients (Prantikos et al., 2022; 2023), neutron diffusion (Wang et al., 2022), and full transport modelling (Geng et al., 2026). Neural operators have emerged as powerful function-space surrogates: Deep Operator Networks (Lu et al., 2021a) have been used for particle transport and thermal-hydraulic monitoring (Kobayashi & Alam, 2024; Hossain et al., 2025), while Fourier Neural Operators (Li et al., 2021a) have demonstrated six-orders-of-magnitude computational acceleration over traditional solvers in predicting plasma dynamics within Tokamak reactors (Gopakumar et al., 2024), making them a natural candidate for fusion applications also included in our benchmark. Sequence-to-state architectures such as Shallow Recurrent Decoders (Williams et al., 2024; Faraji et al., 2025) reconstruct the full core state from as few as three out-of-core sensors, with strong performance on molten salt reactors (Riva et al., 2025; Tomasetto et al., 2025). Broader reviews of the field are available in (Zhang et al., 2026) and (Huang et al., 2023).

Despite this rapid growth, the field faces a reproducibility issue documented across scientific machine learning (McGreivy & Hakim, 2024): methods are evaluated on self-selected datasets, with both training and test data available to developers, creating structural incentives for optimistic reporting. Without a truly withheld test set, guarding against selective re-training until a favourable outcome is obtained is impossible: Weak baselines, inconsistent protocols, and non-reproducible results inevitably follow (Wyder et al., 2026). The nuclear domain amplifies this problem: experimental data from operating reactors are scarce and proprietary; full-scale demonstrators for Generation-IV concepts do not exist; and high-fidelity simulation datasets, the *de-facto* reference, are rarely shared in standardised formats. The OECD/NEA has acknowledged the need through its Task Force on AI/ML for Scientific Computing in Nuclear Engineering (OECD/NEA, 2024), with benchmark exercises on critical heat flux and reactor kinetics; yet these efforts remain institution-specific, lack a community-accessible withheld test set, and do not cover the transient state-estimation and surrogate-modelling tasks most critical for digital twin applications.

1.1 Related Work

Common Task Frameworks have been the primary mechanism by which ML communities establish rigorous, reproducible progress metrics (Donoho, 2017; Goldfeder et al., 2026b). The transformative role of landmark CTFs is well-established: ImageNet (Deng et al., 2009) catalysed deep convolutional networks (Krizhevsky et al., 2012); competitive games enabled AlphaZero (Silver et al., 2018); the Arcade Learning Environment (Bellemare et al., 2013) and OpenAI Gym (Brockman et al., 2016) accelerated reinforcement learning; CASP (Jumper et al., 2021) drove AlphaFold in structural biology; and natural language processing has benefited from code-generation (Chen et al., 2021) and mathematical reasoning benchmarks (Hendrycks et al., 2021; Cobbe et al., 2021); recently, DeepSeek-R1 used these to demonstrate its competitive performance against competitors' LLMs (Guo et al., 2025a).

Following the above, CTFs for scientific ML have recently gained momentum. Wyder et al. (Wyder et al., 2026) proposed a framework for dynamical systems, benchmarking methods on the Kuramoto-Sivashinsky PDE and the Lorenz attractor across twelve task-specific metrics covering forecasting, reconstruction, noise, limited data, and parametric generalisation. Yermakov et al. (Yermakov et al., 2026) extended this to seismological wave-field modelling with three datasets at global, crustal, and local scales. Other scientific benchmarking efforts include PDEBench (Takamoto et al., 2022), The Well (Ohana et al., 2024), and CoDBench (Burark et al., 2024), which provide datasets and

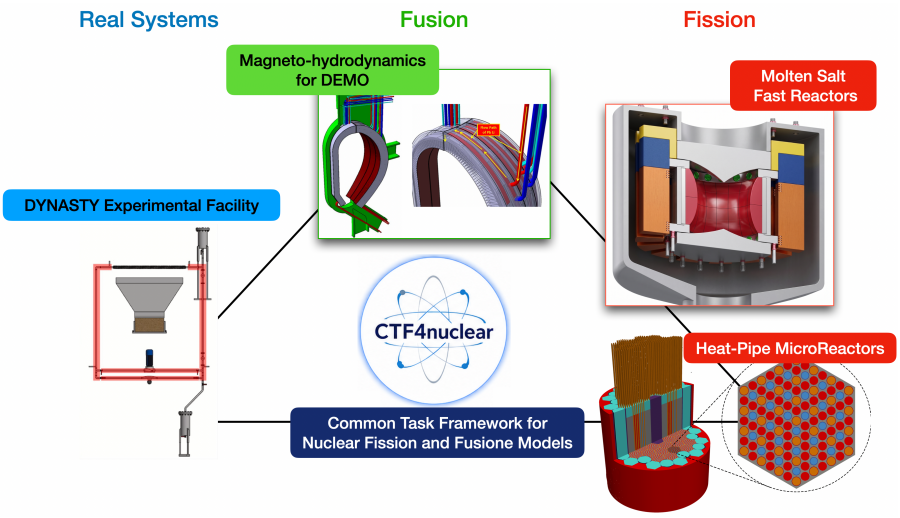


Figure 1: Scheme of the Common Task Framework (CTF) for Nuclear Fission and Fusion Models, with different datasets: Molten Salt Fast Reactor (Brovchenko et al., 2013), Heat-Pipe Micro Reactor (Prince et al., 2024), Magneto-hydrodynamics case studies (Lo Verso et al., 2024) and real experimental facilities like DYNASTY (Pini et al., 2016) and TRIGA (International Atomic Energy Agency, 2016).

evaluation protocols for learning to solve differential equations, but which allow for self-reporting. However, none of these targets nuclear reactor systems (both fission and fusion) whose physics, observability constraints, and safety requirements are sufficiently distinct to warrant a dedicated framework.

1.2 Common Task Framework for Nuclear Reactors

In light of the gap identified in the previous Section, we introduce CTF4Nuclear, the first open common task framework (benchmarking) platform for ML surrogate modelling and state estimation of nuclear reactor systems, built on the openly available `ctf4science` package (Wyder et al., 2026; Yermakov et al., 2026). A distinguishing feature of CTF4Nuclear is the deliberate diversity of its benchmark suite and the path towards higher-dimensional cases. We collect datasets from four physically distinct nuclear engineering systems (see Fig. 1), adopting the main modelling choices to simulate the dynamics of a nuclear system: lumped 0D models composed by a system of ordinary differential equations (Duderstadt & Hamilton, 1976; Introini et al., 2019), system codes focusing on the main flow direction in thermal-hydraulics (Todreas & Kazimi, 1990; Fletcher & Schultz, 1992), and the most recent multi-physics (MP) approach (Demazière, 2020; Aufiero et al., 2014; Fiorina et al., 2015). Within the CTF4Nuclear we include different systems: the Molten Salt Fast Reactor (MSFR) (Brovchenko et al., 2013), a Generation-IV circulating-fuel concept with fully coupled neutronics and thermal-hydraulics and strictly out-of-core sensors, modelled using a MP approach (Aufiero et al., 2014); a Heat-Pipe Micro Reactor (HPMR) (Prince et al., 2024; Wang et al., 2025), a compact solid-fuel concept with a different observability structure, simulated as a 0D system governed by the point kinetics equations; electrically conducting coolants under magneto-hydrodynamic effects (Lo Verso et al., 2024) for fusion energy applications, following the MP approach; and a simulation of a real experimental facility, DYNASTY, a natural-circulation loop (Pini et al., 2016), with a 1D approach (Missaglia et al., 2025). This breadth of case studies is intentional: a method that succeeds on one system (and one modelling choice) but fails on others reveals scope limitations invisible in single-dataset comparisons. Establishing such cross-system conclusions is, by construction, a community effort: in this work we release all four datasets and demonstrate the framework in depth on the most strongly coupled system, the MSFR, while the remaining datasets are released with reference baselines so that the community can populate them.
In addition to the twelve task-specific metrics of (Wyder et al., 2026), covering forecasting and reconstruction under noise, limited data, and parametric variability, CTF4Nuclear introduces a new

task class specific to monitoring engineering system: the reconstruction of physically unobservable fields (such as delayed neutron precursor concentrations or core velocity) from sparse measurements (Argaud et al., 2018; Cammi et al., 2024; Gong et al., 2023). As it will be detailed later, here the difference lies in the available information provided: in particular, ML methods should be able to infer the whole state of the system from local and sparse observations. Although the metrics adopted stay the same, the paradigm of this is essentially different. This novel task has no analogue in existing scientific CTFs, directly targets the central challenge described above, and path the way towards a CTF for Data Assimilation. On a dataset, a composite score is computed as the mean of all task scores to evaluate a ML model; the multi-metric design prevents winner-takes-all outcomes and exposes method profiles (a method robust to noise may perform poorly under parametric shift, for instance), helping practitioners select methods appropriate to their deployment context.
The paper is organized as follows: Section 2 focuses on the description of the datasets and the metrics; Section 3 presents the CTF evaluation of highly-cited ML model to the MSFR dataset; Section 4 focuses on the main limitations and future developments of the work, while in Section 5 the main conclusions are drawn. Finally, the paper is accompanied by some supplementary appendices detailing more deeply the metrics (B) and the ML models adopted (D), as well as some baseline results on the other available datasets (C).

2 Datasets & Evaluation Metrics

The CTF4Nuclear introduces a new set of datasets related to nuclear fission and fusion systems. The datasets are selected to cover a wide range of systems, modelled with different levels of fidelity. Figure 1 shows some pictures of the geometries of the datasets considered; we stress that due to the modularity of the CTF package it allows to include new reactors/systems in a rather simple way.

2.1 Dataset details

Figure 2a shows the data collection procedure when multiple fields are involved[1]: for any field $\psi^{(i)}$, the state at time t_j, with $j = 1, \dots, m$, corresponds to the evaluation of it over the grid points of size n. The full data matrix is then obtained by vertically concatenating the different fields (see Fig. 2b), obtaining a matrix of size $(p \cdot n) \times m$, where p is the number of fields. For interested readers, Section A will provide more details on the mathematical aspects, while here a short summary is reported.

HPMR: The HPMR dataset is generated from transient simulations of a representative Heat Pipe Micro Reactor (Prince et al., 2024), focusing on the neutronic behaviour. In particular, the point kinetics equations (Duderstadt & Hamilton, 1976), with the neutronic parameters computed using Monte Carlo simulations, are used to describe the reactor in a lumped fashion. The reactivity (deviation from critical/steady state) drives the dynamics of the system and it is governed by a Lorenz ODE system. Overall, the state space vector $\mathcal{V}$ is composed by 6 variables: the power of the reactor, two-groups concentration of precursors (accounting for delayed neutron production due to decay) and the three Lorenz variables.

DYNASTY: The DYNASTY dataset (Pini et al., 2016) is generated using a RELAP5[2] thermo-hydraulic one-dimensional model of the experimental facility (Missaglia et al., 2025). DYNASTY was built to study the phenomenology of natural circulation in coupled systems under internal heat generation, akin to the heating and cooling conditions encountered in the MSFR. The heating transient is driven by imposing different heating conditions to the facility, at different heating power and cooling heat exchanger coefficient parametric values. Additionally, physical thermocouples and a mass flow meter installed in the facility itself are used as additional source of information, providing live experimental data. The state space vector consists of temperature T (spatially distributes in 122 nodes) and mass flow rate Γ (single value since the flow is incompressible).

[1] This is a specific feature of MP systems, such as nuclear systems, where the state is generally composed by more than one quantity of interest.

[2] RELAP5 is a validated nuclear code, vastly used by the nuclear community for thermal-hydraulics and accidental scenario simulations in commercial reactors.

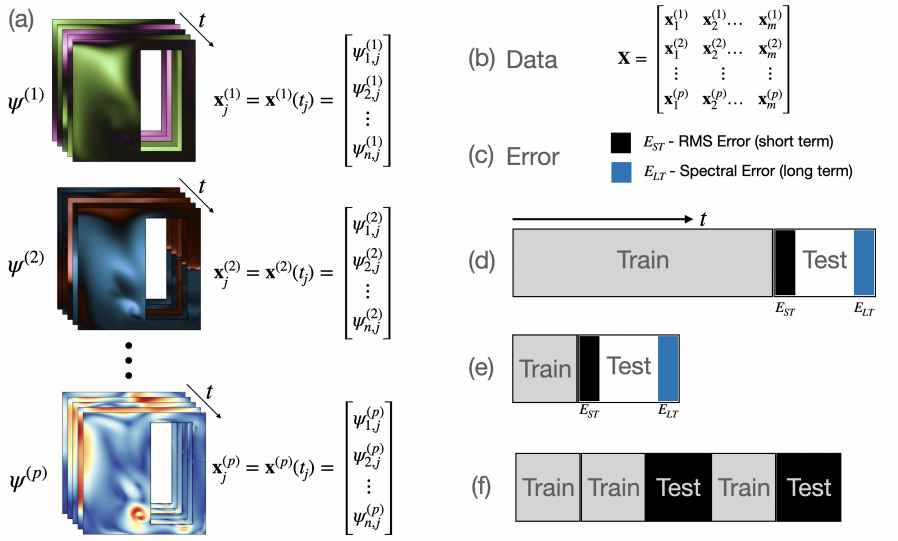


Figure 2: The CTF scores the performance of methods on nuclear reactors data. (a) Data is collected and organized into matrices, which is then split into testing and training sets. (b) Multiple fields $(\psi_1, \psi_2, ...\psi_p)$ are vertically concatenated. (c) RMSE errors are computed for reconstruction and short-time forecasting, while the spectral error computes the statistics of long-time forecasting (spatial or temporal). (d) Forecasting and reconstruction tasks are evaluated on noise-free, low-noise and high-noise data. Methods are also evaluated when (e) Only limited data is available and (f) For reconstruction of parametrically dependent data.

MHD Channel Flow: MHD refers to the coupling between magnetic field and fluid-dynamics that can occur in fusion reactor components. In particular, the MHD datasets focus on lead-lithium blanket flow within a 2D channel (Lo Verso et al., 2024; Scardino et al., 2026) with two cooling cylinders, a basic configuration that nevertheless represents the flow within the breeding blanket of fusion systems: transients are generated by imposing external magnetic fields (varying in time), which impact the dynamics of the lead-lithium flow and promote magneto-convection depending on the overall magnitude of the magnetic field[3]. The coupled system is solved using finite-volume discretisation and a custom coupled solver. The state space vector consists in the lead-lithium temperature T, pressure p and velocity components (u_x, u_y). The size of the state vector is $30600 \cdot 4 = 122400$.

MSFR: The MSFR dataset comes from a multi-physics numerical simulation of a 2D axisymmetric wedge of the EVOL geometry (Brovchenko et al., 2013). The OpenFOAM-based solver couples multi-group neutronic diffusion and incompressible Reynolds-Averaged Navier-Stokes equations (Aufiero et al., 2014), counting more than 20 coupled partial differential equations. In fact, the high-dimensional state space vector $\mathcal{V}$ is composed by more than 20 different fields, of which we extract five of particular interest to fully describe the dynamics of the system, related to neutronics and thermal-hydraulics: $q'''_{\text{prompt}}, q'''_{\text{decay}}, T, u_x, u_z$, being the power density related to prompt and delayed neutrons, the temperature and two velocity components. The mesh is composed by 3880 cells, making the state vector being characterised by 19400 degrees of freedom. Among all possible accidental transients, to minimise predictability complex functions were applied to the momentum source (pump velocity) and to the heat sink (heat transfer coefficient and external temperature), starting from steady-state conditions.

2.2 METRICS

The evaluation metrics of the CTF4NUCLEAR are designed to comprehensively assess the performance of ML models across a range of tasks and conditions, reflecting the real-world challenges

[3]This simulates the realistic scenario of stray magnetic fields coming from the plasma chamber, where the fusion process occurs.

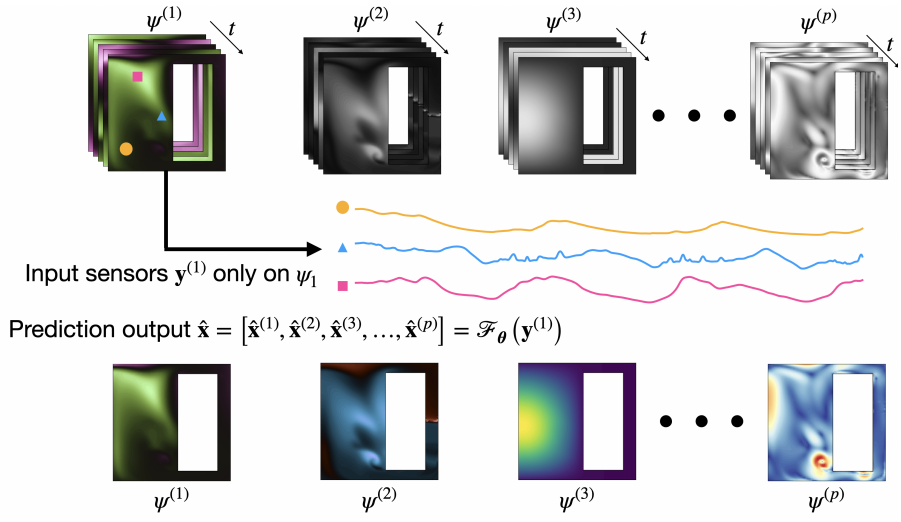


Figure 3: The CTF4NUCLEAR introduces a new task focused on the reconstruction of the state of the system from sparse sensor measurements. In this task, the model is trained to reconstruct the full state of the system (e.g., neutron flux, temperature, velocity) from a limited number of sensor measurements of a single field.

faced in nuclear engineering applications. Following previous works on CTF for scientific ML (Wyder et al., 2026; Yermakov et al., 2026), we propose a standard set of metrics for spatio-temporal data, including forecasting, noisy data, limited data and parametric generalisation (Figure 2). A composite score (*AvgScore*) is then calculated for each dataset by taking the mean of the metrics for each method. Every score is generated by optimizing a single method per task to achieve its highest possible value, using only the released training data for model selection and hyperparameter tuning. The test set remains withheld and is used exclusively for final evaluation, preventing adaptation of the methods to the test data. All scores are bounded to the range $E_i \in [-100, 100]$. For tasks where a method fails to generate a result, the lowest score of -100 is assigned. The 12 scores in the standard set of metrics include different tasks[4]. **Forecasting** (2 scores, E1-E2): evaluate the ability of the model to predict future states of the system (short-term and long-term) based on past observations. **Noisy Data** (4 scores, E3-E6): assess the robustness of the model to noise in the input data, which is common in real-world measurements. This can be done by adding synthetic noise to the test data and evaluating the model's performance. Both reconstruction, a.k.a. de-noising (E3, E5) and forecasting (E4-E6), are considered. **Limited Data** (4 scores, E7-E10): evaluate the model's performance when trained on limited data, with different forecasting horizons and either noisy or clean data. **Parametric Generalisation** (2 scores, E11-E12): assess the model's ability to generalize to unseen parameter regimes, for interpolation (E11) and extrapolation (E12).

Evaluation under Partial Observability Additionally, we introduce a novel paradigm focused on important aspects for the nuclear engineering community, such as the estimation of the state of the system from sparse sensor measurements, a critical aspect for monitoring and control purposes (Argaud et al., 2018; Cammi et al., 2024). The standard set of metrics assumes that the state of the system is fully observable at a certain time t_j to infer the state at a future time t_{j+1}. However, in many real-world applications, we might have only access to sparse and partial measurements, in s points, of the state itself. Thus, we may want to change perspective and evaluate the ability of a model to reconstruct the full state of the system from sparse sensor measurements. More specifically, the paradigm from (Wyder et al., 2026; Yermakov et al., 2026) adopted training data for all the fields $\mathbf{X}_{\text{train}}$ to predict the test data $\mathbf{X}_{\text{test}}$, adopting the following paradigm $\mathbf{x}_{j+1} = \mathcal{F}_{\boldsymbol{\theta}}(\mathbf{x}_j)$. On the other hand, the novel approach, summarised in Figure 3, provides as training data the state matrix $\mathbf{X}_{\text{train}}$ and the measurements of only field 1 (as an example), $\mathbf{y}^{(1)}_{\text{train}} = \mathcal{H}_{(1)}(\mathbf{X}_{\text{train}})$, where $\mathcal{H}_{(1)}$ denotes the corresponding observation operator, to predict the test matrix $\mathbf{X}_{\text{test}}$ starting from $\mathbf{y}^{(1)}_{\text{test}} = \mathcal{H}_{(1)}(\mathbf{X}_{\text{test}})$

[4]For more details on the mathematical aspects, the reader is referred to Appendix B.

only. Accordingly, the task is now changed to be $\mathbf{x}_j = \mathcal{G}_{\boldsymbol{\theta}}(\mathbf{y}_j^{(1)})$. We introduce partial observability as an additional evaluation dimension and we do not require defining a separate family of error measures, since the prediction target will always be the full state.

3 Evaluation Results

In the previous sections, the datasets have been briefly described: in particular, we remind that they include three different modelling strategies for nuclear reactors (lumped, system codes and multi-physics approach). In the following, we will present the evaluation of the standard set of metrics on the MSFR dataset as demonstration of the framework: this reactor is indeed the most interesting case study due to the strong intrinsic coupling between thermal-hydraulics and neutronics, resulting in a system of around 20 coupled PDEs (Appendix A). Furthermore, a summary of a sensitivity analysis on the new paradigm on sparse reconstruction adopting the Empirical Interpolation Method[5] (Maday et al., 2008; 2015; Silva et al., 2021) is presented; more details are available in Appendix C.1.
For the standard evaluation on the MSFR, we have investigated several highly-cited modelling methods: both regression techniques (e.g., DMD, Koopman-based methods, SINDy), deep learning methods (e.g., LSTM, DeepONet, FNO) and foundation models are included. In addition to the methods, the CTF4NUCLEAR includes three baselines: a constant zero prediction (Baseline Zeros), a constant average prediction based on the training data (Baseline Averages) and a last-value prediction (Baseline Last), in which the last observed state is used as a prediction for all future states[6]. These baselines have been also run over the other datasets and are reported in Appendix C.

3.1 Standard Evaluation Results on the MSFR

Table 1 shows all the scored methods and their resulting performance scores on the standard set of metrics, ranking them by their average score across all tasks: we notice how only Koopman-based methods are able to outperform the last-value baseline. Furthermore, due to the high dimensionality of the spatial domain and the number of fields $O(10^4)$, some methods have failed to generate a result for some or all the tasks, caused by memory issues/excess of time required to train the model. This is the case for some foundation models such as Moirai-MoE, Chronos-T5, Chronos-2, TabPFN and LLMTime (grouped in Foundation Models OOM), which have not been designed to handle such high-dimensional spatio-temporal data; the same issue is observed for Spacetime. In order to reduce the dimensionality of the data, we applied a POD dimensionality reduction as a pre-processing step (Quarteroni et al., 2015): only following this step we were able to obtain results for these models. This approach has been also applied to Reservoir computing and SINDy. In the end, we want to recall that if a method fails to generate a result for a task, it is assigned the lowest score of -100.
At first, we notice that *PyKoopman* is the only method to outperform the last-value baseline, reaching the highest average score overall, while also exhibiting the lowest task-to-task variability among the competitive methods. This is a significant finding, as it suggests that the MSFR transient under study is dominated by low-frequency persistent dynamics: turbulence has been modelled using a Reynolds-Averaged Navier-Stokes approach, which filters out high-frequency fluctuations, and the imposed pump and heat-exchanger forcing drives the reactor towards a new power level around which the state oscillates. Recovering this new pseudo-steady state therefore already accounts for most of the signal energy, and a method that merely propagates the last observed state scores highly. This aspect is of great importance for nuclear reactors operators: they are frequently more concerned with the global behaviour of the core than with low-scale features. Therefore, persistence is a legitimately strong reference for slowly evolving reactor transients.
Furthermore, we can see that regression techniques such as DMD and SINDy are able to perform well over almost all the tasks, being better than deep learning methods such as LSTM, DeepONet and FNO, with Reservoir computing being an exception. This last method performs extremely well for forecasting, noisy data and parametric generalisation, but it fails completely to generalise to limited data. (Wyder et al., 2026; Yermakov et al., 2026) have highlighted that Reservoir computing is a method that can either perform extremely well or fail completely depending on the task, and this

[5]This method is considered as the baseline for both state estimation and sensor positioning.

[6]In slowly-evolving engineering systems, persistence is not a weak baseline but a genuinely strong predictor, and it is precisely the baseline that industrial monitoring practice can rely on. A surrogate model that does not beat persistence provides no operational value.

Table 1: Model scores on the MSFR dataset (higher is better; bold marks column maxima), ordered by their average score. The related column reports the mean and standard deviation over E_1–E_{12}. Cell colors encode rank within each column (darker = better) using task-specific hues. We evaluate: Chronos-T5 (Ansari et al., 2024), Chronos-2 (Ansari et al., 2025), DeepONet (Chen & Chen, 1995), FNO (Li et al., 2021b), KAN (Liu et al., 2025c), LLMTime (Gruver et al., 2023), LSTM (Hochreiter & Schmidhuber, 1997), Moirai-MoE (Liu et al., 2025a), Moirai-2 (Liu et al., 2025a), NeuralODE (Ruthotto, 2024), ODE-LSTM (Coelho et al., 2024), Opt DMD (Askham & Kutz, 2018), Panda (Lai et al., 2025), PyKoopman (Brunton et al., 2022; Pan et al., 2024), Reservoir (Jaeger, 2001b; Maass & Markram, 2004; Pathak et al., 2018), SINDy (Brunton et al., 2016; Fasel et al., 2022), SpaceTime (Zhang et al., 2023), Sundial (Liu et al., 2025b), and TabPFN (Hoo et al., 2025). Failed runs are assigned a sentinel score of -100: in particular, Chronos-2, LLMTime, TabPFN, Chronos-T5, Spacetime and Moirai-MoE are grouped in Foundation Models OOM (Out-Of-Memory).

Model	Average	Forecasting		Noisy Data				Limited Data				Parametric Gen.	
	Score ± SD	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12
PyKoopman	**70.97 ± 22.03**	87.46	31.12	90.91	71.81	86.92	**85.00**	**92.72**	61.21	**92.59**	65.95	43.36	42.56
Baseline Last	62.67 ± 34.29	81.20	71.79	25.45	71.79	-25.46	71.79	85.90	49.52	85.88	49.53	94.03	90.65
Reservoir	57.22 ± 73.93	**92.36**	**87.44**	98.85	**88.42**	98.85	66.02	92.09	**92.78**	-100	-100	86.08	83.75
SINDy	55.52 ± 14.04	75.35	42.89	41.38	42.90	41.47	42.80	73.83	52.89	73.83	52.90	70.06	55.92
Opt DMD	53.14 ± 25.40	72.13	41.22	90.71	41.25	91.33	41.11	60.87	50.11	61.91	52.27	36.57	-1.80
ODE-LSTM	52.70 ± 25.24	80.73	61.42	5.69	30.06	93.01	52.45	78.31	42.27	34.42	66.85	56.52	30.74
LSTM	50.07 ± 29.69	49.26	30.35	97.02	14.21	96.39	12.15	81.42	49.22	58.03	23.59	59.12	30.02
Moirai-2	47.24 ± 70.27	84.85	72.54	-100	71.72	-100	66.36	89.33	53.74	88.77	51.52	94.79	**93.29**
Moirai-2 (POD)	46.04 ± 69.75	86.18	70.86	-100	70.84	-100	70.84	86.52	49.12	86.49	48.97	90.26	92.44
NeuralODE	40.29 ± 41.67	37.98	60.71	-16.67	7.20	2.24	73.27	-20.61	69.98	12.71	71.47	94.23	90.98
Moirai-MoE (POD)	38.70 ± 67.64	83.78	52.95	-100	56.61	-100	49.84	81.71	33.01	81.19	43.42	90.37	91.55
Chronos-2 (POD)	37.10 ± 68.61	89.20	50.67	-100	50.71	-100	50.34	84.50	29.68	84.44	20.89	91.57	93.23
DeepONet	36.64 ± 17.07	35.42	34.42	26.79	57.75	26.23	56.80	32.97	32.33	34.90	68.81	28.22	5.02
Sundial (POD)	29.32 ± 69.20	89.55	53.37	-100	19.36	-100	8.81	86.52	14.40	86.91	12.59	89.69	90.63
Chronos-T5 (POD)	28.97 ± 62.99	86.38	3.65	-100	23.97	-56.00	30.72	89.13	1.51	88.30	2.03	90.58	87.33
Sundial	27.37 ± 68.54	24.19	82.62	-100	21.82	-100	27.51	9.36	86.54	6.24	**83.60**	**94.97**	91.64
Baseline Average	9.58 ± 24.63	9.38	1.02	0.51	1.02	0.51	1.02	-14.42	5.16	-14.42	5.16	57.27	62.69
KAN	5.95 ± 23.37	9.18	0.97	72.10	21.96	3.23	0.91	5.08	-14.10	2.31	-4.19	-7.06	-18.96
FNO	1.59 ± 82.94	77.82	-100	**99.97**	4.58	**99.88**	-100	79.42	-100	7.73	-100	56.82	-7.12
Baseline Zeros	0.00 ± 0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Panda	-37.84 ± 77.18	44.97	-100	-100	-100	-100	-100	32.34	-100	46.01	-100	63.99	58.57
Panda (POD)	-76.55 ± 40.91	-100	-100	-100	-100	-100	-100	-27.61	-100	-10.97	-100	8.16	-88.24
Foundation Models OOM	-100 ± 0.00	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100	-100

is also observed in this work. The fact that regression techniques outperform deep learning methods is also related to the fact that the amount of time series data available for training is relatively small, and the system's dynamics are dominated by a few coherent modes, which favour linear and quasi-linear decompositions over non-linear architectures.
Finally, the failure of the direct use of foundation models may be related to a scope mismatch: these architectures are designed for univariate or low-dimensional multivariate time series, and the MSFR dataset comprises spatial degrees of freedom coupled across all physical fields. The only exceptions are Moirai-2, which achieves a respectable average of 47.24, and Sundial which is able to achieve a score of 27.4 adopting a spatial batching strategy. This also illustrates the principle that specialized models often beat general ones in complex domains (Goldfeder et al., 2026a).

3.2 Sparse State-Estimation Analysis

To provide an initial demonstration of the sparse-observation paradigm introduced in Section 2, we consider the Empirical Interpolation Method (EIM) as a transparent reference method for full-state reconstruction from sparse measurements: this method is an established baseline within the nuclear engineering community to monitor reactors (Gong et al., 2016; 2022a; Cammi et al., 2024). As mentioned above, the task shifts from forecasting future states towards estimating the full state $\mathcal{O}(n)$ from sparse observations only $\mathcal{O}(s)$, $s \ll n$. From the practical point of view, in nuclear reactors, not all the quantities of interest are observable: it is indeed quite easy to measure the temperature or the power but rather difficult to have local access to the velocity or precursors concentrations (Gong et al., 2022b; Riva et al., 2025). Furthermore, this task opens more questions on how to evaluate a method; in fact, state-estimation performance depends not only on the physical quantity being measured, but also on the number of available sensors.
This is confirmed on the MSFR dataset (more details are reported in Appendix C.1), where significant gains are generally obtained within $5 \div 20$ sensors. More importantly, the most informative field

is task dependent: velocity and power-density measurements are particularly effective for several transient forecasting tasks, whereas temperature measurements provide the strongest reconstruction for parametric interpolation. This may be related to the fact that the power and the velocity are the main driving forces of the transient and exhibit faster dynamics; whereas, the temperature is more useful in discriminant dominant physical behaviour. These results illustrate why sparse state estimation constitutes a distinct benchmark dimension rather than merely a reduced-input version of the standard forecasting problem.

4 LIMITATIONS & FUTURE WORK

The framework contains datasets spanning different reactor concepts, physical regimes, modelling fidelities, and observability conditions, the comprehensive evaluation on highly cited ML models has been presented in the main text to the MSFR dataset. We do not claim that the method ranking obtained on the MSFR transfers to the other datasets automatically: in fact, our rationale for assembling a collection of datasets spanning different modelling fidelities is that such transfer cannot be assumed. The HPMR, DYNASTY, and MHD datasets are therefore released with reference baselines and a withheld test set, so that the corresponding evaluations can be established by the community under the same protocol. Secondly, the datasets in the current release are generated by numerical models and therefore inherit the assumptions and modelling errors of the corresponding simulators: for instance, the MSFR dataset adopts RANS for turbulence modelling and multi-group diffusion for neutronics, resulting in a relatively smooth, low-frequency behaviour. Similarly, noise has been added to the data (for some tasks) as uncorrelated Gaussian noise, which cannot capture the full structure of measurement uncertainty in nuclear instrumentation. Moreover, the number of snapshots per dataset is also intentionally small relative to those commonly used to train modern deep-learning and foundation models. This reflects an important characteristic of nuclear engineering, where high-fidelity simulations are expensive to obtain. Finally, the sparse state estimation study presented here is an initial demonstration of the proposed task rather than a comprehensive benchmark. The EIM is used as a transparent reference method and investigate sensitivity to the measured field and number of sensors. The present metrics primarily quantify predictive and reconstruction accuracy; they do not yet evaluate uncertainty calibration or anomalous operating conditions. These are particularly relevant for safety-critical applications and represent natural extensions of CTF4NUCLEAR. The purpose of the current release is therefore not to establish a final ranking of learning algorithms, but to define a reproducible evaluation protocol upon which progressively more demanding datasets, tasks, and methods can be added. Following this aim, we believe that the current release can be adopted by researchers within the ML and nuclear community to establish a common benchmark for state estimation and data assimilation in nuclear systems.

5 CONCLUSIONS

The nuclear reactor community lacks a rigorous, standardized framework to evaluate the performance of data-driven surrogate models under the unique multi-physics and observability constraints of nuclear systems. To this end, we introduced CTF4NUCLEAR, the first open Common Task Framework specifically tailored to the unique multi-physics and observability constraints of nuclear systems. By incorporating a diversity of datasets and modelling choices, we want to establish a rigorous proving ground for scientific machine learning. Furthermore, by supplementing standard spatio-temporal metrics with a novel task dedicated to full-state reconstruction from sparse, out-of-core sensors, this framework aligns model evaluation directly with the safety and operational requirements of the nuclear industry. The initial evaluation on the MSFR dataset yields crucial insights into the current state of scientific machine learning for nuclear reactors. Most notably, standard deep learning architectures and time-series foundation models struggled to process the high spatial dimensionality and low-data regime. Conversely, regression-based approaches and operator theoretic methods, such as PyKoopman or SINDy, demonstrated superior robustness and predictive accuracy, effectively capturing the low-frequency dynamics dominant in the system.
In the end, we highlight that by exposing the advantages and weaknesses of celebrated learning algorithms we want to stress that the current CTF is not a final verdict on the value of any method; on the contrary, it is an invitation to researchers in the community to refine architectures and to build a more comprehensive benchmark suite for scientific machine learning in nuclear engineering.

AI USE STATEMENT

In this work, we used generative AI tools for research ideation and execution. Specifically, brainstorming sessions were carried out with the support of generative AI tools, though the final ideas, claims, and conclusions are the authors' own. AI tools assisted in cleaning and commenting the code structure; AI tools were used to generate the plotting code for the figures. AI tools also assisted in writing the code used to generate the synthetic benchmark trajectories for the HPMR point-kinetics dataset. AI tools were further used to aid and polish the writing: AI tools were used to check for typos and grammar errors; any generated text was subsequently rewritten by the authors, who are responsible for the final wording. We did not use generative AI tools to generate the benchmark datasets themselves, which are produced by established physics solvers (OpenFOAM and RELAP5). All AI-assisted work has been reviewed by the authors; AI-generated code was verified and tested for correctness. We take responsibility for the final content of this work, including text, claims and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The datasets (training data only) are available at hf.co/datasets/n-u-c-suppl-material. This work builds on the openly available `ctf4science` package to evaluate models, and the configuration files of each ML model are available in the same repository as the datasets. Furthermore, on HF we provide the scores for the MSFR dataset, as well as the code for the sensitivity analysis for state estimation with EIM. To preserve the integrity of the CTF and prevent adaptation to the evaluation data, the benchmark test set will remain withheld and final test scores will be computed through the benchmark evaluation pipeline.

REFERENCES

Francesco Andreuzzi, Nicola Demo, and Gianluigi Rozza. A Dynamic Mode Decomposition Extension for the Forecasting of Parametric Dynamical Systems. *SIAM Journal on Applied Dynamical Systems*, 22(3):2432–2458, September 2023. ISSN 1536-0040. doi: 10.1137/22M1481658. URL https://epubs.siam.org/doi/10.1137/22M1481658.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Yuyang Wang. Chronos: Learning the language of time series, 2024. URL https://arxiv.org/abs/2403.07815.

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, Xiyuan Zhang, et al. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

J.-P. Argaud, B Bouriquet, F de Caso, H Gong, Y Maday, and O Mula. Sensor placement in nuclear reactors based on the generalized empirical interpolation method. *Journal of Computational Physics*, 363:354–370, 2018. doi: 10.1016/j.jcp.2018.02.050. URL https://www.sciencedirect.com/science/article/pii/S0021999118301414.

Travis Askham and J. Nathan Kutz. Variable projection methods for an optimized dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 17(1):380–416, 2018. URL https://doi.org/10.1137/M1124176.

Manuele Aufiero, Antonio Cammi, Olivier Geoffroy, Mario Losa, Lelio Luzzi, Marco E. Ricotti, and Hervé Rouch. Development of an OpenFOAM model for the Molten Salt Fast Reactor transient analysis. *Chemical Engineering Science*, 111:390–401, May 2014. ISSN 0009-2509. doi: 10.1016/j.ces.2014.03.003. URL https://www.sciencedirect.com/science/article/pii/S0009250914001146.

M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, June 2013. ISSN 1076-9757. doi: 10.1613/jair.3912. URL http://dx.doi.org/10.1613/jair.3912.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym, 2016. URL https://arxiv.org/abs/1606.01540.

Mariya Brovchenko, Elsa Merle Lucotte, Hervé Rouch, Fabio Alcaro, M Allibert, M Aufiero, Antonio Cammi, S Dulla, O Feynberg, L Frima, et al. Optimization of the pre-conceptual design of the MSFR, 2013. URL https://www.janleenkloosterman.nl/reports/evol_d22_201309.pdf.

David Brown. Artificial intelligence for accelerating nuclear applications, science, and technology. Technical report, Brookhaven National Laboratory (BNL), Upton, NY (United States), 07 2022. URL https://www.osti.gov/biblio/1913826.

Steven L. Brunton and J. Nathan Kutz. *Data-Driven Science and Engineering: Machine Learning, Dynamical Systems, and Control*. Cambridge University Press, 2nd edition, 2022. doi: 10.1017/9781009089517.

Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016. doi: 10.1073/pnas.1517384113. URL https://www.pnas.org/doi/abs/10.1073/pnas.1517384113.

Steven L. Brunton, Marko Budišić, Eurika Kaiser, and J. Nathan Kutz. Modern Koopman Theory for Dynamical Systems. *SIAM Review*, 64(2):229–340, 2022. doi: 10.1137/21M1401243. URL https://doi.org/10.1137/21M1401243.

Priyanshu Burark, Karn Tiwari, Meer Mehran Rashid, Prathosh AP, and NM Anoop Krishnan. Codbench: a critical evaluation of data-driven models for continuous dynamical systems. *Digital Discovery*, 3(6):1172–1181, 2024.

A Cammi, MT Cauzzi, L Luzzi, A Pini, M Ricotti, A Siviero, D Lecarpentier, C Peniguel, D Lathouwers, M Tiberga, et al. Dynamics in natural circulation loop for internally heated molten salt. Technical report, SAMOFAR Project (H2020), 2019.

Antonio Cammi, Stefano Riva, Carolina Introini, Lorenzo Loi, and Enrico Padovani. Data-driven model order reduction for sensor positioning and indirect reconstruction with noisy data: Application to a Circulating Fuel Reactor. *Nuclear Engineering and Design*, 421:113105, May 2024. ISSN 0029-5493. doi: 10.1016/j.nucengdes.2024.113105. URL https://www.sciencedirect.com/science/article/pii/S002954932400205X.

E. Cervi, S. Lorenzi, A. Cammi, and L. Luzzi. Development of a multiphysics model for the study of fuel compressibility effects in the molten salt fast reactor. *Chemical Engineering Science*, 193: 379–393, 2019. ISSN 0009-2509. doi: https://doi.org/10.1016/j.ces.2018.09.025. URL https://www.sciencedirect.com/science/article/pii/S0009250918306742.

Kathleen Champion, Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116 (45):22445–22451, 2019. doi: 10.1073/pnas.1906995116. URL https://www.pnas.org/doi/abs/10.1073/pnas.1906995116.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 6572–6583, Red Hook, NY, USA, 2018. Curran Associates Inc.

Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its applications to dynamic systems. *Neural Networks, IEEE Transactions on*, pp. 911 – 917, 08 1995. doi: 10.1109/72.392253.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

C. Coelho, M. Fernanda P. Costa, and Luis L. Ferrás. Enhancing continuous time series modelling with a latent ode-lstm approach. *Applied Mathematics and Computation*, 475:128727, 2024.

Brian de Silva, Kathleen Champion, Markus Quade, Jean-Christophe Loiseau, J. Kutz, and Steven Brunton. Pysindy: A python package for the sparse identification of nonlinear dynamical systems from data. *Journal of Open Source Software*, 5(49):2104, May 2020. ISSN 2475-9066. doi: 10.21105/joss.02104. URL http://dx.doi.org/10.21105/joss.02104.

Christophe Demazière. 6 - Neutronic/thermal-hydraulic coupling. In Christophe Demazière (ed.), *Modelling of Nuclear Reactor Multi-physics*, pp. 311–336. Academic Press, 2020. ISBN 978-0-12-815069-6. doi: https://doi.org/10.1016/B978-0-12-815069-6.00006-4. URL https://www.sciencedirect.com/science/article/pii/B9780128150696000064.

Nicola Demo, Marco Tezzele, and Gianluigi Rozza. PyDMD: Python Dynamic Mode Decomposition. *Journal of Open Source Software*, 3(22):530, 2018. doi: 10.21105/joss.00530. URL https://doi.org/10.21105/joss.00530. Publisher: The Open Journal.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, 2009.

David Donoho. 50 years of data science. *Journal of Computational and Graphical Statistics*, 26(4): 745–766, 2017.

James J. Duderstadt and Louis J. Hamilton. *Nuclear Reactor Analysis*. John Wiley & Sons, New York, 1976.

Farbod Faraji, Maryam Reza, Aaron Knoll, and J. Nathan Kutz. Data-driven local operator finding for reduced-order modelling of plasma systems: II. Application to parametric dynamics, 2024. URL https://arxiv.org/abs/2403.01532. _eprint: 2403.01532.

Farbod Faraji, Maryam Reza, and J Nathan Kutz. Shallow recurrent decoder for reduced order modeling of e × b plasma dynamics. *Machine Learning: Science and Technology*, 6(2): 025024, apr 2025. doi: 10.1088/2632-2153/adcd20. URL https://dx.doi.org/10.1088/2632-2153/adcd20.

Urban Fasel, J. Nathan Kutz, Bingni W. Brunton, and Steven L. Brunton. Ensemble-sindy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 478(2260):20210904, 2022. doi: 10.1098/rspa.2021.0904. URL https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2021.0904.

Carlo Fiorina, Ivor Clifford, Manuele Aufiero, and Konstantin Mikityuk. Gen-foam: a novel openfoam® based multi-physics solver for 2d/3d transient analysis of nuclear reactors. *Nuclear Engineering and Design*, 294:24–37, 2015. ISSN 0029-5493. doi: https://doi.org/10.1016/j.nucengdes.2015.05.035. URL https://www.sciencedirect.com/science/article/pii/S0029549315003829.

CD Fletcher and RR Schultz. Relap5/mod3 code manual. Technical report, Nuclear Regulatory Commission, Washington, DC (United States). Div. of . . . , 1992.

Generation IV International Forum. Technology roadmap update for generation IV nuclear energy systems. Technical report, OECD Nuclear Energy Agency, 2014. URL https://www.gen-4.org/gif/jcms/c_60729/technology-roadmap-update-2014.

Xiaoxu Geng, Dandan Chen, Jiayi Yu, Chong Wang, Changjun Hu, and Zhaoshun Wang. Physics-informed neural network for solving neutron transport equation with differential order transformation and fourier feature. *Journal of Computational Physics*, 546:114519, 2026. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2025.114519. URL https://www.sciencedirect.com/science/article/pii/S0021999125008010.

Péter German, Mauricio Tano, Carlo Fiorina, and Jean C. Ragusa. Gen-rom—an openfoam®-based multiphysics reduced-order modeling framework for the analysis of molten salt reactors. *Progress in Nuclear Energy*, 146:104148, 2022. ISSN 0149-1970. doi: https://doi.org/10.1016/j.pnucene.2022.104148. URL https://www.sciencedirect.com/science/article/pii/S0149197022000282.

Judah Goldfeder, Philippe Wyder, Yann LeCun, and Ravid Shwartz Ziv. Ai must embrace specialization via superhuman adaptable intelligence. *arXiv preprint arXiv:2602.23643*, 2026a.

Judah Goldfeder, Philippe M Wyder, J Nathan Kutz, and David Donoho. Driving science faster and more credibly with the common task framework. *Computing in Science & Engineering*, 2026b.

H Gong, J P Argaud, B Bouriquet, and Y Maday. The empirical interpolation method applied to the neutron diffusion equations with parameter dependence. *Physics of Reactors 2016, PHYSOR 2016: Unifying Theory and Experiments in the 21st Century*, 1(May):54–63, 2016. ISSN 9781510825734.

Helin Gong, Zhang Chen, and Qing Li. Generalized Empirical Interpolation Method With H1 Regularization: Application to Nuclear Reactor Physics. *Frontiers in Energy Research*, 9, 2022a. ISSN 2296-598X. doi: 10.3389/fenrg.2021.804018. URL https://www.frontiersin.org/articles/10.3389/fenrg.2021.804018.

Helin Gong, Sibo Cheng, Zhen Chen, and Qing Li. Data-enabled physics-informed machine learning for reduced-order modeling digital twin: application to nuclear reactor physics. *Nuclear Science and Engineering*, 196(6):668–693, 2022b. doi: 10.1080/00295639.2021.2014752.

Helin Gong, Tao Zhu, Zhang Chen, Yaping Wan, and Qing Li. Parameter identification and state estimation for nuclear reactor operation digital twin. *Annals of Nuclear Energy*, 180: 109497, January 2023. ISSN 03064549. doi: 10.1016/j.anucene.2022.109497. URL https://linkinghub.elsevier.com/retrieve/pii/S0306454922005278.

Vignesh Gopakumar, Stanislas Pamela, Lorenzo Zanisi, Zongyi Li, Ander Gray, Daniel Brennand, Nitesh Bhatia, Gregory Stathopoulos, Matt Kusner, Marc Peter Deisenroth, Anima Anandkumar, JOREK Team, and MAST Team. Plasma surrogate modelling using Fourier neural operators. *arXiv preprint arXiv:2311.05967*, 2024. URL https://arxiv.org/abs/2311.05967.

Michael W. Grieves. *Virtually Intelligent Product Systems: Digital and Physical Twins*, pp. 175–200. Progress in Astronautics and Aeronauting, 2019. doi: 10.2514/5.9781624105654.0175.0200. URL https://arc.aiaa.org/doi/abs/10.2514/5.9781624105654.0175.0200.

Nate Gruver, Marc Finzi, Shikai Qiu, and Andrew G Wilson. Large language models are zero-shot time series forecasters. *Advances in Neural Information Processing Systems*, 36:19622–19635, 2023.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 09 2025a. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Yue Guo, Milan Korda, Ioannis G Kevrekidis, and Qianxiao Li. Learning parametric koopman decompositions for prediction and control. *SIAM Journal on Applied Dynamical Systems*, 24(1): 744–781, 2025b.

Junyan He, Shashank Kushwaha, Jaewan Park, Seid Koric, Diab Abueidda, and Iwona Jasiuk. Sequential deep operator networks (s-deeponet) for predicting full-field solutions under time-dependent loads. *Engineering Applications of Artificial Intelligence*, 127:107258, January 2024a. ISSN 0952-1976. doi: 10.1016/j.engappai.2023.107258. URL http://dx.doi.org/10.1016/j.engappai.2023.107258.

Junyan He, Shashank Kushwaha, Jaewan Park, Seid Koric, Diab Abueidda, and Iwona Jasiuk. Predictions of transient vector solution fields with sequential deep operator network. *Acta Mechanica*, 235(8):5257–5272, June 2024b. ISSN 1619-6937. doi: 10.1007/s00707-024-03991-2. URL http://dx.doi.org/10.1007/s00707-024-03991-2.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021. URL https://arxiv.org/abs/2103.03874.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, January 2025. ISSN 1476-4687. doi: 10.1038/s41586-024-08328-6. URL https://doi.org/10.1038/s41586-024-08328-6.

Shi Bin Hoo, Samuel Müller, David Salinas, and Frank Hutter. From tables to time: How tabpfn-v2 outperforms specialized time series forecasting models, 2025. URL https://arxiv.org/abs/2501.02945.

Raisa Hossain, Farid Ahmed, Kazuma Kobayashi, Seid Koric, Diab Abueidda, and Syed Bahauddin Alam. Virtual sensing-enabled digital twin framework for real-time monitoring of nuclear systems leveraging deep neural operators. *npj Materials Degradation*, 9(1):21, 2025. doi: 10.1038/s41529-025-00557-y. URL https://doi.org/10.1038/s41529-025-00557-y.

Qingyu Huang, Shinian Peng, Jian Deng, Hui Zeng, Zhuo Zhang, Yu Liu, and Peng Yuan. A review of the application of artificial intelligence to nuclear reactors: Where we are and what's next. *Heliyon*, 9(3):e13883, 2023. ISSN 2405-8440. doi: https://doi.org/10.1016/j.heliyon.2023.e13883. URL https://www.sciencedirect.com/science/article/pii/S2405844023010903.

IAEA. Advances in small modular reactor technology developments. Technical report, International Atomic Energy Agency, Vienna, 2023.

Sara M. Ichinaga, Francesco Andreuzzi, Nicola Demo, Marco Tezzele, Karl Lapo, Gianluigi Rozza, Steven L. Brunton, and J. Nathan Kutz. PyDMD: A Python package for robust dynamic mode decomposition, 2024. URL https://arxiv.org/abs/2402.07463. _eprint: 2402.07463.

International Atomic Energy Agency. History, Development and Future of TRIGA Research Reactors. Technical Report 482, IAEA, 2016. URL https://www.iaea.org/publications/10943/history-development-and-future-of-triga-research-reactors.

Carolina Introini, Antonio Cammi, Stefano Lorenzi, and Giovanni Magrotti. An improved zero-dimensional model for simulation of triga mark ii dynamic response. *Progress in Nuclear Energy*, 111:85–96, 2019. ISSN 0149-1970. doi: https://doi.org/10.1016/j.pnucene.2018.10.025. URL https://www.sciencedirect.com/science/article/pii/S0149197018302774.

J. Nathan Kutz, Steven L. Brunton, Bingni W. Brunton, and Joshua L. Proctor. *Dynamic Mode Decomposition*. Society for Industrial and Applied Mathematics (SIAM), 2016. doi: 10.1137/1.9781611974508. URL https://epubs.siam.org/doi/book/10.1137/1.9781611974508.

Herbert Jaeger. The "echo state" approach to analysing and training recurrent neural networks – with an Erratum note. *GMD Report 148*, 2001a.

Herbet Jaeger. "the 'echo state' approach to analyzing and training recurrent neural networks". Technical report, German National Research Center for Information Technology, Technical Report GMD 148, 2001b.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021. doi: 10.1038/s41586-021-03819-2. URL https://doi.org/10.1038/s41586-021-03819-2.

Michael G. Kapteyn, David J. Knezevic, and Karen Willcox. Toward predictive digital twins via component-based reduced-order models and interpretable machine learning. In *AIAA Scitech 2020 Forum*, 2020. doi: 10.2514/6.2020-0418. URL https://arc.aiaa.org/doi/abs/10.2514/6.2020-0418. _eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2020-0418.

Kazuma Kobayashi and Syed Bahauddin Alam. Deep neural operator-driven real-time inference to enable digital twin solutions for nuclear energy systems. *Scientific Reports*, 14(1):2101, 2024. doi: 10.1038/s41598-024-51984-x. URL https://doi.org/10.1038/s41598-024-51984-x.

Andrei Nikolaevich Kolmogorov. On the representations of continuous functions of many variables by superposition of continuous functions of one variable and addition. In *Dokl. Akad. Nauk USSR*, volume 114, pp. 953–956, 1957.

Katiana Kontolati, Somdatta Goswami, George Em Karniadakis, and Michael D. Shields. Learning nonlinear operators in latent spaces for real-time predictions of complex dynamics in physical systems. *Nature Communications*, 15(1), June 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-49411-w. URL http://dx.doi.org/10.1038/s41467-024-49411-w.

Bernard O Koopman. Hamiltonian systems and transformation in hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.

Bernard O Koopman and J von Neumann. Dynamical systems of continuous spectra. *Proceedings of the National Academy of Sciences*, 18(3):255–263, 1932.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

Jeffrey Lai, Anthony Bao, and William Gilpin. Panda: A pretrained forecast model for chaotic dynamics. *arXiv preprint arXiv:2505.13755*, 2025.

Toni Lassila, Andrea Manzoni, Alfio Quarteroni, and Gianluigi Rozza. Model Order Reduction in Fluid Dynamics: Challenges and Perspectives. In *Reduced Order Methods for Modeling and Computational Reduction*, pp. 235–273. Springer International Publishing, Cham, 2014. ISBN 978-3-319-02090-7. doi: 10.1007/978-3-319-02090-7_9. URL https://doi.org/10.1007/978-3-319-02090-7_9.

Soledad Le Clainche and José M. Vega. Higher order dynamic mode decomposition. *SIAM Journal on Applied Dynamical Systems*, 16(2):882–925, 2017. doi: 10.1137/15M1054924. URL https://doi.org/10.1137/15M1054924.

Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for partial differential equations. In *International Conference on Learning Representations (ICLR)*, 2021a. URL https://arxiv.org/abs/2010.08895.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=c8P9NQVtmnO.

Guang Lin, Christian Moya, and Zecheng Zhang. Learning the dynamical response of nonlinear non-autonomous dynamical systems with deep operator neural networks. *Engineering Applications of Artificial Intelligence*, 125:106689, October 2023. ISSN 0952-1976. doi: 10.1016/j.engappai.2023.106689. URL http://dx.doi.org/10.1016/j.engappai.2023.106689.

Chenghao Liu, Taha Aksu, Juncheng Liu, Xu Liu, Hanshu Yan, Quang Pham, Silvio Savarese, Doyen Sahoo, Caiming Xiong, and Junnan Li. Moirai 2.0: When less is more for time series forecasting. *arXiv preprint arXiv:2511.11698*, 2025a.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models, 2025b. URL https://arxiv.org/abs/2502.00816.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. Kan: Kolmogorov-arnold networks, 2025c. URL https://arxiv.org/abs/2404.19756.

Matteo Lo Verso, Carolina Introini, Eric Cervi, Francesca Giacobbo, Luciana Barucca, Marco Caramello, Matteo Prinzio, Laura Savoldi, and Antonio Cammi. A novel openfoam library for magneto-hydrodynamics studies in the nuclear fusion field. In *Proceedings of NUTHOS-14 Conference*, 08 2024.

Lu Lu, Pengzhan Jin, Guofei Pang, Zhongqiang Zhang, and George Em Karniadakis. Learning nonlinear operators via deeponet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, 2021a. doi: 10.1038/s42256-021-00302-5. URL https://doi.org/10.1038/s42256-021-00302-5.

Lu Lu, Xuhui Meng, Zhiping Mao, and George Em Karniadakis. DeepXDE: A deep learning library for solving differential equations. *SIAM Review*, 63(1):208–228, 2021b. doi: 10.1137/19M1274067.

Lu Lu, Xuhui Meng, Shengze Cai, Zhiping Mao, Somdatta Goswami, Zhongqiang Zhang, and George Em Karniadakis. A comprehensive and fair comparison of two neural operators (with practical extensions) based on fair data. *Computer Methods in Applied Mechanics and Engineering*, 393:114778, 2022. ISSN 0045-7825. doi: https://doi.org/10.1016/j.cma.2022.114778. URL https://www.sciencedirect.com/science/article/pii/S0045782522001207.

Wolfgang Maass and Henry Markram. On the computational power of circuits of spiking neurons. *Journal of Computer and System Sciences*, 69(4):593–616, December 2004. ISSN 0022-0000. doi: 10.1016/j.jcss.2004.04.001. URL https://www.sciencedirect.com/science/article/pii/S0022000004000406.

Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations. *Neural Computation*, 14(11):2531–2560, November 2002. ISSN 0899-7667. doi: 10.1162/089976602760407955. URL https://doi.org/10.1162/089976602760407955.

Y Maday, O Mula, A T Patera, and M Yano. The Generalized Empirical Interpolation Method: Stability theory on Hilbert spaces with an application to the Stokes equation. *Computer Methods in Applied Mechanics and Engineering*, 287:310–334, 2015. doi: 10.1016/j.cma.2015.01.018. URL http://dx.doi.org/10.1016/j.cma.2015.01.018. Publisher: Elsevier B.V.

Yvon Maday. Reduced basis method for the rapid and reliable solution of partial differential equations. *International Congress of Mathematicians, ICM 2006*, 3:1255–1270, 2006. doi: 10.4171/022-3/60.

Yvon Maday, Ngoc Nguyen, Anthony Patera, and George Shu Heng Pau. A general multipurpose interpolation procedure: The magic points. *Communications on Pure and Applied Analysis*, 8, 2008. doi: 10.3934/cpaa.2009.8.383. URL https://doi.org/10.3934/cpaa.2009.8.383.

Yvon Maday, Anthony Patera, James Penn, and Masayuki Yano. A parameterized-background data-weak approach to variational data assimilation: formulation, analysis, and application to acoustics. *International Journal for Numerical Methods in Engineering*, 102, 2014. doi: 10.1002/nme.4747. URL https://doi.org/10.1002/nme.4747.

Amit Mangal, Vikas Jain, and AK Nayak. Capability of the relap5 code to simulate natural circulation behavior in test facilities. *Progress in Nuclear Energy*, 61:1–16, 2012.

Nick McGreivy and Ammar Hakim. Weak baselines and reporting biases lead to overoptimism in machine learning for fluid-related partial differential equations. *Nature Machine Intelligence*, 6(10):1256–1269, Oct 2024. doi: 10.1038/s42256-024-00897-5. URL https://doi.org/10.1038/s42256-024-00897-5.

Katarzyna Michałowska, Somdatta Goswami, George Em Karniadakis, and Signe Riemer-Sørensen. Neural operator learning for long-time integration in dynamical systems with recurrent neural networks. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2024. doi: 10.1109/IJCNN60899.2024.10650331.

Andrea Missaglia, Gabriele Benzoni, Carolina Introini, Marco Enrico Ricotti, Antonio Cammi, and Francois Foulon. Modeling and validation of the dynasty natural circulation loop with the relap5 code. *Results in Engineering*, 28:107466, 2025. ISSN 2590-1230. doi: https://doi.org/10.1016/j.rineng.2025.107466. URL https://www.sciencedirect.com/science/article/pii/S2590123025035212.

Subhasish Mohanty and Richard B. Vilim. Physics-Infused AI/ML Based Digital-Twin Framework for Flow-Induced-Vibration Damage Prediction in a Nuclear Reactor Heat Exchanger. Technical report, Argonne National Lab. (ANL), 2021. URL https://www.osti.gov/biblio/1830413.

OECD/NEA. https://www.oecd-nea.org/jcms/pl_77779, 2024. Accessed May 2026.

Ruben Ohana, Michael McCabe, Lucas Meyer, Rudy Morel, Fruzsina Agocs, Miguel Beneitez, Marsha Berger, Blakesly Burkhart, Stuart Dalziel, Drummond Fielding, et al. The well: a large-scale collection of diverse physics simulations for machine learning. *Advances in Neural Information Processing Systems*, 37:44989–45037, 2024.

Shaowu Pan, Eurika Kaiser, Brian M. de Silva, J. Nathan Kutz, and Steven L. Brunton. PyKoopman: A Python Package for Data-Driven Approximation of the Koopman Operator. *Journal of Open Source Software*, 9(94):5881, 2024. doi: 10.21105/joss.05881. URL https://doi.org/10.21105/joss.05881.

Jaideep Pathak, Brian Hunt, Michelle Girvan, Zhixin Lu, and Edward Ott. Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach. *Physical Review Letters*, 120(2):024102, January 2018. doi: 10.1103/PhysRevLett.120.024102. URL https://link.aps.org/doi/10.1103/PhysRevLett.120.024102.

A Pini, A Cammi, M Cauzzi, F Fanale, and L Luzzi. An Experimental Facility to Investigate the Natural Circulation Dynamics in Presence of Distributed Heat Sources. *Energy Procedia*, 101:10–17, 2016. doi: 10.1016/j.egypro.2016.11.002. URL https://www.sciencedirect.com/science/article/pii/S1876610216312115.

Jason A. Platt, Stephen G. Penny, Timothy A. Smith, Tse-Chun Chen, and Henry D. I. Abarbanel. A systematic exploration of reservoir computing for forecasting complex spatiotemporal dynamics. *Neural Networks*, 153:530–552, September 2022. ISSN 0893-6080. doi: 10.1016/j.neunet.2022.06.025. URL https://www.sciencedirect.com/science/article/pii/S0893608022002404.

Konstantinos Prantikos, Lefteri H. Tsoukalas, and Alexander Heifetz. Physics-informed neural network solution of point kinetics equations for a nuclear reactor digital twin. *Energies*, 15 (20), 2022. ISSN 1996-1073. doi: 10.3390/en15207697. URL https://www.mdpi.com/1996-1073/15/20/7697.

Konstantinos Prantikos, Stylianos Chatzidakis, Lefteri H. Tsoukalas, and Alexander Heifetz. Physics-informed neural network with transfer learning (tl-pinn) based on domain similarity measure for prediction of nuclear reactor transients. *Scientific Reports*, 13(1):16840, 2023. doi: 10.1038/s41598-023-43325-1. URL https://doi.org/10.1038/s41598-023-43325-1.

Zachary M. Prince, Joshua T. Hanophy, Vincent M. Labouré, Yaqi Wang, Logan H. Harbour, and Namjae Choi. Neutron transport methods for multiphysics heterogeneous reactor core simulation in griffin. *Annals of Nuclear Energy*, 200:110365, 2024. ISSN 0306-4549. doi: https://doi.org/10.1016/j.anucene.2024.110365. URL https://www.sciencedirect.com/science/article/pii/S0306454924000276.

A Quarteroni, A Manzoni, and F Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction*. UNITEXT. Springer Cham, 1 edition, 2015. ISBN 978-3-319-15431-2. URL https://doi.org/10.1007/978-3-319-15431-2.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

Stefano Riva, Sophie Deanesi, Carolina Introini, Stefano Lorenzi, and Antonio Cammi. Neutron Flux Reconstruction from Out-Core Sparse Measurements using Data-Driven Reduced Order Modelling. In *Proceedings of the International Conference on Physics of Reactors, PHYSOR 2024*, pp. 1632 – 1641, 2024. doi: 10.13182/PHYSOR24-43444. URL https://www.ans.org/pubs/proceedings/article-55632/.

Stefano Riva, Carolina Introini, Antonio Cammi, and J. Nathan Kutz. Robust state estimation from partial out-core measurements with shallow recurrent decoder for nuclear reactors. *Progress in Nuclear Energy*, 189:105928, 2025. ISSN 0149-1970. doi: https://doi.org/10.1016/j.pnucene.2025.105928. URL https://www.sciencedirect.com/science/article/pii/S0149197025003269.

Clarence W. Rowley, Igor Mezić, Shervin Bagheri, Philipp Schlatter, and Dan S. Henningson. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641:115–127, December 2009. ISSN 1469-7645, 0022-1120. doi: 10.1017/S0022112009992059. URL https://www.cambridge.org/core/journals/journal-of-fluid-mechanics/article/abs/spectral-analysis-of-nonlinear-flows/311041E1027AE7FEE7DDA36AC9AD4270. Publisher: Cambridge University Press.

Gianluigi Rozza, Martin Hess, Giovanni Stabile, Marco Tezzele, Francesco Ballarin, Carmen Gräßle, Michael Hinze, Stefan Volkwein, Francisco Chinesta, Pierre Ladeveze, Yvon Maday, Anthony Patera, and J Farhat Char. *Model Order Reduction: Volume 2: Snapshot-Based Methods and Algorithms*. De Gruyter, 2020. ISBN 978-3-11-067149-0. doi: 10.1515/9783110671490. URL https://doi.org/10.1515/9783110671490.

Mark F. Ruth, Owen R. Zinaman, Mark Antkowiak, Richard D. Boardman, Robert S. Cherry, and Morgan D. Bazilian. Nuclear-renewable hybrid energy systems: Opportunities, interconnections, and needs. *Energy Conversion and Management*, 78:684–694, 2014. ISSN 0196-8904. doi: https://doi.org/10.1016/j.enconman.2013.11.030. URL https://www.sciencedirect.com/science/article/pii/S0196890413007516.

Lars Ruthotto. Differential equations for continuous-time deep learning. *arXiv preprint arXiv:2401.03965*, 2024.

Diya Sashidhar and J. Nathan Kutz. Bagging, optimized dynamic mode decomposition for robust, stable forecasting with spatial and temporal uncertainty quantification. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 380(2229):20210199, 2022. doi: 10.1098/rsta.2021.0199. URL https://royalsocietypublishing.org/doi/abs/10.1098/rsta.2021.0199.

Claudio Scardino, Stefano Riva, Carolina Introini, Matteo Lo Verso, Eric Cervi, Antonio Cammi, and Laura Savoldi. Real-time monitoring of mhd liquid metal flows with shallow recurrent decoders, 2026. URL https://arxiv.org/abs/2608.28366.

P. J. Schmid. Dynamic Mode Decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010. doi: 10.1017/S0022112010001217. URL https://doi.org/10.1017/S0022112010001217. Publisher: Cambridge University Press.

Francesco A B Silva, Stefano Lorenzi, and Antonio Cammi. An empirical interpolation method for two-dimensional vector fields and vector measurements. *International Journal for Numerical Methods in Engineering*, 122(15):3733–3748, 2021. doi: 10.1002/nme.6679.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.

Makoto Takamoto, Timothy Praditia, Raphael Leiteritz, Daniel MacKinlay, Francesco Alesiani, Dirk Pflüger, and Mathias Niepert. Pdebench: An extensive benchmark for scientific machine learning. *Advances in Neural Information Processing Systems*, 35:1596–1611, 2022.

Gouhei Tanaka, Toshiyuki Yamane, Jean Benoit Héroux, Ryosho Nakane, Naoki Kanazawa, Seiji Takeda, Hidetoshi Numata, Daiju Nakano, and Akira Hirose. Recent advances in physical reservoir computing: A review. *Neural Networks*, 115:100–123, July 2019. ISSN 0893-6080. doi: 10.1016/j.neunet.2019.03.005. URL https://www.sciencedirect.com/science/article/pii/S0893608019300784.

Neil E. Todreas and Mujid S. Kazimi. *Nuclear Systems I: Thermal Hydraulic Fundamentals*. Hemisphere Publishing Co., New York, 1990. ISBN 0891169350.

Matteo Tomasetto, Jan P. Williams, Francesco Braghin, Andrea Manzoni, and J. Nathan Kutz. Reduced order modeling with shallow recurrent decoder networks. *Nature Communications*, 16 (1):10260, 2025. doi: 10.1038/s41467-025-65126-y. URL https://doi.org/10.1038/s41467-025-65126-y.

H K Versteeg and W Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Pearson Education Limited, 2007. ISBN 9780131274983. URL https://books.google.it/books?id=RvBZ-UMpGzIC.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, Ilhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, Aditya Vijaykumar, Alessandro Pietro Bardelli, Alex Rothberg, Andreas Hilboll, Andreas Kloeckner, Anthony Scopatz, Antony Lee, Ariel Rokem, C. Nathan Woods, Chad Fulton, Charles Masson, Christian Häggström, Clark Fitzgerald, David A. Nicholson, David R. Hagen, Dmitrii V. Pasechnik, Emanuele Olivetti, Eric Martin, Eric Wieser, Fabrice Silva, Felix Lenders, Florian Wilhelm, G. Young, Gavin A. Price, Gert-Ludwig Ingold, Gregory E. Allen, Gregory R. Lee, Hervé Audren, Irvin Probst, Jörg P. Dietrich, Jacob Silterra, James T Webber, Janko Slavič, Joel Nothman, Johannes Buchner, Johannes Kulick, Johannes L. Schönberger, José Vinícius de Miranda Cardoso, Joscha Reimer, Joseph Harrington, Juan Luis Cano Rodríguez, Juan Nunez-Iglesias, Justin Kuczynski, Kevin Tritz, Martin Thoma, Matthew Newville, Matthias Kümmerer, Maximilian Bolingbroke, Michael Tartre, Mikhail Pak, Nathaniel J. Smith, Nikolai Nowaczyk, Nikolay Shebanov, Oleksandr Pavlyk, Per A. Brodtkorb, Perry Lee, Robert T. McGibbon, Roman Feldbauer, Sam Lewis, Sam Tygier, Scott Sievert, Sebastiano Vigna, Stefan Peterson, Surhud More, Tadeusz Pudlik, Takuya Oshima, Thomas J. Pingel, Thomas P. Robitaille, Thomas Spura, Thouis R. Jones, Tim Cera, Tim Leslie, Tiziano Zito, Tom Krauss, Utkarsh Upadhyay, Yaroslav O. Halchenko, and Yoshiki Vázquez-Baeza. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature Methods*, 17(3):261–272, February 2020. ISSN 1548-7105. doi: 10.1038/s41592-019-0686-2. URL http://dx.doi.org/10.1038/s41592-019-0686-2.

P. R. Vlachas, J. Pathak, B. R. Hunt, T. P. Sapsis, M. Girvan, E. Ott, and P. Koumoutsakos. Backpropagation algorithms and Reservoir Computing in Recurrent Neural Networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks: The Official Journal of the International Neural Network Society*, 126:191–217, June 2020. ISSN 1879-2782. doi: 10.1016/j.neunet.2020.02.016.

Jiangyu Wang, Xingjie Peng, Zhang Chen, Bingyan Zhou, Yajin Zhou, and Nan Zhou. Surrogate modeling for neutron diffusion problems based on conservative physics-informed neural networks with boundary conditions enforcement. *Annals of Nuclear Energy*, 176:109234, 2022. ISSN 0306-4549. doi: https://doi.org/10.1016/j.anucene.2022.109234. URL https://www.sciencedirect.com/science/article/pii/S0306454922002699.

Sifan Wang, Mohamed Aziz Bhouri, and Paris Perdikaris. Fast pde-constrained optimization via self-supervised operator learning, 2021. URL https://arxiv.org/abs/2110.13297.

Yaqi Wang, Zachary M Prince, Hansol Park, Olin W Calvin, Namjae Choi, Yeon Sang Jung, Sebastian Schunert, Shikhar Kumar, Joshua T Hanophy, Vincent M Labouré, et al. Griffin: A moose-based reactor physics application for multiphysics simulation of advanced nuclear reactors. *Annals of Nuclear Energy*, 211:110917, 2025.

Jan P. Williams, Olivia Zahn, and J. Nathan Kutz. Sensing with shallow recurrent decoder networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 480(2298):20240054, 2024. doi: 10.1098/rspa.2024.0054. URL https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2024.0054.

Philippe Wyder, Judah Goldfeder, Alexey Yermakov, Yue Zhao, Stefano Riva, Jan Williams, David Zoro, Amy Rude, Matteo Tomasetto, Joe Germany, et al. Common task framework for a critical

evaluation of scientific machine learning algorithms. *Advances in Neural Information Processing Systems*, 38, 2026.

Alexey Yermakov, Yue Zhao, Marine Denolle, Yiyu Ni, Philippe Martin Wyder, Judah Goldfeder, Stefano Riva, Jan P. Williams, David Zoro, Amy Sara Rude, Matteo Tomasetto, Joe Germany, Joseph Bakarji, Georg Maierhofer, Miles Cranmer, and J. Nathan Kutz. The seismic wavefield common task framework. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=u4N7Kl6gzE.

Michael Zhang, Khaled K Saab, Michael Poli, Tri Dao, Karan Goel, and Christopher Ré. Effectively modeling time series with simple discrete state spaces. *arXiv preprint arXiv:2303.09489*, 2023.

Rui-Zhi Zhang, Sheng-Feng Zhu, Kan Wang, Ding She, Jean-Philippe Argaud, Bertrand Bouriquet, Qing Li, and He-Lin Gong. Artificial intelligence in reactor physics: current status and future prospects. *Nuclear Science and Techniques*, 37(6):100, 2026. doi: 10.1007/s41365-026-01928-z. URL https://doi.org/10.1007/s41365-026-01928-z.

Overview: this appendix contains the supplementary materials for the paper. In particular, the metrics are more rigorously defined and the mathematical models behind each dataset are presented. Furthermore, baseline results on all datasets are reported, along with a preliminary study on reconstruction from sparse measurements is presented for the MSFR. Finally, for each ML model that was evaluated, we share additional implementation and hyperparameter tuning details.

A Detailed description of the datasets

In this section, we are going to present the mathematical models behind each dataset, along with some comments on the numerical solution used to generate the data.

A.1 Heat-Pipe MicroReactor: lumped 0D model with point kinetics

The HPMR dataset is generated from transient simulations of a representative heat pipe microreactor (Prince et al., 2024) focusing on time-dependent neutron kinetics. In the real system, the transients are reactivity-driven and are initiated by prescribed control-drum motion, which induces changes in both the spatial flux shape and the overall neutron flux magnitude. Such reactivity insertions result in short-time-scale transients.
For the CTF4Nuclear, we adopt a lumped 0D description of the same reactivity-driven kinetics, in which the spatially distributed neutronic fields are collapsed onto a low-dimensional state that retains the stiff coupling between prompt and delayed neutrons. This reduction follows the classical point-kinetics (PK) approximation (Duderstadt & Hamilton, 1976): the space–time dependent neutron flux is replaced by a single amplitude $n(t)$, interpreted as the normalized reactor power ($n \equiv P/P_0$), and the delayed-neutron dynamics are represented by a small number of precursor groups $c_i(t)$. The reactivity insertion is encoded in a scalar $\rho(t)$, acting as control input for the system.

We consider two delayed-neutron groups with decay constants λ_1, λ_2 and delayed fractions β_1, β_2, and prompt-neutron generation time Λ. Reactivity $\rho(t)$ is expressed in *dollars* (\$), which is relative to the fraction of delayed neutrons $\beta = \beta_1 + \beta_2$, with criticality corresponding to $\rho = 0$. The PK equations read

$$\begin{cases} \dot{n}(t) = \dfrac{\rho(t) - 1}{\Lambda} \left(\beta_1 + \beta_2\right) n(t) + \lambda_1 c_1(t) + \lambda_2 c_2(t), \\ \dot{c}_i(t) = \dfrac{\beta_i}{\Lambda}\, n(t) - \lambda_i c_i(t), \qquad i = 1, 2, \end{cases} \tag{1}$$

where n is the normalized power, c_i are precursor concentrations in the PK scaling used for data generation. At critical conditions (steady state with $\rho = 0$), the precursor concentrations satisfy $\dot{c}_i = 0$, yielding the equilibrium relations $c_i^{\mathrm{crit}}(n) = \frac{\beta_i}{\Lambda \lambda_i}\, n$. We assume to start from initial power $n(0) = 1$, precursors $c_i(0) = c_i^{\mathrm{crit}}(1)$.

We introduce a nonlinear reactivity feedback $\rho(t)$ that depends on the Lorenz system, which is a classical low-dimensional chaotic system. The Lorenz system is defined by the following set of ordinary differential equations:

$$\begin{cases} \dot{x} = \sigma\,(y - x), \\ \dot{y} = x\,(\zeta - z) - y, \\ \dot{z} = x\,y - \xi\,z, \end{cases} \tag{2}$$

where σ, ζ, and ξ are positive parameters. In particular, the reactivity is defined as $\rho(t) = \varepsilon_1(x + y + z) + \varepsilon_2 \log n$. The term $\varepsilon_1(x + y + z)$ emulates a fast, state-dependent reactivity insertion analogous to control motion and coupled feedback, while $\varepsilon_2 \log n$ introduces a logarithmic power feedback. The state vector is then composed of the PK variables (n, c_1, c_2) and the Lorenz variables (x, y, z), resulting in a 6-dimensional system of ODEs. The complete set of equations governing the HPMR dataset is thus given by the combination of the PK equations equation 1 and the Lorenz equations equation 2, with the reactivity $\rho(t)$ defined as above. Due to the very different magnitude of the variables, the power in the state vector is represented in logarithmic scale, i.e., $\log n$; whereas, the precursors are normalized by their critical values $c_i / c_i^{\mathrm{crit}}(n)$.

A.2 DYNASTY: 1D SYSTEM CODE

DYNASTY (DYnamics of NAtural circulation for molten SalT internallY heated) is a natural circulation loop facility (Cammi et al., 2019) designed to study the dynamics induced by a distributed heat source. It has played an important role in nuclear research over the past years, being the focus of extensive stability analysis aimed at investigating instabilities in natural circulation loops, mimicking the operating conditions of the passive heat removal system of the European Molten Salt Fast Reactor. In the loop, heat is supplied by electric heating strips made of copper and encased in fibreglass, mounted on the two vertical legs and the bottom horizontal leg. The upper horizontal leg hosts a finned pipe, acting as a cooler: to enhance heat transfer, a fan is located below the cooler. This setup allows for various heating configurations to be set up.

The numerical framework adopted in this work is a 1D simulation of the fluid region using RELAP5/MOD3.3 (Fletcher & Schultz, 1992) system code, providing physically consistent, space- and time-resolved data. RELAP5 (R5) was originally developed at the Idaho National Engineering Laboratory for the U.S. Nuclear Regulatory Commission and is widely used for the simulation of transient thermal–hydraulic phenomena in nuclear and non-nuclear systems.
R5 is based on a one-dimensional, non-homogeneous, non-equilibrium two-fluid formulation, solved through a partially implicit numerical scheme. This modelling approach allows the code to capture the dominant physical mechanisms governing buoyancy-driven flows while remaining computationally efficient, a key requirement for generating large datasets suitable for reduced-order representations and data-driven architectures. The governing equations consist of separate conservation laws for mass, momentum, and energy for each phase, formulated over control volumes aligned along differential stream tubes. The dependent variables represent volume- and time-averaged quantities, while time and one spatial coordinate constitute the independent variables. A comprehensive description of the mathematical formulation implemented in R5 is reported in (Mangal et al., 2012). A uniform axial discretization is adopted, with a control volume length of 100 mm, resulting in a total of 122 computational nodes. Sensitivity analyses on numerical methods and nodalization were performed in (Missaglia et al., 2025). The state vector $\mathcal{V}$ is composed of 2 fields: the temperature which is spatially dependent and the mass flow rate which is spatially uniform, due to the incompressibility assumption.

A.3 MAGNETO-HYDRODYNAMICS: MULTI-PHYSICS APPROACH

Magneto-hydrodynamics describes the flow of an electrically conductive fluid subjected to the Lorentz force induced by the superposition of an externally imposed magnetic field and the self-induced one. This phenomenon is of particular interest in the nuclear field, as it is relevant for the design of fusion reactors, in particular in the external blanket. The model couples the Navier-Stokes equations for the fluid dynamics with the Maxwell equations for the electromagnetic field, resulting in a strongly coupled multi-physics problem. The fluid dynamics equations include also the energy equation and two additional terms associated with the Lorentz force and the Joule heating, which are responsible for the interaction between the fluid and the electromagnetic field.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{3a}$$

$$\frac{\partial (\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \rho \mathbf{g} + \frac{1}{\mu_0} (\nabla \times \mathbf{B}) \times \mathbf{B}, \tag{3b}$$

$$\frac{\partial (\rho c T)}{\partial t} + \nabla \cdot (\rho c T \mathbf{u}) = \kappa \Delta T + \frac{1}{\sigma \mu_0^2} |\nabla \times \mathbf{B}|^2, \tag{3c}$$

$$\rho = \rho_0 \left[1 - \beta (T - T_0)\right], \tag{3d}$$

$$\boldsymbol{\tau} = \mu \left[\nabla \mathbf{u} + (\nabla \mathbf{u})^T - \frac{2}{3} (\nabla \cdot \mathbf{u}) \mathbf{I} \right]. \tag{3e}$$

On the other hand, the Maxwell equations for the electromagnetic field include the induction equation and the solenoidal constraint for the magnetic field:

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{u} \times \mathbf{B}) + \frac{1}{\sigma \mu_0} \Delta \mathbf{B}, \tag{4a}$$

$$\nabla \cdot \mathbf{B} = 0. \tag{4b}$$

MHD equations are solved using a custom solver in OpenFOAM, called **magnetoHDFoam** (Lo Verso et al., 2024), for buoyant flows of an electrically conducting fluid, using the flexible PIMPLE algorithm for time-resolved and pseudo-transient simulations. The case study adopted in this work consists of a bi-dimensional channel, in which 2 cylinders are inserted. The fluid, flowing towards the positive $\hat{\mathbf{x}}$ coordinates, is a liquid metal, specifically Pb-17Li. An external magnetic field $\mathbf{B}$ directed upwards is applied to study the effect of the magnetic field intensity on the metal's fluid dynamics. The state vector $\mathcal{V}$ is composed of 4 fields (each with 30600 spatial degrees of freedom): two components of the velocity $\mathbf{u}$, the pressure p and the temperature T; the magnetic field $\mathbf{B}$ is imposed and not part of the state vector, since only weak spatial dynamics is observed in the magnetic field.

A.4 Molten Salt Fast Reactor: multi-physics approach

The Molten Salt Fast Reactor (Generation IV International Forum, 2014; Brovchenko et al., 2013) is an innovative reactor concept in which the fuel is liquid and homogeneously mixed with the molten salt thermal carrier; this configuration offers significant advantages, for example from the point of view of fuel reprocessing and closure of the fuel cycle, but also poses a lot of unique challenges in terms of monitoring and safety. Moreover, due to the strict coupling between the different physics, even more so than other nuclear reactor configurations, the mathematical modelling of such system requires suitable tools to integrate different physics together, starting from the neutronics and thermal-hydraulics (Aufiero et al., 2014; Cervi et al., 2019).
The governing equations of the MSFR consist of the Navier-Stokes equations with the energy balance, including turbulence modelling with RANS (Realizable $\kappa - \epsilon$ model) and the Boussinesq approximation for density variations, the multi-group neutron diffusion and the precursors advection-diffusion: for completeness, this section reports the governing equations and briefly discusses their coupling. The Navier-Stokes equations for the MSFR reads (Versteeg & Malalasekera, 2007):

$$\nabla \cdot \mathbf{u} = 0 \tag{5a}$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = + \nabla \cdot [\nu_{\text{eff}}(\nabla \mathbf{u} + (\nabla \mathbf{u})^T)] - \frac{1}{\rho}\nabla p + [1 - \beta_T(T - T_0)]\mathbf{g} \tag{5b}$$

$$\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{u}T) = \nabla \cdot (\alpha_{\text{eff}} \nabla T) + \frac{q'''}{\rho c_p} \tag{5c}$$

given $\mathbf{u}$ as the velocity, ρ as the density, p as the pressure, ν_{eff} as the effective kinematic viscosity (accounting for turbulence modelling), T as the temperature, β_T as the thermal expansion, $\mathbf{g}$ as the gravity acceleration, α_{eff} as the thermal diffusivity, q''' as the power density, which includes both neutron fission and delayed heat sources, and c_p as the specific heat capacity. This set of equations shows the strong coupling between the two different physics; in particular, the source term of the energy equation directly depends on the neutron fluxes, i.e.

$$q''' = q'''_{\text{prompt}} + q'''_{\text{decay}} = (1 - \beta_h) \sum_g \bar{E}_{f,g} \Sigma_{f,g} \phi_g + \sum_i \lambda_{h,i} d_i \tag{6}$$

with ϕ_g being the neutron flux of energy group g, β_h being the fraction of delayed heat source, $\bar{E}_{f,g}$ being the average energy released by fission in group g, $\Sigma_{f,g}$ being the fission cross section of group g, $\lambda_{h,i}$ being the decay constant of delayed decay group d_i. Neutronics is governed by the Multi-Group Diffusion Equation (Duderstadt & Hamilton, 1976; Aufiero et al., 2014):

$$\frac{1}{v_g}\frac{\partial \phi_g}{\partial t} - \nabla \cdot (D_{n,g} \nabla \phi_g) + \Sigma_{r,g} \phi_g = S_g \tag{7}$$

with $D_{n,g}$ being the neutron diffusion coefficient of group g and $\Sigma_{r,g}$ being the removal cross-section accounting for all the reactions which decrease the number of neutrons of group g, i.e. absorption

$\Sigma_{a,g}$ and infra-group scattering $\Sigma_{s,g\to g'}$:

$$\Sigma^0_{r,g} = \left(\Sigma^0_{a,g} + \sum_{g' \neq g} \Sigma^0_{s,g\to g'} \right) \tag{8}$$

where the superscript 0 refers to the values at the reference temperature. In fact, the cross section are strongly affected by the neutron energy due to the Doppler broadening effect and the variation in density, i.e.

$$\Sigma_{r,g} = \left(\Sigma^0_{r,g} + A^0_{r,g} \ln \frac{T}{T^\Sigma_0} \right) \cdot \frac{1 - \beta_T (T - T_0)}{1 - \beta_T (T^\Sigma_0 - T_0)} \tag{9}$$

In the end, the neutron diffusion equations have a source term S_g accounting for the production of neutrons in the group energy g due to prompt fission, decay of precursors and infra-group scattering:

$$S_g = (1 - \beta)\chi_{p,g}\, S_{p,g} + \chi_{d,g}\, S_d + S_{s,g} \tag{10a}$$

$$S_{p,g} = \sum_{g'} \bar{\nu}_{g'} \Sigma_{f,g'} \phi_{g'} \tag{10b}$$

$$S_d = \sum_k \lambda_k c_k \tag{10c}$$

$$S_{s,g} = \sum_{g' \neq g} \Sigma_{s,g'\to g} \phi_{g'} \tag{10d}$$

with $\chi_{p,g}$ and $\chi_{d,g}$ as the fission spectrum of prompt and delayed neutrons and $\bar{\nu}_{g'}$ as the average number of neutrons produced in group g'.

When dealing with transient scenarios, the production of neutrons is split into prompt (due to fission) and delayed (due to decay): in the nuclear reactors world, this latter contribution is described by the group precursors c_k which are governed by an advection-diffusion equation in the salt:

$$\begin{aligned} \frac{\partial c_k}{\partial t} + \nabla \cdot (\mathbf{u} c_k) = &+ \nabla \cdot (D_{\text{eff}} \nabla c_k) - \lambda_k c_k \\ &+ \beta_{d,k} \sum_g \bar{\nu}_g \Sigma_{f,g} \phi_g \end{aligned} \tag{11}$$

with D_{eff} the effective diffusion coefficient and λ_k the decay constant of delayed neutron precursor group k. In the end, the precursors contribute also to the energy equation equation 5c, where they are grouped together into the decay heat group d_i

$$\begin{aligned} \frac{\partial d_i}{\partial t} + \nabla \cdot (\mathbf{u} d_i) = &+ \nabla \cdot (D_{\text{eff}} \nabla d_i) - \lambda_{h,i} d_i \\ &+ \beta_{h,l} \sum_g \bar{E}_{f,g} \Sigma_{f,g} \phi_g \end{aligned} \tag{12}$$

The amount of coupled equations can reach about 20 according to the number of energy groups and delayed neutron precursors considered. For this work, the full-order state space vector $\mathcal{V}$ would then include more than 20 different coupled fields. In order to keep the dimension of the state space vector $\mathcal{V}$ manageable, we selected a subset of the fields of particular interest to fully describe the dynamics of the system: $q'''_{\text{prompt}}, q'''_{\text{decay}}, T, u_x, u_z$, being the power density related to prompt and delayed neutrons, the temperature and two velocity components. Among all possible accidental transients, to minimise predictability complex functions were applied to the momentum source (pump velocity) and to the heat sink (heat transfer coefficient and external temperature), starting from steady-state conditions.

B Standard Set of Metrics

Throughout the benchmark, each stored trajectory is a matrix $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ whose *rows* index uniform time samples and whose *cols* index the (possibly stacked) state components, consistent with

the evaluation module of the `ctf4science` package. Subscripts such as $\boldsymbol{X}_{j\text{train}}$, $\boldsymbol{X}_{j\text{test}}$, and $\boldsymbol{X}_{j\text{pred}}$ refer to the j-th dataset file in the catalogue (e.g. `X3test.npz`), not necessarily to task pair j. Pairs 2–5 reuse the medium- and high-noise training files $\boldsymbol{X}_{2\text{train}}$ and $\boldsymbol{X}_{3\text{train}}$ for both denoising and noisy forecasting; pairs 6–7 use the limited-data files $\boldsymbol{X}_{4\text{train}}$ and $\boldsymbol{X}_{5\text{train}}$; pairs 8–9 use three full-length parametric training trajectories $\boldsymbol{X}_{6\text{train}}, \boldsymbol{X}_{7\text{train}}, \boldsymbol{X}_{8\text{train}}$ together with burn-in initial segments $\boldsymbol{X}_{9\text{train}}$ and $\boldsymbol{X}_{10\text{train}}$.

B.1 Forecasting (2 scores)

The first set of tasks involves the approximation of the future state of the system. Thus, given a training matrix representing the dynamics over $t \in [0, T]$ ($\boldsymbol{X}_{1\text{train}} \in \mathbb{R}^{m\times n}$), a generated forecast is requested from the model being tested for $t \in [T, T']$ ($\boldsymbol{X}_{1\text{pred}} \in \mathbb{R}^{m'\times n}$) to compare with the ground truth ($\boldsymbol{X}_{1\text{test}} \in \mathbb{R}^{m'\times n}$), with n the state dimension and m, m' the numbers of training and forecast time steps. Relative errors S_{ST} and S_{LT} are mapped to benchmark scores E_1 and E_2: S_{ST} is a normalized root-mean-square error over an initial time window, and S_{LT} compares log power spectra after a short-time burn-in. Short-time forecasting measures trajectory accuracy where deterministic prediction is feasible, while long-time forecasting measures statistical fidelity where only the system's broad statistical properties are recoverable.

For the challenge dynamics of interest, sensitivity to initial conditions is common, making long-range pointwise forecasting an unreasonable task, given fundamental limitations set by Lyapunov times. The long-time error uses the log power spectrum $\mathcal{P}(\boldsymbol{X}, \boldsymbol{\kappa}, k_{\text{short}}) = \ln\big(|\text{FFT}(\boldsymbol{X}[k_{\text{short}}{:}-1, \boldsymbol{\kappa}])|^2\big)$, where `fftshift` places the data in the wavenumber domain, $\boldsymbol{\kappa} = n/2 - \kappa_{\max}{:}\, n/2 + (\kappa_{\max} + 1)$ with $\kappa_{\max} = 100$, and k_{short} is the number of initial rows discarded before spectral comparison (dataset-specific). Let $\hat{\boldsymbol{X}}$ denote ground truth, $\tilde{\boldsymbol{X}}$ the prediction, and k_{short} the number of initial time rows used for the short-time score. The relative errors are

$$S_{\text{ST}}(\tilde{\boldsymbol{X}}, \hat{\boldsymbol{X}}) = \frac{\|\hat{\boldsymbol{X}}[1{:}k_{\text{short}}, :] - \tilde{\boldsymbol{X}}[1{:}k_{\text{short}}, :]\|}{\|\hat{\boldsymbol{X}}[1{:}k_{\text{short}}, :]\|}, \tag{13}$$

$$S_{\text{LT}}(\tilde{\boldsymbol{X}}, \hat{\boldsymbol{X}}) = \frac{\|\mathcal{P}(\hat{\boldsymbol{X}}, \boldsymbol{\kappa}, k_{\text{short}}) - \mathcal{P}(\tilde{\boldsymbol{X}}, \boldsymbol{\kappa}, k_{\text{short}})\|}{\|\mathcal{P}(\hat{\boldsymbol{X}}, \boldsymbol{\kappa}, k_{\text{short}})\|}. \tag{14}$$

It is clear that there are many ways to evaluate the long-range forecasting capabilities. We followed in the footsteps of (Wyder et al., 2026; Yermakov et al., 2026) and chose a simple and transparent metric, fully understanding that more nuanced scoring could be used. To provide a reasonable range, we then compute the two scores

$$E_1 = 100\big(1 - S_{\text{ST}}(\boldsymbol{X}_{1\text{pred}}, \boldsymbol{X}_{1\text{test}})\big), \quad E_2 = 100\big(1 - S_{\text{LT}}(\boldsymbol{X}_{1\text{pred}}, \boldsymbol{X}_{1\text{test}})\big), \tag{15}$$

meaning in each case a score of $E_i = 100$ corresponds to a perfect match. Note that, as a baseline, a solution guess of zeros $\tilde{\boldsymbol{X}}[1{:}k_{\text{short}}, :] = \mathbf{0}$ (corresponding also to $\mathcal{P}(\tilde{\boldsymbol{X}}, \boldsymbol{\kappa}, k_{\text{short}}) = \mathbf{0}$) gives $E_1 = E_2 = 0$.

Input: $\boldsymbol{X}_{1\text{train}} \in \mathbb{R}^{m\times n}$; **Output:** $\boldsymbol{X}_{1\text{pred}} \in \mathbb{R}^{m'\times n}$; **Scores:** E_1, E_2.

B.2 Noisy Data (4 scores)

The ability to handle noise is critical in all data-driven applications as sensors and measurement technologies are by default embedded with varying levels of noise. Methods that work with numerically accurate data, for example, data points that are 10^{-6} accurate, may be useful for model reduction, but are rarely suitable for discovery and engineering design from real-world data. Both strong and weak noise are considered, as these represent realistic challenges to be addressed in practice.

This task is very similar to the forecasting described above, but now with noise added to the data. Specifically, the model is provided $\boldsymbol{X}_{2\text{train}}, \boldsymbol{X}_{3\text{train}} \in \mathbb{R}^{m\times n}$ representing medium- and high-noise evolutions, respectively (pairs 2–3 and 4–5). The objective is first to reconstruct the clean reference on the training window ($\boldsymbol{X}_{2\text{pred}}, \boldsymbol{X}_{4\text{pred}} \in \mathbb{R}^{m\times n}$, scored against $\boldsymbol{X}_{2\text{test}}$ and $\boldsymbol{X}_{4\text{test}}$), and second to forecast the held-out future ($\boldsymbol{X}_{3\text{pred}}, \boldsymbol{X}_{5\text{pred}} \in \mathbb{R}^{m'\times n}$, scored against $\boldsymbol{X}_{3\text{test}}$ and $\boldsymbol{X}_{5\text{test}}$) using

the same noisy training files. Reconstruction uses S_{ST}; noisy forecasting pairs 3 and 5 use S_{LT} (long-time statistical score), leading to

$$E_3 = 100\big(1 - S_{\mathrm{ST}}(\boldsymbol{X}_{2\mathrm{pred}}, \boldsymbol{X}_{2\mathrm{test}})\big), \quad E_4 = 100\big(1 - S_{\mathrm{LT}}(\boldsymbol{X}_{3\mathrm{pred}}, \boldsymbol{X}_{3\mathrm{test}})\big),$$
$$E_5 = 100\big(1 - S_{\mathrm{ST}}(\boldsymbol{X}_{4\mathrm{pred}}, \boldsymbol{X}_{4\mathrm{test}})\big), \quad E_6 = 100\big(1 - S_{\mathrm{LT}}(\boldsymbol{X}_{5\mathrm{pred}}, \boldsymbol{X}_{5\mathrm{test}})\big).$$

Input: $\boldsymbol{X}_{2\mathrm{train}}, \boldsymbol{X}_{3\mathrm{train}} \in \mathbb{R}^{m\times n}$;
Output: $\boldsymbol{X}_{2\mathrm{pred}}, \boldsymbol{X}_{4\mathrm{pred}} \in \mathbb{R}^{m\times n}$, $\boldsymbol{X}_{3\mathrm{pred}}, \boldsymbol{X}_{5\mathrm{pred}} \in \mathbb{R}^{m'\times n}$; **Scores:** E_3, E_4, E_5, E_6.

B.3 Limited Data (4 scores)

Data limitations are common in real-world physical systems and often affect the success of data-driven methods. Thus, testing for model performance on low data is critically important and provides important insight for potential users.

In this case, only a limited number of snapshots N_{snap} of numerically accurate data are given ($\boldsymbol{X}_{4\mathrm{train}} \in \mathbb{R}^{N_{\mathrm{snap}}\times n}$, pair 6). From this limited data, a forecast must be made and evaluated with equation 13 and equation 14 on the approximated future $\boldsymbol{X}_{6\mathrm{pred}} \in \mathbb{R}^{m'\times n}$ against $\boldsymbol{X}_{6\mathrm{test}}$. The experiment is repeated with noise on the measurements ($\boldsymbol{X}_{5\mathrm{train}} \in \mathbb{R}^{N_{\mathrm{snap}}\times n}$, pair 7), producing $\boldsymbol{X}_{7\mathrm{pred}} \in \mathbb{R}^{m'\times n}$ scored against $\boldsymbol{X}_{7\mathrm{test}}$.

$$E_7 = 100\big(1 - S_{\mathrm{ST}}(\boldsymbol{X}_{6\mathrm{pred}}, \boldsymbol{X}_{6\mathrm{test}})\big), \quad E_8 = 100\big(1 - S_{\mathrm{LT}}(\boldsymbol{X}_{6\mathrm{pred}}, \boldsymbol{X}_{6\mathrm{test}})\big),$$
$$E_9 = 100\big(1 - S_{\mathrm{ST}}(\boldsymbol{X}_{7\mathrm{pred}}, \boldsymbol{X}_{7\mathrm{test}})\big), \quad E_{10} = 100\big(1 - S_{\mathrm{LT}}(\boldsymbol{X}_{7\mathrm{pred}}, \boldsymbol{X}_{7\mathrm{test}})\big).$$

Two error scores (analogous to E_1 and E_2) are produced for the noise-free and noisy limited data. These scores are E_7 (short-time forecast) and E_8 (long-time forecast) for the noise free case and E_9 (short-time forecast) and E_{10} (long-time forecast) for the noisy case.

Input: $\boldsymbol{X}_{4\mathrm{train}}, \boldsymbol{X}_{5\mathrm{train}} \in \mathbb{R}^{N_{\mathrm{snap}}\times n}$; **Output:** $\boldsymbol{X}_{6\mathrm{pred}}, \boldsymbol{X}_{7\mathrm{pred}} \in \mathbb{R}^{m'\times n}$; **Scores:** E_7, E_8, E_9, E_{10}.

B.4 Parametric Generalization (2 scores)

Finally, the ability of a model to generalize to different parameter values is evaluated. For this case, the model's ability to interpolate and extrapolate to new parameter regimes is considered with noise-free data. The interpolation and extrapolation are each their own score, resulting in two scores that evaluate parametric dependence.

Three training datasets are provided at three different (unknown) parameter values ($\boldsymbol{X}_{6\mathrm{train}}, \boldsymbol{X}_{7\mathrm{train}}, \boldsymbol{X}_{8\mathrm{train}} \in \mathbb{R}^{m\times n}$). Construction of the dynamics in interpolatory and extrapolatory regimes is required ($\boldsymbol{X}_{8\mathrm{pred}}, \boldsymbol{X}_{9\mathrm{pred}} \in \mathbb{R}^{m'\times n}$, pairs 8 and 9). For each task, a burn-in segment $\boldsymbol{X}_{9\mathrm{train}}$ (interpolation) or $\boldsymbol{X}_{10\mathrm{train}}$ (extrapolation) of size $N_{\mathrm{snap}} \times n$ ($N_{\mathrm{snap}} \leq m$) initializes the forecast; performance is evaluated with equation 13.

$$E_{11} = 100\big(1 - S_{\mathrm{ST}}(\boldsymbol{X}_{8\mathrm{pred}}, \boldsymbol{X}_{8\mathrm{test}})\big), \quad E_{12} = 100\big(1 - S_{\mathrm{ST}}(\boldsymbol{X}_{9\mathrm{pred}}, \boldsymbol{X}_{9\mathrm{test}})\big).$$

Input: $\boldsymbol{X}_{6\mathrm{train}}, \boldsymbol{X}_{7\mathrm{train}}, \boldsymbol{X}_{8\mathrm{train}} \in \mathbb{R}^{m\times n}$; burn-in $\boldsymbol{X}_{9\mathrm{train}}, \boldsymbol{X}_{10\mathrm{train}} \in \mathbb{R}^{N_{\mathrm{snap}}\times n}$;

Output: $\boldsymbol{X}_{8\mathrm{pred}}, \boldsymbol{X}_{9\mathrm{pred}} \in \mathbb{R}^{m'\times n}$; **Scores:** E_{11}, E_{12}.

B.5 Composite Score

We compute a composite score ($AvgScore$) per dataset from metrics E_1 through E_{12} by averaging the resulting scores for each method. This score is evaluated per method, not per model. Thus, each method can fit a model for each task and produce the best possible score. All scores are clipped such that $E_i \in [-100, 100]$, thus $AvgScore \in [-100, 100]$. Methods that cannot produce a result for a given task receive the minimum score -100.

C Additional Evaluation Results

This section reports additional evaluation results for the methods in the benchmark: in particular, we report the sensitivity analysis on the sparse sensor reconstruction task and the baseline results on the other datasets.

C.1 Sensitivity Analysis for Sparse Reconstruction with Empirical Interpolation Method

In this section, we present a preliminary sensitivity analysis of the state estimation problem, introduced as a new task in the CTF4Nuclear: in particular, we will focus on a baseline method, i.e., the Empirical Interpolation Method (EIM) (Maday, 2006; Maday et al., 2015). As noted in Section 2, the task is defined as follows: given a set of sparse measurements of a physical field, the goal is to reconstruct the full state of the system. The quality of the reconstruction is evaluated using the standard CTF metrics E_1–E_{12}, as described in section B. For each score the input and output of the task can be summarized as follows:

Input: $\mathbf{X}_{\text{train}} = [\mathbf{X}^{(1)}_{\text{train}}, \mathbf{X}^{(2)}_{\text{train}}, ..., \mathbf{X}^{(p)}_{\text{train}}] \in \mathbb{R}^{m \times n}, \mathbf{y}^{(1)}_{\text{train}} \in \mathbb{R}^{m \times s}, \mathbf{y}^{(1)}_{\text{test}} \in \mathbb{R}^{m' \times s}, s \ll n;$

Output: $\mathbf{X}_{\text{pred}} = [\mathbf{X}^{(1)}_{\text{pred}}, \mathbf{X}^{(2)}_{\text{pred}}, ..., \mathbf{X}^{(p)}_{\text{pred}}] \in \mathbb{R}^{m' \times n}.$

C.1.1 Empirical Interpolation Method

The Empirical Interpolation Method (EIM) (Maday, 2006) is a model reduction technique that allows for the approximation of a function $u(\mathbf{x}; \boldsymbol{\mu})$ defined over a spatial domain Ω and parameter space $\mathcal{D}$, using a reduced basis representation. The method constructs a set of basis functions $\{q_m\}_{m=1}^M$ (magic functions) and a set of interpolation points $\{\mathbf{x}_m\}_{m=1}^M$ (magic points) such that the function can be approximated as:

$$u(\mathbf{x}; \boldsymbol{\mu}) \simeq \mathcal{I}_M[u](\mathbf{x}; \boldsymbol{\mu}) = \sum_{m=1}^{M} \beta_m(\boldsymbol{\mu}) \cdot q_m(\mathbf{x}) \tag{16}$$

where the spatial behaviour is captured by the magic functions $\{q_m\}_{m=1}^M$ and the parametric dependence by the coefficients $\{\beta_m\}_{m=1}^M$, and where M is the dimension of the reduced space and $\boldsymbol{\mu}$ is the parameters vector. The coefficients $\{\beta_m\}_{m=1}^M$ are computed by enforcing the interpolation conditions at the magic points $\{\mathbf{x}_m\}_{m=1}^M$:

$$\mathcal{I}_M[u](\mathbf{x}_m; \boldsymbol{\mu}) = u(\mathbf{x}_m; \boldsymbol{\mu}), \qquad m = 1, \ldots, M. \tag{17}$$

The EIM uses an iterative greedy procedure which at every time step builds a reduced space. The offline procedure takes as inputs the available snapshots/training data and returns the set of magic functions and magic points $\{v_m\}_{m=1}^M$ by minimizing at every time step the interpolation error. As new measurements become available, the EIM can be used to reconstruct the full state of the system by solving a linear system for the coefficients $\{\beta_m\}_{m=1}^M$ and then evaluating the interpolant $\mathcal{I}_M[u](\mathbf{x}; \boldsymbol{\mu})$ at any point in the domain.

This algorithm is particularly useful because it allows for both the selection of the most informative measurement locations (magic points) and the construction of a reduced basis that captures the essential features of the system's dynamics. We have decided to use the EIM as a baseline method for the sparse reconstruction task, as it is a physics-aware method and has been widely used in the context of model reduction and state estimation, especially in the context of nuclear systems (Gong et al., 2016; Cammi et al., 2024).

C.1.2 Numerical Results and Discussion

We assess how the quality of full-state reconstruction from sparse measurements depends on the number of EIM sensors s, the observed physical field, and the candidate locations from which magic points are selected. The analysis is carried out on the MSFR dataset. The state vector comprises five coupled fields: prompt and decay power densities, temperature (T) and velocity components. We have set the maximum number of available sensors to choose to be $s_{\max} = 100$. Another important

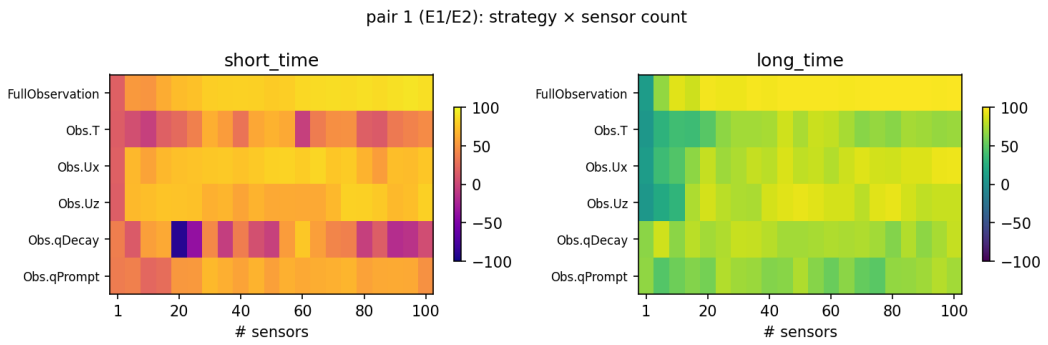


Figure 4: Scores E_1 and E_2 as a function of the number of sensors s (horizontal axis) and of the observation strategy (vertical axis). The two panels show the short-time (E_1, left) and long-time (E_2, right) metrics. Two effects are visible: scores saturate beyond $s \approx 20$, and the choice of observed field matters more than the sensor count: for instance, measuring q_{decay} worsens the performance for short-time forecasting while measuring U_z or U_x is not, at identical s.

aspect to consider is the observation strategy, which defines the library of candidate locations from which EIM selects the magic points. We consider three different strategies: (i) full observation, where all degrees of freedom of all fields are admissible; (ii) dense single-field observation, where candidates are restricted to the nodes of one physical field; and (iii) random subsampling from the full batch or from a single field only.

Dependence on the number of sensors and the observable field. As expected, the quality of the reconstruction improves with the number of sensors s, especially if the full state is measured; on the other hand, if the library of points is restricted to a single field, the performance is strongly dependent on the choice made. Figure 4 reports, as an example, the scores E_1 and E_2 depending on the number of sensors and the observation strategy. If the state is fully observable, the EIM can select the most informative locations across all fields and the inverse problem is well-posed. A saturation around $s \sim 20$ is observed, highighting the limitations of a linear reconstruction method in a high-dimensional system. Conversely, the choice of the observable field is critical: for instance, measuring the decay power density or the temperature field can lead to poor reconstruction performance, while measuring the velocity components or the prompt power density is more informative. The reason for this is that the velocity and prompt power fields carry the fastest dynamics of the system, while the decay power and temperature fields can be seen as delayed responses to the fast dynamics and therefore are less informative for short-time forecasting.
Furthermore, we want to stress that the choice of the observable field generally has a stronger impact on the reconstruction performance than the number of sensors. This is informative for nuclear reactor operators, as it suggests that the choice of the physical quantity to measure is more important than the number of sensors deployed. In other words, a poor choice of observable field cannot be compensated by adding more sensors. For real world applications, we would recommend to select multiple physical quantities to measure, if possible, to improve the reconstruction performance.

Structured versus unstructured placement. Comparing dense single-field observation against random subsampling of the mesh at $s \approx 20$ shows that structured, field-constrained observation wins decisively on most tasks, as expected: on pair 1 (long-time) observing U_z outperforms random full-mesh sensing by $\Delta S = +32.2$ (given S as the score), and on parametric interpolation (pair 8) observing T exceeds it by $\Delta S = +40.4$. Random placement is occasionally competitive on limited-data short-time forecasting (pair 6, $\Delta S = -3.6$) but collapses on the corresponding long-time metric ($\Delta S = +35.5$). Pair 9 is the notable exception, where random full-mesh sensing ($S = 61.0$) surpasses the best single-field strategy ($S = 48.6$): when no single field is informative enough, unstructured spatial diversity can partially compensate. Taken together, these comparisons indicate that sensor placement and field selection are not separable design choices, and that the appropriate strategy depends on the task.

Table 2: Reference baseline scores on the three datasets not evaluated in the main text.

			Forecasting		Noisy Data				Limited Data				Param. Gen.	
Dataset	**Baseline**	**Avg**	**E1**	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**	**E8**	**E9**	**E10**	**E11**	**E12**
HPMR	Average	−35.58	0.00	−93.07	0.00	−89.60	0.00	−92.33	0.00	−96.40	0.10	−95.73	24.98	15.09
	Zeros	−38.93	0.00	−93.07	0.00	−89.60	0.00	−92.33	0.00	−96.40	0.00	−95.73	0.00	0.00
	Last	−49.03	−38.84	−93.87	−100.00	−95.00	−0.03	−93.13	11.30	−95.40	−4.47	−95.80	−33.00	49.94
DYNASTY	Last	80.82	86.26	89.41	71.65	90.09	71.35	88.82	89.27	65.82	89.26	65.81	84.89	77.26
	Average	77.17	80.39	66.90	77.35	68.41	77.35	68.39	90.84	67.55	90.85	67.57	88.71	81.69
	Zeros	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
MHD	Zeros	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Average	−14.81	0.00	0.12	0.00	0.02	0.00	0.02	0.00	−100.00	0.00	−100.00	22.13	0.00
	Last	−26.86	−64.99	43.85	−75.14	14.81	−77.69	14.55	−73.11	37.21	−74.31	42.42	−34.45	−75.49

Discussion and scope. These results quantify a central practical constraint for state estimation: sparse-sensor reconstruction is highly sensitive to what is measured at least as much as to how many sensors are deployed. For MSFR, velocity or temperature observations at $s \approx 10$–20 often reach $S = 70$–94 on the forecasting, noisy and interpolation tasks, whereas unstructured random sensing typically stagnates near $S \approx 50$–60. The present study is intentionally scoped: EIM is used as a transparent, physics-aware reference without learned sensor placement or cross-dataset transfer, and its scores should be read as a reference level for data-driven methods evaluated under the same observation protocol.

Since the score depends on the observation design as strongly as on the estimator, that design must be part of the task specification rather than a choice left to each participant; otherwise a difference between two entries cannot be attributed to the reconstruction algorithm rather than to a more favourable choice of measured field. For future developments, we therefore suggest fixing three elements. First, the observable field, with separate tracks for the quantities that are actually instrumentable: temperature and power density are easy to measure in real systems, whereas local velocity is generally not. Second, the number of sensors, for which the saturation around $s \approx 20$ suggests that $s \in \{5, 10, 20\}$ suffices to characterise a method. Third, whether placement is prescribed or optimised by the method, as selecting one's own locations solves an experimental-design problem in addition to a reconstruction one and should be reported as a separate track.

C.2 Reference Baselines on the Remaining Datasets

Table 2 reports the three reference baselines of section D on the `pke-hpmr`, `dynasty` and `mhd-2cyls` datasets, provided as calibration points for future entries rather than as a method comparison. They already illustrate how strongly the difficulty profile varies across the suite: the baseline-last reaches an average of 80.82 on `dynasty`, where the approach to the natural-circulation pseudo-steady state is near-monotonic, but −26.86 on `mhd-2cyls`, whose magneto-convective transients are considerably less predictable, and −49.03 on `pke-hpmr`, where the Lorenz-driven reactivity feedback renders the dynamics genuinely chaotic. These simple baselines are already informative on the necessity of a complete evaluation suite: the nuclear industry is still adopting lumped and system codes to design, monitor and control nuclear power plants, being multi-physics modelling more widespread among the research community; this is also one of the reasons why this work has been developed and collected a diverse set of datasets, to provide a benchmark for the community to test and compare different methods on a common ground.

D Machine Learning Models

D.1 Baselines

As already mentioned in the work, we adopt three baselines. None of them learns a temporal surrogate: each predicts from a single elementary statistic of the training window and has no hyperparameters, so no tuning is performed. Let $\boldsymbol{X}_{\text{train}} \in \mathbb{R}^{m \times n}$ be the training matrix in the row-oriented

layout of section B and m' the number of requested prediction steps. For $j = 1, \ldots, m'$:

$$\text{Zeros:} \quad \boldsymbol{X}_{\text{pred}}[j,:] = \mathbf{0}, \tag{18}$$

$$\text{Average:} \quad \boldsymbol{X}_{\text{pred}}[j,:] = \tfrac{1}{m} \textstyle\sum_{k=1}^{m} \boldsymbol{X}_{\text{train}}[k,:], \tag{19}$$

$$\text{Last:} \quad \boldsymbol{X}_{\text{pred}}[j,:] = \boldsymbol{X}_{\text{train}}[m,:]. \tag{20}$$

For the parametric tasks (pairs 8–9) the burn-in segment replaces $\boldsymbol{X}_{\text{train}}$, so that Last persists the state immediately preceding the forecast. Baseline Zeros fixes the origin of the scoring scale: a zero prediction gives a relative error of exactly one in both equation 13 and equation 14, hence $E_i = 0$ on every task. Baseline Last (persistence) is the strongest of the three and the default reference in engineering practice; its score measures how slowly a dataset evolves relative to its evaluation horizon, and it varies widely across the suite (Table 2). None of the baselines can fail in the sense of section B, so they are never assigned the -100 sentinel.

D.2 Dynamic Mode Decomposition

The Dynamic Mode Decomposition (DMD) is a data-driven method developed by Schmid (Schmid, 2010) in the fluid dynamics community to identify spatio-temporal coherent structures from high-dimensional data. The DMD algorithm is based on the Singular Value Decomposition (SVD) of a data matrix; in particular, DMD is able to provide a modal decomposition where each mode consists of spatially correlated structures that have the same linear behaviour in time. The DMD method is found to have a significant connection with the Koopman operator theory (Rowley et al., 2009): in particular, the DMD can be formulated as an algorithm able to learn the best-fit linear dynamical system to advance in time (Figure 5).

There are many variants of DMD, connected to existing techniques from system identification and modal extraction (Brunton & Kutz, 2022). Here, we will provide a brief overview of the underlying idea of the original DMD algorithm, from which all the other variants can be derived. The first step is to collect a set of snapshots of the system at different time steps. Following the Schmid convention, the snapshot matrix stacks spatial states in columns, $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{N_t}] \in \mathbb{C}^{\mathcal{N}_h \times N_t}$, where $\boldsymbol{x}_j \in \mathbb{C}^{\mathcal{N}_h}$ is the state at time t_j and N_t is the number of snapshots (contrast the row-oriented CTF layout $\boldsymbol{X} \in \mathbb{R}^{m \times n}$ in section B). The original formulation from (Schmid, 2010; Rowley et al., 2009) supposed uniform sampling in time, i.e. $t_j = j\Delta t$. Overall, the DMD algorithm seeks the leading spectral decomposition of the best-fit linear operator $\boldsymbol{A} \in \mathbb{C}^{\mathcal{N}_h \times \mathcal{N}_h}$ that advances the system in time, i.e.

$$\boldsymbol{x}_{j+1} \approx \boldsymbol{A}\boldsymbol{x}_j \quad \longleftrightarrow \quad \boldsymbol{X}_{2:N_t} \approx \boldsymbol{A}\boldsymbol{X}_{1:N_t-1}, \tag{21}$$

where $\boldsymbol{X}_{1:N_t-1}$ and $\boldsymbol{X}_{2:N_t}$ denote the first $N_t - 1$ and last $N_t - 1$ snapshot columns of $\boldsymbol{X}$, respectively. As we said above, the DMD algorithm is based on the SVD of the data matrix $\boldsymbol{X}$ of rank r, which can be written as $\boldsymbol{X} \simeq \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^*$: $\boldsymbol{U} \in \mathbb{C}^{\mathcal{N}_h \times r}$ represents the left singular vectors (spatial modes); $\boldsymbol{\Sigma} \in \mathbb{R}^{r \times r}$ contains the singular values; and $\boldsymbol{V}^* \in \mathbb{C}^{r \times N_t}$ represents the temporal coefficients. This compression allows one to compute $\boldsymbol{A}$ efficiently (J. Nathan Kutz et al., 2016; Brunton & Kutz, 2022), avoiding the direct inversion of the high-dimensional snapshot matrix.

Indeed, in the literature different variants of DMD have been proposed: in this context, the High-Order DMD (HODMD) (Le Clainche & Vega, 2017), which exploits time delay embedding to fit the optimal Koopman Operator, and the Optimised DMD (OptDMD) (Askham & Kutz, 2018; Sashidhar & Kutz, 2022), which is a variant of DMD that can also use the Bagging algorithm to improve the robustness of the DMD algorithm against noise. This latter variant has been shown to be the most robust and stable algorithm for real-world applications (Faraji et al., 2024). The implementation of the DMD algorithm is available in the `pyDMD` package (Demo et al., 2018; Ichinaga et al., 2024), which is a Python library for DMD and its variants. The library is designed to be easy to use and flexible, allowing users to customise the algorithm for their specific needs.

Parametric DMD The extension of DMD to parametric systems is a recent development in the field of system identification. Different approaches have been proposed in the literature; in this work, the implementation of Andreuzzi et al. (Andreuzzi et al., 2023) within `pyDMD` is adopted. Up to now, the package does not support the OptDMD algorithm directly, we have implemented a wrapper to use the OptDMD algorithm with the parametric DMD following the same approach of the

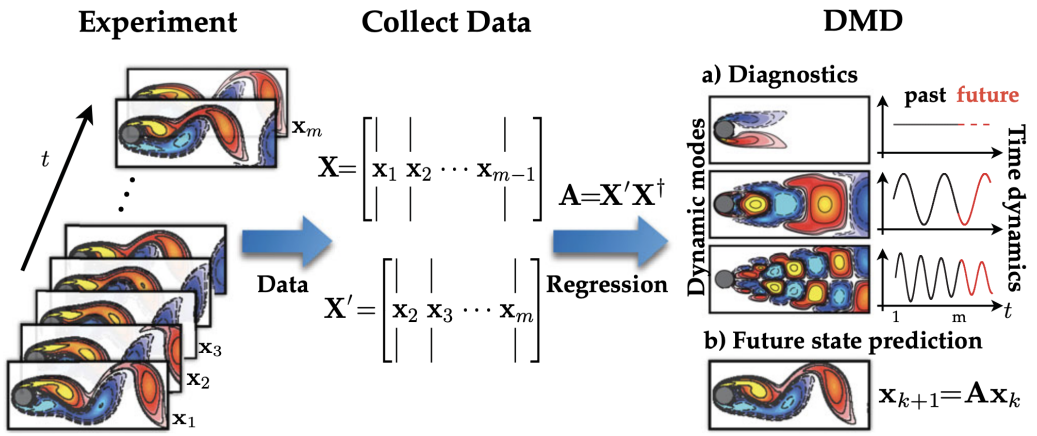


Figure 5: Scheme of the Dynamic Mode Decomposition algorithm from (J. Nathan Kutz et al., 2016). The data matrix $\boldsymbol{X}$ is constructed by stacking the snapshots in columns. The SVD of the data matrix is computed, and the dynamical matrix is fitted to the data. This allows us to compute the state of the system for future time instances.

hyperparameter	type	min (or options)	max (or none)
rank	randint	3	50
delay	randint	0	100
parametric	choice	{ "partitioned", "monolithic"}	
eig_constraints	choice	{ "none", "stable", "conjugate_pairs"}	

Table 3: Hyperparameter search space for the OptDMD algorithm (the parametric hyperparameter has an effect only for metrics E_{11} and E_{12}).

package, based on the interpolation of the forecasted reduced dynamics. We appreciate that further work and rigorous testing of this implementation are planned for future work. Similar to SINDy, since the parameter values employed for data generation are not publicly available, fictitious values mimicking the interpolatory and extrapolatory regimes have been used.

Hyperparameter tuning The hyperparameters of the DMD algorithm depend on the specific variant adopted. Every DMD algorithm has a set of hyperparameters that can be tuned to improve the performance of the algorithm; however, the rank of the SVD is common to all of them and plays a crucial role in the reduction process. The HODMD algorithm also includes the delay embedding, defining the size of the lagging window to use. The OptDMD algorithm can also put constraints on the DMD eigenvalues to ensure that the dynamics follow a certain behaviour. In the end, the parametric DMD can operate in two different modes: partitioned and monolithic. The hyperparameters of both DMD algorithms are listed in Table 3.

D.3 Koopman operator-based dynamic system prediction

The Koopman operator Koopman operator theory is a useful tool that has found increasing attention in the data-driven scientific computing community and can essentially be seen as an extension of dynamic mode decomposition - viewing the statespace of the dynamic system through the lens of nonlinear observables. This point-of-view dates back to early work by (Koopman, 1931; Koopman & von Neumann, 1932) and a modern review can be found in (Brunton et al., 2022). We outline the method briefly before describing the set-up for the chosen implementation and our testing on the CTF. Consider a dynamical system (either an ODE or a semi-discretisiation of a PDE) of the form:

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}), \tag{22}$$

where $\mathbf{f} : \mathbb{R}^N \to \mathbb{R}^N$ may be a nonlinear forcing. The central idea in Koopman operator theory is then to learn a coordinate transform (i.e. a set of nonlinear observables) $\Phi : \mathbb{R}^N \to \mathbb{R}^M$, under which the dynamics becomes (approximately) linear, i.e.

$$\frac{d\mathbf{z}}{dt} \approx \mathbf{A}\mathbf{z}, \quad \mathbf{z}(t) = \Phi(\mathbf{x}(t)). \tag{23}$$

In this new coordinate system, the exact solution of the linear dynamics is straightforward. The inference of $\mathbf{\Phi}$ and $\mathbf{A}$ can be formulated as a regression problem.

Numerical implementation and parameter choices In our current CTF test we use the `PyKoopman` Python library as the main reference point for the Koopman method for dynamic system prediction (Pan et al., 2024). The Python package serves as a good reference since it is regularly maintained and has an up-to-date implementation of several central features of the Koopman operator framework. As mentioned above there are two central parameters that affect the performance of the Koopman method: the observables and the regression method. Exploiting the existing implementation in `PyKoopman` we allowed in our CTF testing the variation of the following set of parameters:

- Type of observable: Options include the identity, polynomials of variable degree, time delay (of variable depth), radial basis functions (of variable number) and random Fourier features, as well as the concatation of all of the aforementioned observables with the identity;
- Type of regressor: DMD, EDMD, HAVOK and KDMD;
- Regressor rank;
- Least-squares regularisation and rank of the regularisation (this option is implemented only in EDMD and KDMD).

Note that in principle a neural network-based DMD is also implemented in the PyKoopman package, but in our fine-tuning we found that this lead consistently to worse performance than the above four types of regressors thus we did exclude it from the hyperparameter tuning.

Parametric PyKoopman Out-of-the-box `PyKoopman` does not have a parametric implementation, thus in order to test the Koopman method on task 4, we loosely follow (Andreuzzi et al., 2023; Guo et al., 2025b) and implement a custom parametric version of `PyKoopman` by spline interpolation of the learned Koopman operator and corresponding eigenfunctions. We acknowledge that further work and rigorous testing of various parametric versions of the Koopman method are required to identify the best performing implementation for task 4.

Further comments on the use with chaotic systems We note that the performance of the Koopman operator on the chaotic systems is notably subpar, especially when compared to results reported in prior work (Pan et al., 2024). This is not unexpected and a likely source of challenge is the chaotic nature of equations like Lorenz or Kuramoto-Sivashinsky, which has also been noticed by the authors of the `PyKoopman` package. Essentially, in chaotic systems there may not be a dominating low-rank structure that can be learned and exploited with the Koopman method (cf. the section on "Unsuccessful examples of using Dynamic mode decomposition on PDE system" in (Pan et al., 2024)).

Hyperparameter tuning Based on the available choices implemented in the PyKoopman package and the examples described in the documentation (Pan et al., 2024), we performed a hyperparameter search over the following parameters: type of observable and potential concatenation with the identity, observables integer parameter (representing the polynomial degree in case of polynomial observables, the number of time delay steps in the case of time delay observables and the parameter D in the random Fourier feature case), the number of centers for the radial basis function observables, observables float parameter (representing the radial basis function kernel width and the parameter γ in the radial basis function case respectively), regressor type, regressor rank, TLSQ rank (the regularisation rank called only when the regressor is EDMD and KDMD). The details of the parameter space explored are shown in Table 4.

hyperparameter	type	min (or options)	max (or none)
observables	choice	{Identity, Polynomial, TimeDelay, RadialBasisFunctions, RandomFourierFeatures}	·
Identity concatenation	choice	{true, false}	
Integer parameter	randint	1	10
# RBF centers	randint	10	1000
Float parameter	uniform	0.5	2.0
regressor type	choice	{DMD,EDMD, HAVOK, KDMD}	·
regressor rank	randint	1	200
TLSQ rank	randint	1	200

Table 4: Hyperparameter search space for the PyKoopman model.

D.4 Sparse Identification of Nonlinear Dynamics

Sparse Identification of Nonlinear Dynamics (SINDy) (Brunton et al., 2016) is a powerful algorithm designed to discover interpretable and parsimonious governing equations from time-series data. Given the data matrices

$$X = \begin{bmatrix} x_1(t_1) & x_1(t_2) & ... & x_1(t_m) \\ \vdots & \vdots & \ddots & \vdots \\ x_n(t_1) & x_n(t_2) & ... & x_n(t_m) \end{bmatrix}; \quad \dot{X} = \begin{bmatrix} \dot{x}_1(t_1) & \dot{x}_1(t_2) & ... & \dot{x}_1(t_m) \\ \vdots & \vdots & \ddots & \vdots \\ \dot{x}_n(t_1) & \dot{x}_n(t_2) & ... & \dot{x}_n(t_m) \end{bmatrix} \tag{24}$$

collecting, respectively, the state vector $\mathbf{x}(t) = [x_1(t), ..., x_n(t)]$ and the corresponding time derivatives $\dot{\mathbf{x}}(t) = [\dot{x}_1(t), ..., \dot{x}_n(t)]$ at the time instances $t_1, ..., t_m$, we aim at identifying the (possibly nonlinear) underlying governing equation $\dot{\mathbf{x}}(t) = f(\mathbf{x}(t))$. To this aim, SINDy considers the following approximation

$$\dot{X} = \Theta(X)\Xi \tag{25}$$

where $\Theta(X)$ is a library of candidate regression terms, such as polynomials or trigonometric functions, while Ξ are the corresponding regression coefficients. Sparsity promoting strategies are crucial to identify simple and interpretable dynamical systems, capable of avoiding overfitting and accurately extrapolating beyond training data. In particular, the regression coefficients Ξ are determined through sparse regression strategies, such as Least Absolute Shrinkage and Selection Operator (LASSO) or Sequentially Thresholded Least SQuares (STLSQ). See Figure 6 for a scheme of the SINDy algorithm on the Lorenz system.

SINDy can easily handle parametric dependencies: indeed, augmenting the state vector with the (possibly time-dependent) parameter values $\boldsymbol{\mu}$ and adding $\boldsymbol{\mu}$-dependent terms in the library $\Theta(X, \boldsymbol{\mu})$, it is possible to identify parametric sparse dynamical systems.

Identifying sparse dynamical systems from high-dimensional data may be computationally expensive. A possible workaround is given by dimensionality reduction techniques, such as Proper Orthogonal Decomposition (POD) (Brunton et al., 2016) or autoencoders (Champion et al., 2019), which project state snapshots onto a low-dimensional manifold. SINDy can thus be applied on the low-dimensional latent variables, allowing for efficient and accurate forecasting of the high-dimensional state evolution.

SINDy implementation The implementation of SINDy is based on the *PySINDy* library (de Silva et al., 2020). After collecting the data and approximating the time derivatives through numerical schemes, the SINDy algorithm is applied to identify a sparse dynamical system describing the data evolution over time. The integrator *solve_ivp* by *scipy* (Virtanen et al., 2020) is considered to simulate the system and to predict future state values. Notice that, whenever the identified model is very complex and the integrator fails, the static dynamical system $\dot{\mathbf{x}} = 0$ is employed.

For high dimensional data, Proper Orthogonal Decomposition (POD) is thus exploited to compress the temporal data, and SINDy is applied to identify the dynamics of the most energetic POD coefficients. Therefore, the predictions are retrieved by integrating the SINDy model and projecting the POD coefficients onto the original high-dimensional state space.

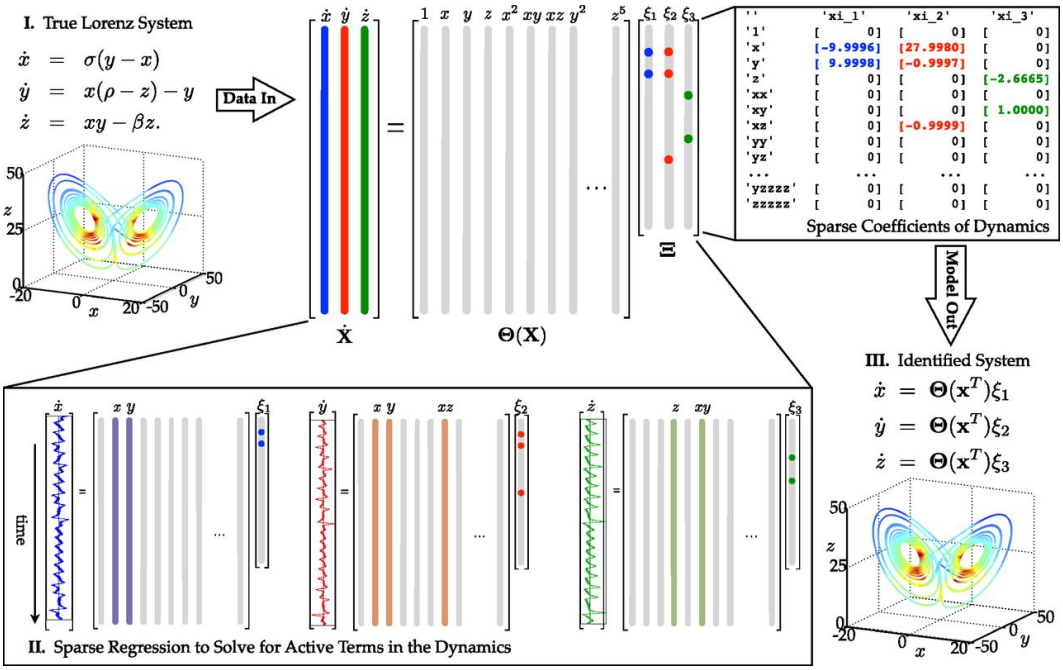


Figure 6: Schematic of the Sparse Identification of Nonlinear Dynamics (SINDy) algorithm from (Brunton et al., 2016), demonstrated on the Lorenz equations. The temporal evolution of the state variable and its derivative are collected in the data matrices X and $\dot{X}$. The dynamical system $\dot{X} = \Theta(X)\Xi$ is then identified through sparsity promoting algorithms.

hyperparameter	type	min (or options)	max (or none)
POD_modes	integer	1	50
differentiation_method	choice	{ ”finite_difference”, ”spline”, ”savitzky_golay”, ”spectral”, ”trend_filtered”, ”kalman” }	·
differentiation_method_order	integer	1	10
feature_library	choice	{ ”polynomial”, ”Fourier”, ”mixed” }	·
feature_library_order	integer	1	10
optimizer	choice	{”STLSQ”, ”SR3”, ”SSR”, ”FROLS”}	·
threshold	choice	{ ”adam”, ”L-BFGS” }	·
learning_rate	loguniform	10^{-3}	10^{3}
alpha	loguniform	10^{-3}	10^{1}

Table 5: Hyperparameter search space for SINDy.

Parametric SINDy models are considered when testing the ability of the model to generalize to different parameter values. Since the parameter values employed for data generation are not publicly available, we take into account fictitious values mimicking the interpolatory and extrapolatory regimes.

Hyperparameters The SINDy algorithm can exploit different differentiation methods to approximate time derivatives, different terms in the library $\Theta(X)$ – such as, e.g., polynomials and/or trigonometric functions up to a chosen order – as well as different sparse regression techniques. Table 5 provides a summary of the hyperparameters in play, along with the corresponding search spaces explored for hyperparameter tuning.

hyperparameter	type	min (or options)	max (or none)
hidden_state_size	randint	8	256
seq_length	randint	5	512
lr	log_uniform	10^{-5}	10^{-2}

Table 6: Hyperparameter search space for the ODE-LSTM and LSTM models on metrics E_1 through E_6 for the tested datasets. We train with a batch size of 128 for 200 epochs.

hyperparameter	type	min (or options)	max (or none)
hidden_state_size	randint	8	256
seq_length	randint	5	74
lr	log_uniform	10^{-5}	10^{-2}

Table 7: Hyperparameter search space for the ODE-LSTM and LSTM models on metrics E_7 through E_{12} for the tested datasets. We train with a batch size of 5 for E_7 through E_{10} and a batch size of 128 for E_{11} and E_{12} for 200 epochs.

D.5 LSTM/ODE-LSTM

LSTM networks are a specialized type of recurrent neural network (RNN) designed to address the vanishing gradient problem inherent in traditional RNNs (Hochreiter & Schmidhuber, 1997). They achieve this through a unique architecture featuring memory cells and gating mechanisms (input, forget, and output gates), which regulate the flow of information over time. These gates enable LSTMs to selectively retain or discard historical data, making them particularly adept at capturing long-term dependencies in sequential data. In time-series forecasting, LSTMs excel at modeling temporal patterns, such as trends, seasonality, and irregular fluctuations, by leveraging past observations to predict future values. Their ability to handle complex, non-linear relationships and variable-length input sequences makes them a robust choice for tasks like stock prediction, energy load forecasting, or weather modeling, where historical context is critical to accurate predictions.

ODE-LSTMs are a flavor of LSTMs that try to further tackle the vanishing gradient problem by using an ODE solver to model the hidden state of the LSTM (Coelho et al., 2024). They show that traditional LSTMs can still suffer from a vanishing or exploding gradient and provide theory demonstrating ODE-LSTMs do not suffer from either of these problems.

We evaluate both a classical LSTM and the ODE-LSTM by searching over the following hyperparameters: hidden_state_size (dimension of the latent space), seq_length (input sequence length), and lr (learning rate).

D.6 Reservoir Computing

In its broadest sense, reservoir computing (RC) is a general machine learning framework for processing sequential data. RC functions by projecting data into a high-dimensional dynamical system and training a simple readout from these dynamics back to a quantity or signal of interest. Although there exists a large and ever-growing body of literature on leveraging physical systems to act as high-dimensional "reservoirs" (Tanaka et al., 2019), the most common form of RC remains an echo state network (ESN) (Jaeger, 2001a; Maass et al., 2002). ESNs are a form of recurrent neural network (RNN) that have been demonstrated to achieve state-of-the-art performance in the forecasting of chaotic dynamical systems (Platt et al., 2022; Vlachas et al., 2020). We now introduce the specific form of ESN we use in evaluating performance on the CTF datasets, following many of the conventions presented in (Platt et al., 2022).

ESNs for Low-Dimensional Systems. Given a time series $u_0, \ldots, u_T$, a randomly instantiated, high-dimensional dynamical system is evolved according to

$$h_{t+1} = (1-\alpha)h_t + \alpha \tanh\left(W_{hh}h_t + W_{hu}u_t + \sigma_b \mathbf{1}\right) \tag{26}$$

where α is the so-called leak rate hyperparameter, W_{hh} and W_{hu} are fixed, random matrices, σ_b is a bias hyperparameter and $\mathbf{1}$ denotes a vector of ones. $W_{hh} \in \mathbb{R}^{N_h \times N_h}$ (N_h denotes the number of

hyperparameter	type	min (or options)	max (or none)
α	uniform	0	1
σ	loguniform	0.0001	1.0
σ_b	uniform	0	2
ρ	uniform	0.02	1
β	loguniform	10^{-10}	10^{-1}
N_h	randint	500	7500

Table 8: Hyperparameter search space for the reservoir model on the tested dataset.

hyperparameter	type	min (or options)	max (or none)
hidden_state_size	randint	8	1024
seq_length	randint	5	74
batch_size	randint	5	120
lr	log_uniform	10^{-5}	10^{-2}

Table 9: Hyperparameter search space for Neural-ODE models. We train for 100 epochs.

entries in h) is taken to be a random, sparse matrix with density $\approx 2\%$ and non-zero entries sampled from $\mathcal{U}(-1,1)$ and then scaled such that the spectral radius of W_{hh} is ρ. $W_{hu} \in \mathbb{R}^{N_h \times N_u}$ (N_u denotes the number of entries in u) is a random matrix with each entry drawn independently from $\mathcal{U}(-\sigma,\sigma)$. Initializing h_0 as $h_0 = \mathbf{0}$, we generate a sequence of training reservoir states $h_0,\ldots,h_T$. We discard the initial N_{spin} training states as an initial transient and then perform a Ridge regression (with Tikhonov regularization β) to learn a linear map W_{uh} such that

$$W_{uh} g(h_i) \approx u_i. \tag{27}$$

$g : N_h \to N_h$ is often taken to be the identity map or simply squaring every odd indexed entry of h_i. We assume the latter convention, following the work of Pathak et al (Pathak et al., 2018). Once trained the reservoir dynamics can be run autonomously as

$$h_{t+1} = (1-\alpha)h_t + \alpha \tanh\left(W_{hh}h_t + W_{hu}W_{uh}g(h_t) + \sigma_b \mathbf{1}\right) \tag{28}$$

to obtain a forecast of arbitrary length. For the considered CTF4Nuclear datasets, an ESN as described above is trained to forecast the evolution of coefficients for the first 32 POD modes approximated from the training data. The complete hyperparameter search space for the tested datasets are given in Table 8.

D.7 NEURAL-ODE

Nerual-ODEs are a type of neural network that uses an ODE solver to model the hidden state of a neural network.(Chen et al., 2018). This is very similar to ODE-LSTMs, another model evaluated in this work, except it makes use of a vanilla MLP instead of LSTM.

We search over the following hyperparameters: hidden_state_size (dimension of the latent space), seq_length (input sequence length), batch size, and lr (learning rate).

D.8 SPACETIME

State-Space Models (SSMs) are mathematical frameworks that describe systems using latent (hidden) states evolving over time, observed through measurable outputs. They are widely used in control theory, signal processing, and time-series analysis to model dynamic systems. Modern adaptations like S4 (Structured State Space for Sequence Modeling) and SpaceTime are deep learning variants of SSMs tailored for sequential data. These models parameterize state transitions with structured matrices to efficiently capture long-range dependencies while remaining computationally tractable. Unlike LSTMs, SSMs are particularly effective at time-series forecasting of long-range dependencies with minimal memory overhead.

SpaceTime (Zhang et al., 2023) is one such SSM that claims to be a state-of-the-art model on time-series forecasting and classification tasks. The authors claim that their model captures "complex,

hyperparameter	type	min (or options)	max (or none)
lag	randint	32	512
horizon	randint	32	512
n_blocks	choice	{3,4,5,6}	·
dropout	choice	{0, 0.25}	·
weight_decay	choice	{0, 0.0001}	·
kernel_dim	choice	{32,64,128}	·
lr	log_uniform	10^{-5}	10^{-2}

Table 10: Hyperparameter search space for the SpaceTime model on metrics E_1 through E_6 for the tested datasets. We train with a batch size of 128 for 200 epochs.

hyperparameter	type	min (or options)	max (or none)
lag	randint	10	45
horizon	randint	10	45
n_blocks	choice	{3,4,5,6}	·
dropout	choice	{0, 0.25}	·
weight_decay	choice	{0, 0.0001}	·
kernel_dim	choice	{32,64,128}	·
lr	log_uniform	10^{-5}	10^{-2}

Table 11: Hyperparameter search space for the SpaceTime model on metrics E_7 through E_{10} for the tested datasets. We train with a batch size of 5 for 200 epochs.

long-range, and *autoregressive*" dependencies, can forecast over long horizons, and is efficient during training and inference. They demonstrate improved performance over the popular S4 SSM and NLinear.

Based on the hyperparameter optimization described in the original paper and the hyperparameters which can be adjusted in the publicly available code, we do a hyperparameter search over the following values: lag (input sequence length), horizon (output sequence length), n_blocks (number of SpaceTime layers in the model encoder), dropout, weight_decay, kernel_dim (dimension of SSM kernel in each block), and lr (learning rate).

D.9 Deep Operator Networks

Deep Operator Networks (DeepONets) (Chen & Chen, 1995) recently emerged as a powerful tool designed to efficiently model high-dimensional physical systems and complex input-output relationships, as well as to solve challenging problems in scientific machine learning and engineering, such as partial differential equations. Specifically, DeepONets are a class of neural operators which decompose an operator $G : \mathcal{V} \to \mathcal{U}$ between infinite-dimensional functional spaces $\mathcal{V}$ and $\mathcal{U}$ into two cooperating sub-networks, namely *branch* and *trunk net*. The trunk encodes the input function $v \in \mathcal{V} : \Omega' \subset \mathbb{R}^d \to \mathbb{R}^{n_v}$ – which is typically sampled at a finite set of n fixed sensors, resulting in the measurement vector $\mathbf{v} \in \mathbb{R}^{n \cdot n_v}$ – into p coefficients $\mathbf{b}(v) \in \mathbb{R}^p$. Instead, the branch net provides the evaluation of a neural learnable p-dimensional basis $\mathbf{t}(\xi) \in \mathbb{R}^p$ at the spatial coordinates ξ in the domain $\Omega \subset \mathbb{R}^d$. Doing so, the value of the output function $u \in \mathcal{U} : \Omega \to \mathbb{R}^{n_u}$ at the evaluation point $\xi \in \Omega$ is approximated through the basis expansion

$$u(\xi) = G(v)(\xi) \approx \mathbf{b}(v) \cdot \mathbf{t}(\xi).$$

See (Chen & Chen, 1995; Lu et al., 2022) for a complete presentation of DeepONets, including also universal approximation theorems for operators. A graphical summary of the DeepONet architecture is available in Figure 7.

DeepONets for dynamical systems DeepONets are versatile neural architectures designed to learn mappings between functional spaces. DeepONets are traditionally exploited for inferring the space-time evolution of physical variables, such as the solution of partial differential equations, starting from known quantities, such as forcing terms, initial conditions, parameters or control variables

hyperparameter	type	min (or options)	max (or none)
lag	randint	10	45
horizon	randint	10	45
n_blocks	choice	{3,4,5,6}	·
dropout	choice	{0, 0.25}	·
weight_decay	choice	{0, 0.0001}	·
kernel_dim	choice	{32,64,128}	·
lr	log_uniform	10^{-5}	10^{-2}

Table 12: Hyperparameter search space for the SpaceTime model on metrics E_{11} through E_{12} for the tested datasets. We train with a batch size of 128 for 200 epochs.

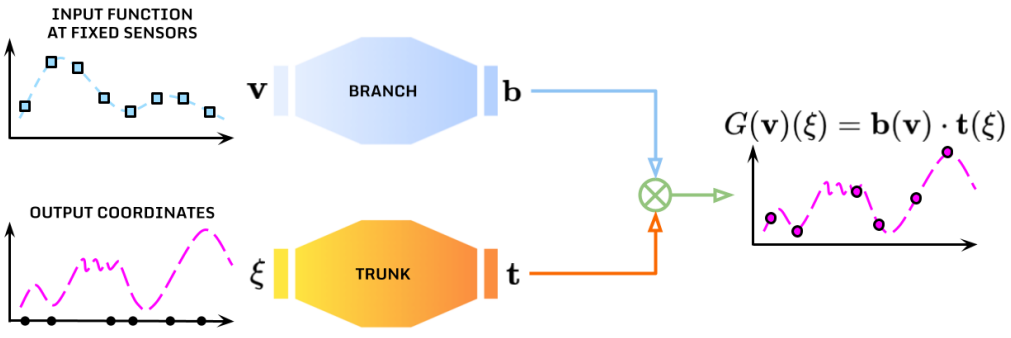


Figure 7: Architecture of the Deep Operator Network. The target field at the evaluation point ξ is approximated by the inner product of the outputs of the branch net, which takes as input the measurements $\mathbf{v}$ of the input function $v \in \mathcal{V}$ and returns a set of coefficients $\mathbf{b}(\mathbf{v})$, and the trunk net, which encodes the coordinates ξ into a vector $\mathbf{t}(\xi)$.

hyperparameter	type	min (or options)	max (or none)
lag	integer	1	99
branch_layers	integer	1	5
trunk_layers	integer	1	5
neurons	integer	1	512
activation	choice	{"tanh", "relu", "elu"}	·
initialization	choice	{"Glorot normal", "He normal"}	·
optimizer	choice	{ "adam", "L-BFGS" }	·
learning_rate	loguniform	10^{-5}	10^{-1}
epochs	integer	10000	10000

Table 13: Hyperparameter search space for DeepONet.

(Chen & Chen, 1995; Lu et al., 2022; Wang et al., 2021; Kontolati et al., 2024). However, it is possible to adapt and employ DeepONets in the proposed CTF in order to model and forecast time-series data and dynamical systems, as proposed by, e.g., (Lin et al., 2023; He et al., 2024a;b; Michałowska et al., 2024). Specifically, we consider the operator

$$u_t(\xi) = G\left(u_{t-1}, ..., u_{t-k}\right)(\xi) \approx \mathbf{b}\left(\mathbf{u}_{t-1}, ..., \mathbf{u}_{t-k}\right) \cdot \mathbf{t}\left(\xi\right)$$

where $u_t : \Omega \to \mathbb{R}^{n_u}$ and $\mathbf{u}_t \in \mathbb{R}^n$ are, respectively, the solution of the dynamical system under investigation at time t and the corresponding spatial discretization, k is the lag parameter and $\xi \in \Omega \subset \mathbb{R}^d$ are the spatial coordinates where to predict the evolution of the dynamics. Along with the evaluation point ξ, the trunk input may be enlarged with the time instance t or the time-step Δt, as proposed by (Lu et al., 2022; Lin et al., 2023).

DeepONets implementation The implementation of DeepONets within the proposed CTF is based on the *DeepXDE* library (Lu et al., 2021b). In particular, when dealing with forecasting tasks, we predict the state evolution in an autoregressive manner, and we enlarge the trunk input with the time-step Δt, as it results in better performance. As proposed by (Lu et al., 2022), we consider a scaler to normalize the data before training. Moreover, we employ branch and trunk networks with the same number of neurons per hidden layer, so as to reduce the number of hyperparameters. The datasets are discretized and evaluated over uniformly spaced domains. Notice that we take into account the same locations across all the input-output pairs, resulting in a lower computational cost.

Hyperparameters The DeepONet hyperparameters mainly concern the neural network architectures and the corresponding training procedure. In addition, the lag parameter determines the length of the past state history fed into the branch input for forecasting. Notice that the lag value cannot be larger than the dimension of burn-in data, and it is set equal to zero when dealing with reconstruction tasks. Table 13 provides a summary of the hyperparameters in play, along with the corresponding search spaces explored for hyperparameters tuning.

D.10 Fourier Neural Operator

Neural operators are a class of machine learning models designed to learn mappings between function spaces, in contrast to the finite-dimensional Euclidean spaces typically used in conventional neural networks. Although the inputs and outputs are discretized in practice, neural operators aim to generalize across discretizations and treat functions as the primary objects of learning.

The Fourier Neural Operator (FNO), in particular, is a neural operator architecture that replaces the kernel integral operator with a convolution operator defined in Fourier space, which allows for learning of operators in the frequency domain. It maps the input to the frequency domain using the Fourier transform, applies spectral convolution by multiplying learnable weights with the lower Fourier modes, and maps the result back to the physical domain via the inverse Fourier transform. This allows the model to learn families of PDEs, rather than solving individual instances. Without the high cost of evaluating integral operators, it maintains competitive computational efficiency.

Let $D \subset \mathbb{R}^d$ be a bounded domain. We consider learning an operator G that maps between function spaces:

$$G : \mathcal{A} \to \mathcal{U} \tag{29}$$

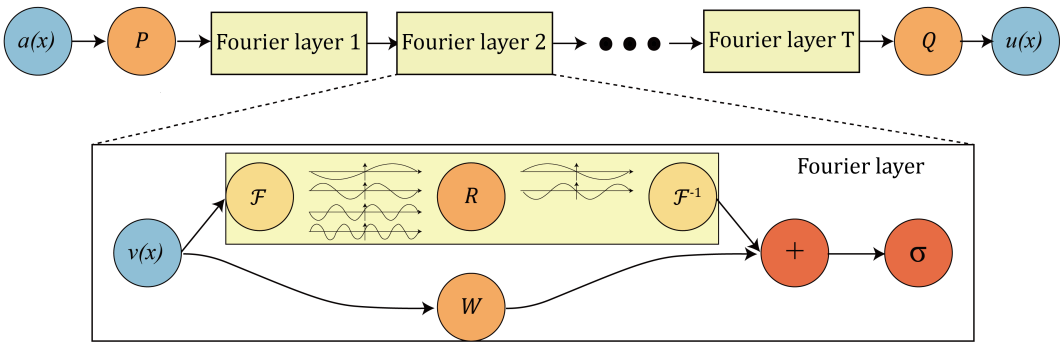


Figure 8: Architecture of the Fourier Neural Operator from (Li et al., 2021b)

where $\mathcal{A} = L^2(D; \mathbb{R}^{d_a})$ is the input function space and $\mathcal{U} = L^2(D; \mathbb{R}^{d_u})$ is the output function space.

Given an input function $a \in \mathcal{A}$, the FNO approximates the operator G through a kernel integral operator:

$$G(a)(x) = \sigma\left(Wa(x) + b + \int_D \kappa(x, y)a(y)\, dy\right) \tag{30}$$

where $W \in \mathbb{R}^{d_u \times d_a}$ is a linear transformation, $b \in \mathbb{R}^{d_u}$ is a bias term, $\kappa : D \times D \to \mathbb{R}^{d_u \times d_a}$ is a learnable kernel function, and $\sigma : \mathbb{R}^{d_u} \to \mathbb{R}^{d_u}$ is a pointwise non-linear activation function.

The kernel is parameterized in Fourier space as:

$$\kappa(x, y) = \sum_{k \in \mathbb{Z}^d} \widehat{\kappa}(k) e^{2\pi i k \cdot (x-y)} \tag{31}$$

where $\widehat{\kappa}(k)$ are the Fourier coefficients of the kernel. The translation-invariant kernel $\kappa(x, y) = \kappa(x - y)$ enables efficient convolution. This leads to the implementation:

$$G(a)(x) = \sigma\left(Wa(x) + b + \sum_{k \in \mathbb{Z}^d} \widehat{\kappa}(k)\widehat{a}(k) e^{2\pi i k \cdot x}\right) \tag{32}$$

where $\widehat{a}(k)$ represent the Fourier coefficients of the input function a. In practice, the sum over $k \in \mathbb{Z}^d$ is truncated to a finite number of low-frequency modes.

Model Architecture The architecture (Figure 8) begins with an initial fully connected multilayer perceptron (MLP) that projects the input to a higher-dimensional space, followed by four Fourier layers, and concludes with two fully connected MLPs that project the output to the desired dimensions.

Each Fourier layer performs a spectral convolution by first transforming the data into the frequency domain using Fast Fourier Transform (FFT), then multiplying the Fourier coefficients with learable weights in the frequency space, and finally transforming back to physical space using inverse FFT. The Fourier layer only keeps a limited number of the lower Fourier modes, with high modes being filtered out. Additionally, each layer adds a linearly transformed version of its input to the output of the spectral convolution, which helps preserve local features and adds flexibility to the layer's expressiveness. Every Fourier layer is followed by a GELU activation function to introduce non-linearity.

Hyperparameters Based on our implementation of the FNO model, which closely follows that of the original paper, we test the hyperparameters as shown in Table 14. The number of Fourier modes is tuned separately for each mode.

hyperparameter	type	range or options
Fourier modes	integer	[8,32]
Network width	integer	[32, 128]
Batch size	choice	16, 32, 64, 128
Learning rate (lr)	loguniform	[0.0001, 0.01]

Table 14: Hyperparameter search space for the FNO model.

D.11 KOLMOGOROV-ARNOLD NETWORKS

Kolmogorov–Arnold Networks (KANs) are a recently proposed alternative to traditional Multi-Layer Perceptrons (MLPs) (Liu et al., 2025c). With learnable activation functions placed on edges that replace linear weights, KANs have been shown to provide improved accuracy and greater interpretability compared to traditional methods.
KANs were inspired by the Kolmogrov-Arnold representation theorem which posits that any multivariate continuous function f on a bounded domain can be expressed as a finite composition and addition of univariate continuous functions (Kolmogorov, 1957). In other words, for a smooth function $f : [0, 1]^n \to \mathbb{R}$,

$$f(\mathbf{x}) = f(x_1, x_2, ..., x_n) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right) \tag{33}$$

where $\phi_{q,p} : [0, 1] \to \mathbb{R}$ and $\Phi_q : \mathbb{R} \to \mathbb{R}$.

Model Architecture While the Kolmogrov-Arnold representation theorem is restricted to a small number of terms and only two hidden layers, this theorem can be generalized to increase the width and depth of the network. A single KAN layer is defined as a matrix of 1D functions thus the inner and outer functions in Equation 33, $\phi_{q,p}$ and Φ_q, each represent a single KAN layer. A deeper network can be constructed by adding more KAN layers. A general KAN network with L layers can be represented as a composition of L functions:

$$f(\mathbf{x}) = \sum_{i_{L-1}=1}^{n_{L-1}} \phi_{L-1,i_L,i_{L-1}} \left(\sum_{i_{L-2}=1}^{n_{L-2}} \cdots \left(\sum_{i_2=1}^{n_2} \phi_{2,i_3,i_2} \left(\sum_{i_1=1}^{n_1} \phi_{1,i_2,i_1} \left(\sum_{i_0=1}^{n_0} \phi_{0,i_1,i_0}(x_{i_0}) \right) \right) \right) \cdots \right)$$

where n_l is the number of nodes in the l^{th} layer and $\phi_{l,j,k}$ is the activation function that connects the k^{th} neuron in the l^{th} layer to the j^{th} neuron in the $l + 1$ layer. The network architecture is better illustrated in Figure 9 which was adapted from Figure 2.2 in (Liu et al., 2025c).

Each activation function is comprised of a basis function $b(x)$ and a spline function:

$$\phi(x) = w_b b(x) + w_s \text{spline}(x)$$

where

$$b(x) = \text{silu}(x) = \frac{x}{1 + e^{-x}}$$

$$\text{spline}(x) = \sum_i c_i B_i(x)$$

Initially, w_s is set to 1 and $\text{spline}(x) \approx 0$. The weights of the basis function is initialized according to Xavier initializations.

KAN Implementation Although KANs have primarily been applied to science-related tasks such as function approximation and PDE solving, in this work, we adapted the model to address the reconstruction and forecasting tasks posed in the Common Task Framework. The data was processed and loaded similarly to example 14 in the `pykan` repository.

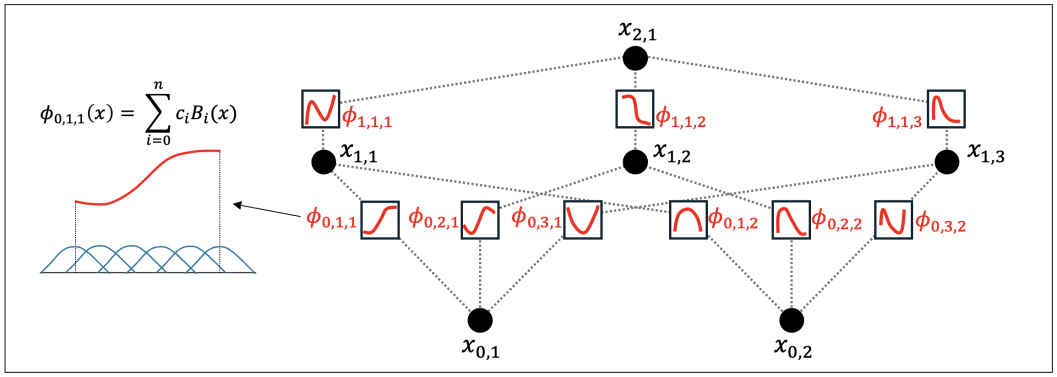


Figure 9: Sample architecture of a Kolmogorov-Arnold Network with three layers of size $[2, 3, 1]$. Activation functions ϕ are placed on the edges and are parametrized as a spline. Each output of a node is a sum of its inputs.

hyperparameter	type	min (or options)	max (or none)
steps	randint	400	10^3
lag*	randint	1	2
lr	loguniform	10^{-5}	10^{-1}
num_layers	randint	1	5
{one-five}_dim**	randint	1	5
grid	randint	1	100
k	randint	1	3
λ	loguniform	10^{-7}	10^{-3}
λ_{coef}	loguniform	10^{-7}	10^{-3}

Table 15: Hyperparameter search space for the KAN model on the tested datasets. NOTE: The lag parameter is set to zero for reconstruction tasks (pair_id = 2 or 4)*. The dimension of each layer is defined separately. For example the number of nodes in layer two would be defined as *two_dim***.

For forecasting tasks, the input-output pairs were constructed in an autoregressive manner, where each input consisted of lagged observations used to predict future values. The input and output dimensions depend on both the number of spatial dimensions in the dataset and the chosen lag (l). By implementing the model in this way, the KAN network approximates the optimal l-step numerical scheme to solve the given dynamical system.
If the dataset dimension is d and with a lag of l, the input dimension was set to $d_{\text{in}} = d \cdot l$ and the output dimension to $d_{\text{out}} = d$.

For reconstruction tasks, the model was trained in an autoencoding fashion, where each input was mapped directly to itself as the target output: the input and output dimensions were both set to $d_{\text{in}} = d_{\text{out}} = d$.

Hyperparameters Based on the hyperparameter settings provided in the `pykan` package and the results reported in the original paper (Liu et al., 2025c), the hyperparameters outlined in Tables 15 were selected and tuned for this model. Broadly, the hyperparameters fall into two categories: (1) model architecture and (2) training.
Architecture-related hyperparameters include the number of layers, dimensions of hidden layers, grid resolution, the polynomial degree of the spline basis (k), and the lag. Training-related hyperparameters include the number of training steps (epochs), learning rate, overall regularization strength (λ), and the regularization coefficient for the spline parameters (λ_{coef}).

D.12 LLMTIME

LLMTime (Gruver et al., 2023) is a time-series foundation model that uses pre-trained LLMs to perform zero-shot forecasting of time-series data. Their approach is to modify the tokenization of each model so that time-series forecasting is casted as a next-token prediction in text problem. For our evaluation, we used the `llama-7b` as LLMTime's base LLM and used the default temperature of 1.0, an alpha of 0.99, and a beta of 0.3. We also used LLMTime's default Llama tokenizer. LLMTime is only able to forecast univariate time-series, so we auto-regressively forecast each dimension with a context of 200 tokens and a prediction length of 100 tokens at a time. Once each dimension has been forecasted, they are concatenated and evaluated on the test set. For reconstruction tasks, we take the first 10 time-steps of the training data and forecast each dimension until we have a vector containing the same number of timesteps as in the testing dataset and then concatenate and calculate our metrics as before.

D.13 CHRONOS

Chronos (Ansari et al., 2024) is a pre-trained probabilistic time-series foundation model from Amazon. The model is informed by the success of transformers and LLMs, and as such tokenizes time series values using scaling and quantization and trains using the cross-entropy loss function. The model is only capable of doing univariate time-series forecasting. For our evaluation, we use the pre-trained `chronos-t5-base` model and do a one-shot forecast of each dimension of each dataset independently and concatenate them when calculating our metrics. For reconstruction tasks, we take the first 10 time-steps of the training data and forecast each dimension until we have a vector containing the same number of timesteps as in the testing dataset and then concatenate and calculate our metrics as before. Chronos has a much smaller context length than LLMTime due to requiring more VRAM for inference.

D.14 MOIRAI

Moirai_MoE (Liu et al., 2025a) is a time-series forecasting foundation model from Salesforce AI Research. The model uses a sparse mixture-of-experts transformer architecture and is able to do one-shot multivariate time-series forecasting on arbitrary time-series datasets. For our evaluation, we used the pre-trained `base` model and predicted 10 time-steps at a time with a context length of 20. For reconstruction tasks, we take the first 10 time-steps of the training data and forecast until we have a matrix containing the same number of timesteps as the testing dataset. Moirai_MoE has a much smaller context length than LLMTime due to requiring more VRAM for inference.

D.15 SUNDIAL

Sundial (Liu et al., 2025b) is a family of native, flexible and scalable time-series foundation models from Tsinghua University, tailored specifically for time series analysis. It is pre-trained on TimeBench (about one trillion time points), adopting a flow-matching approach rather than fixed parametric densities. Sundial directly models the distribution of next-patch values in continuous time-series without discrete tokenisation; it is built on a decoder-only Transformer architecture. For our evaluation, we used the pre-trained `sundial-base-128m` model; the model can handle multivariate time-series forecasting directly. For the evaluation of high-dimensional spatial systems, due to RAM limitations, we have split the "spatial" dimension into batches, forecasting each batch independently and concatenating the results. For reconstruction tasks, we take some of the first time-steps of the training data (around 10%) and forecast until we have a matrix containing the same number of timesteps as the testing dataset.

D.16 PANDA

Panda (Lai et al., 2025) is a foundation model for nonlinear dynamical systems based on Patched Attention for Nonlinear DynAmics. Panda is motivated by dynamical systems theory and adopts an encoder-only architecture with a fixed prediction horizon. It is pre-trained purely on a synthetic dataset of 2×10^4 chaotic dynamical systems, discovered using a structured algorithm for dynamic systems discovery introduced in the same work. For our evaluation, we used the pretrained model

weights provided on the official code repository associated with (Lai et al., 2025). The main free parameter in the forecasts with Panda is the context length.

D.17 TABPFN-TS

TabPFN for Time Series (TabPFN-TS) (Hoo et al., 2025) is based on the tabular foundation model TabPFNv2 (Hollmann et al., 2025), adapted to the task of time series forecasting. We use the pre-trained model weights, leaving the only remaining parameter as the amount of data for each specific system that the model is exposed to before performing zero-short forecasting.